Discovery of Web Services with a P2P Network

Florian Forster and Hermann De Meer

University of Passau
Faculty of Mathematics and Computer Science
Innstrasse 33
94032 Passau
{forsterf,demeer}@fmi.uni-passau.de

Abstract. In the concept of Web Services, Universal Description, Discovery and Integration is still the weakest part. As a central instance, it does not easily scale to a growing number of users and lacks acceptance by the industry. In Peer-to-Peer Networks, which are highly popular, discovery of resources is one of the strongest parts. A central registry is not required when integrating Web Services in a Peer-to-Peer network. Each Web Service is responsible for itself and the Peer-to-Peer Network provides the framework for discovery, publication and registration of Web Services. This paper shows, how both technologies fit together and gives details on both structure and design of the Peer-to-Peer network. This results in a more feasible solution than the central Universal Description, Discovery and Integration infrastructure and provides an easy way for registration, publishing and discovery of Web Services.

1 Introduction

Web Services (WS) and Peer-to-Peer (P2P) networks have evolved to popular technologies over the last few years. P2P was mostly pushed by users of popular software like the MP3-sharing program Napster[1] or the distributed analysis program SETI@Home [14]. WS are a concept coming from global business players like Microsoft, IBM or Sun.

P2P focuses on a de-centralized architecture of equal peers for the purpose of sharing resources of the participating peers, whereas WS are using a Client/Server approach, and have the intention of Business-to-Business (B2B) or Business-to-Costumer (B2C) integration.

WS use the markup language XML [1] for communication via a standardized protocol called SOAP [3]. Description of the services is done by the Web Service Description Language (WSDL [4]) . Publishing and discovery relies on the concept of Universal Description, Discovery and Integration (UDDI [2]). The goal of WS is to provide a common interface for Internet applications enabling everybody who implements software to rely on certain standards. With the use of the Web Service Flow Language (WSFL [15]), virtual organizations can be

[1] http://www.napster.org

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3038, pp. 90–97, 2004.

built up by making use of different WS. Discovery of adequate business partners plays an important role in the creation of virtual organizations.

P2P networks focus on the possibility for users to share their resources. Each peer who joins the network has to register itself, and the provided resources. Hybrid P2P networks use a central instance for registration and discovery, whereas in pure P2P networks this is done by active announcement to the network. A lot of research and development effort is put in discovery algorithms for P2P networks, which resulted in various strategies and P2P network topologies.

P2P networks enforce the sharing of resources, and WS took the first step in the creation of virtual organizations by the invention of WSFL. Creating a P2P network of WS provides an easy way of sharing resources for organizations. In this paper, the P2P network organizes the discovery and sharing of resources, whereas the WS provide and describe the resources shared. This combination also fits with the concept of *The Grid*[2]. The first step will be to show, how discovery, publishing and registration of WS can be done in a P2P network. This paper shows how to distribute information regarding WS among the nodes in a P2P network, thus making the centralized component of the UDDI concept obsolete.

After giving some references to related work in section 2, section 3 provides a brief overview to the concepts of WS and P2P. Section 4 provides a proposal for a P2P architecture replacing UDDI and gives detailed information about publishing (Section 4.3), discovery (Section 4.4) and registration (Section 4.2) in the network. Finally, section 5 summarizes the ideas.

2 Related Work

The basic idea that WS could benefit from P2P networks was mentioned in [5], [6], [7],[12]. A modified Gnutella Client, capable of searching for WSDL files has been implemented by the ProSa Discovery Project[8]. In [9], a general purpose query language was proposed, allowing complex searches suitable for WS. [10] proposed the *Web Service Discovery Architecture* which provides a basic framework for discovery of WS with a P2P network by evaluating different technologies and standards. Using JXTA as an implementation framework resulted in a basic P2P network for WS in [11]. This approach also focused on using WSDL, meaning that complex search queries are not possible in this network. The METEOR-S System [17] offers a scalable infrastructure of registries for semantic publication and discovery of WS. The Web Service Inspection Language [16] is a complementary concept to UDDI which does not need a central infrastructure and therefore is not taken into account in this paper.

[2] http://www.grid.org

3 Motivation

3.1 Web Services

WS basically provide a framework for interoperability between applications, and thus sharing of resources or information, over the Internet. Therefore, the concept of WS defines the roles provider, consumer and registry. Obviously, provider and consumer are absolutely necessary for a WS, but the registry entity is optional. Provider and consumer could contact each other without using the registry.

Compared with WSDL and SOAP, the UDDI standard still is in a relatively immature stage of development. Although IBM, Ariba, SAP and Microsoft are offering an infrastructure for a central UDDI registry, this entity is not widely used. As for the moment, most business still rely on direct contact to their partners. One reason my be the susceptibility of UDDI to security threats. Business simply do not seem to trust the UDDI registries and thus do not use them. Furthermore, the registries tend to be outdated because the update process of the entries in the registry is still often done manually. In the traditional approach, UDDI was designed as a single server, which leads to have a single point of failure for discovery of Web Services. IBM, Ariba and Microsoft tried to solve this by creating several servers, replicating each other. This approach still faces feasibility problems like many Client/Server Architectures. For WS, being one of the future technologies, it is very likely that there will be a huge number of WS in a few years.

That is why an infrastructure for UDDI registries has to be created for WS very fast. It is very unlikely that the growth of the UDDI infrastructure satisfies the upcoming requirements of WS. The capacity of the UDDI infrastructure would enforce an upper bound to the number of enlisted WS, and thus limit the growth of WS. Nevertheless, a global UDDI registry could help to create new B2B or B2C relationships by providing detailed information about a business and their services, allowing complex search queries.

3.2 P2P

P2P is a concept for sharing resources, mostly files, amongst peers participating in a network. P2P is divided into the concept of hybrid and pure P2P. Whereas in hybrid P2P Systems, a central entity, which provides a registry and helps in the discovery process, is available, pure P2P Systems consist of equal peers only. By joining the network, a new peer automatically registers itself to the network, either by signing up at a central entity or by announcing its presence to the network. P2P networks rely on the availability of a good discovery mechanism. Due to the fact that peers join or leave the network often, peers cannot establish direct contact without discovering one another first. Peers mostly provide very simple information during the registration process. Because of this, discovery of information has to stick to simple search queries.

Compared to UDDI, P2P has two important advantages. First, it is widely accepted among users. Second, the registration to a P2P network is done automatically and is very simple in nature. In the next section, a P2P architecture for discovery and publication of WS is presented.

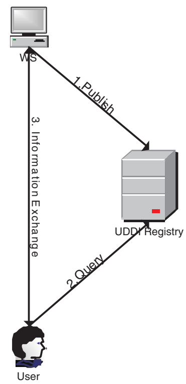

Fig. 1. Service Discovery with UDDI

4 Web Service Publishing and Discovery with P2P

4.1 Discovery and Publishing in WS and P2P

Discovery and publishing in WS is done by using a central entity. A provider of a WS submits several description files to a UDDI registry. The process of publishing is clearly defined by the *UDDI Publishing API*. Consumers searching for services send a query to the UDDI registry. The functionality of the search is defined by the *UDDI Query API*. Due to the fact that UDDI stores many description files, complex search queries can be implemented in the client software. After retrieving information from the UDDI registry, the consumer can contact the provider of the WS and get the requested information (see Fig. 1).

Discovery and publishing in a P2P network can be done in various ways. In a hybrid P2P network, it is very similar to the process of WS. A peer joining the network registers itself at a central instance, which manages all the resources in the P2P network. Search queries are sent to the central instance, which replies with several sources providing the requested information. In a pure P2P network there is no such central instance. Search queries are either broadcasted to a limited neighbourhood or sent to a specific set of peers. The usage of a central instance guarantees a success rate of 100%, meaning that if a ressource is present in the network, then it will be registered at the central instance and thus be found by queries. Guaranteeing a 100% success rate in a pure P2P network results in heavy traffic. Each search query has to be sent to an overwhelming part of the network to give the success assurance of 100%. But a hybrid P2P network depends on the central instance. If the central instance fails, no more discovery is possible within the network, whereas the pure P2P network does not have to face this problem due its distributed design.

As P2P networks are widely accepted and have proved to be feasible even for a large number of users, they could be an appropriate replacement for the UDDI infrastructure. The P2P network has to provide an easy way of registration and publishing of information. Furthermore, it has to give a nearly 100% assurance

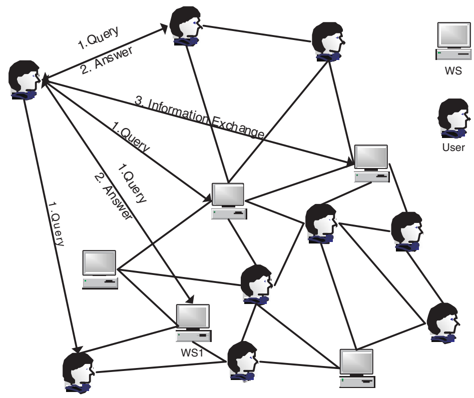

Fig. 2. Service Discovery with P2P

that a WS is found in the network by using appropriate queries, and has to provide the possibility for complex search queries, in order to be suitable as a replacement for UDDI. In this paper every WS is its own registry, similar to peers in pure P2P networks. These hosts are the essential part of the P2P network replacing UDDI. The P2P architecture consists of two different layers. On the first layer there are the peers providing information about WS (WS-Peer), whereas the second layer consists of consumers (User-Peer) joining the network for discovery purpose. Publishing of information is done by simply joining the P2P network. An algorithm like the one in Gnutella [13] is used for discovery. For better performance, more sophisticated algorithms for search have to be implemented in the future. A peer searching for a specific WS, sends its query to the network. The query is then forwarded in the network until its TTL[3] is reached. In case of a match to the query, a peer replies to the requester. After that, the requester can contact the peer providing the requested information directly (see Fig. 2). Details on publishing, discovery and registration are given in the next sections.

4.2 Registration of a Peer

For joining the network, a peer has to know the address of a WS-Peer for bootstrapping purposes. The joining peer is provided with a mixed set of addresses of User-Peers and WS-Peers which becomes the starting set for its neighbourhood. In case of a WS-Peer, only other WS-Peers are considered as neighbours. Each of the neighbours is contacted and the time-stamp of the response is stored along with its address. If a peer forwards a search query or gets responses to a search query, it adds the addresses of the peers involved in this process to its neighbourhood along with the time-stamp of the query received last. In case of a full neighbourhood set, the oldest peer is deleted from the set. For the sake of discovery, at least one of the peers in the neighbourhood has to be a WS-Peer. If none of the User-Peers has contact to the WS-Peer layer no WS at all can be found.

[3] Time to Live

Furthermore, a WS-Peer announces the provided information about its own WS, meaning **key**,**name** and **description** for each local file to its neighbourhood on a regular base. Each peer in the neighbourhood then adds the information to its cache (see section 4.4). Due to the dynamic nature of the neighbourhood, over time, more peers will have information pointing directly to the specific WS-Peer and thus increasing the chance for this WS-peer to be found with a search query.

As seen, the network consists of two layers. User-Peers are connected to each other and to WS-Peers, whereas WS-Peers are only connected to WS-Peers. By making sure that every User-Peer has at least one WS-Peer in its neighbourhood, a search query reaches the WS-Peer-layer. Due to the announcement of information and the dynamics of the neighbourhood, information regarding WS is spread along the caches in the P2P network. User-Peers involved in a search are only forwarding queries or, in case of a local match, provide a direct link to the WS-Peer. As shown, registering peers is as easy as it is in pure P2P networks

4.3 Publishing Information

For the complete description of a WS, UDDI relies on several models. These models are used to provide detailed information about business. For publishing its information a provider joins the P2P network and provides the files containing these models to the network. The uniqueness of **BusinessKey**, **ServiceKey** and **BindingKey**, each identifying a specific model, is a problem as there is no central instance assuring the uniqueness of the key. Usage of a hash-function mapping the static IP address of a provider to an unique identifier is sophisticated enough for this purpose. The **tModelKey** can be generated by using a hash-function on the description of the *tModel*. Besides the adjustment of the creation of an unique identifier for the description files, no further changes have to be made to the UDDI standard for the description files.

UDDI encourages WS covering the same topic to use the same *tModel* for providing a common interface to clients. Because there is no central instance for the storage of such *tModels* the idea of similar services using the same model has to be solved. For example, if a new company is planing to provide a weather service via a WS on its own, it has to query the P2P network for an appropriate *tModel* first. If it does not find one it can create its own and introduce it to the P2P network. If there is already an appropriate *tModel* in the network, the *bindingTemplate* has to contain a reference to this remote file.

Each participating company in the P2P network is now able to publish their information in the network. Remote references to models are solved by querying the P2P network for the unique ID and then storage of the results in a local cache for further requests. There is no additional effort for WS providers, compared to UDDI, by using a P2P network. The same files have to be created for using a P2P network as a replacement for UDDI.

4.4 Finding and Caching of Information

Searching for a business makes use of the function *find_business*. Each peer receiving the query parses the local files, using a SAX or DOM parser, and searches the elements **businessKey**, **name** and **description** for a match to the query. After checking the local repository it checks its local cache. The cache is split into four sections. There is a cache for business, services, bindings and tModels. Each cache contains a complex structure which includes, in the case of the business-cache, the elements **businessKey**, **name**, **description** and a reference to the source of this file. In the case of a match in the local repository, the peer sends the information **businessKey**, **name**, **description** to the requester. In the case of a match in the cache, it sends an answer which redirects the requester to the original source. The requester then caches the information itself and contacts the original source getting requested information. If the original source is not available, the requester informs the peer of the old entry in the cache. In case of using one of the *get*-functions the requester would first use the proper *find*-function with the unique key as parameter. This search is going to result in only one match and the requester is then able to request the file containing detailed information from the providing peer directly. Finding and getting information regarding services, bindings or tModels works the same way.

This strategy enables the peers to fill their local caches, making them able to reply to search queries. The cache is kept up-to-date by the method described in the last paragraph. Although each peer is able to answer queries, the complete file is only available at the WS provider, which limits the possibilities of abuse. Due to the fact that peers can search their local repository, complex search queries are possible within the network. After showing that the P2P network is able to provide the same functionality as the *UDDI Query API*, the next section provides details on the registration process for peers.

5 Conclusion

WS still lack an easy way of registration and discovery. Even though much effort is put into the concept of UDDI, UDDI is still not accepted amongst providers of WS. This is because of the fact that the registration process in UDDI is complicated and updates are still often done manually. Furthermore, as a central instance for registration, publishing and discovery, UDDI may face feasibility problems in the future, as the number of WS is very likely to grow fast in the near future. This paper showed a way on how to distribute the information regarding a WS in a P2P network by providing two architectural layers. For this purpose techniques and algorithms of P2P networks were used. Under the new scheme, the description files for WS remain at the host providing the WS. This host participates in a P2P network and registers its service within the network. Consumers searching for WS also join the network. By using SAX or DOM parsers for local files, complex search queries are still possible for the clients. This results in a P2P architecture suitable for replacing the central UDDI registry, helping to create virtual organizations for the purpose of resource sharing. The success rate for discovery depends on the algorithm used. Further work has to be done for finding an appropriate one.

References

1. Tim Bray et al. (2000): Extensible Markup Language (XML) 1.0 (Second Edition) in *W3C Recommendation 6 October 2000*, W3C, http://www.w3.org/TR/REC-xml
2. Ariba, IBM, Microsoft (2000): UDDI Technical White Paper, UDDI Project, http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf
3. Martin Gudgin et al. (2003): SOAP Version 1.2 Part 1: Messaging Framework in *W3C Recommendation 24 June 2003*, W3C, http://www.w3c.org/TR/2003/REC-soap12-part1-20030624/
4. Erik Christensen et. al (2001): Web Services Description Language (WSDL) 1.1 in *W3C Note 15 March 2001*, W3C, http://www.w3.org/TR/wsdl
5. Remigiusz Wojciechowski, Christof Weinhardt (2002): Web Services und Peer-to-Peer Netzwerke in *Schoder, Fischbach, Teichmann (2002): Peer-to-Peer*, Springer Verlag, 99–117
6. Jeff Schneider (2001): Convergence of Peer and Web Services, http://www.openp2p.com/pub/a/p2p/2001/07/20/convergence.html
7. Nathan Torkington (2001): Why you should care about P2P and Web Services at *Yet Another Perl Conference*, http://prometheus.frii.com/∼gnat/yapc/2001-ws-p2p/
8. Olaf Gregor (2003): ProSa-Discovery-Project - Discovery von WebServices mit Gnutella, http://vsis-www.informatik.uni-hamburg.de/∼1gregor/
9. Wolfgang Hoschek (2002): A Unified Peer-to-Peer Database Framework and its Application for Scalable Service Discovery, in *Proc. of the Int. IEEE/ACM Workshop on Grid Computing, Baltimore, USA, November 2002*, Springer Verlag, 126–144
10. Wolfgang Hoschek (2002): Peer-to-Peer Grid Databases for Web Service Discovery, in *Grid Computing: Making the Global Infrastructure a Reality (to appear), Editors: Fran Berman, Geoffrey Fox and Tony Hey*, Wiley Press.
11. Ludwig Mittermeier and Roy Oberhauser (2002): Ad-hoc-Web-Services durch P2P-Technologien, in *JavaSpektrum 04/2002*, SIGS Verlag, 28–39
12. Mike P. Papazoglou, Bend J. Kraemer, Jian Yang (2003): Leveraging Web-Services and Peer-to-Peer Networks, in *Johann Eder and Michele Missikoff (2003): Lecture Notes in Computer Science Vol. 2681*, Springer, 485–501
13. Gene Kan (2001): Gnutella in *Andy Oram (2001): Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, O'Reilly 94–122
14. David Anderson (2001): Seti@Home in *Andy Oram (2001): Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, O'Reilly, 67–76
15. Frank Leymann (2001): Web Services Flow Language 1.0, IBM Document, http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf
16. Keith Ballinger et al. (2001): Web Services Inspection Language 1.0, http://www-106.ibm.com/developerworks/webservices/library/ws-wsilspec.html
17. Verma, K. et al. (2003): METEOR'S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services in *Journal of Information Technology and Management* (to appear).