

Apply Service Oriented Architecture with Web 2.0 Application for Computer Network Web Services

Ming-Chang Huang

Department of Business Information Systems / Operation Management

University of North Carolina at Charlotte

mhuang5@uncc.edu

Abstract

It is important how wireless hosts find other hosts efficiently for load and web service purposes because hosts in an ad-hoc network moves dynamically. This paper proposes a three-tier architecture, which includes content network, social network and service network. It also presents a structure for web and load services in ad-hoc computer networks, which is a new system architecture using SOA (Service Oriented Architecture) and Web 2.0 concepts to implement functions for web and load services. It is a three-tier system structure based on Web service functions to implement services seeking and load distribution. Furthermore, this project would construct a knowledge sharing and learning platform based on the mentioned three-tier architecture. Different communities can provide their services to each other using this new knowledge platform and this forms a “virtual community.” This will leads to the desired accomplishments of “service reusability” and “service innovation” too. In addition, it can also propose the frameworks of new SOA, and evolve in other application in web2.0 style, and further more provides a platform and more resources in order to enhance the interactions between academia and industry.

Keywords: Web Services, Web 2.0, SOA, Service Oriented Architecture, Ad-hoc computer network, Load service and distribution

1. Introduction

Computer networks can provide parallel computation and services. It is important that hosts find services from other hosts, and send loads to other hosts for some certain function implementation through network transfer. With the increasing popularity of mobile communications and mobile computing, the demand for web and load services grows. When a computer is overloaded or it needs special services from other computers, it may send requests to other computers for web and load services. For example, a computer may need some jobs to be executed with higher quality of services or it needs some jobs to be done with a short period of time that its processor is too slow to perform the jobs; therefore, it may send part those jobs to other computers with higher speeds of processors. Since wireless networks have been widely used in recent years, how a host finds services it needs or how it transfers loads to other nodes has become a very important issue because not all wireless hosts have the ability to manipulate all their loads. For instance, a host with low battery power cannot finish all its jobs on time and should ask other nodes to provide services to finish the jobs, or it should transfer some of them to other hosts.

Before a wireless host transfers its loads to other hosts or asks for load services from other hosts, it has to find available hosts using resource allocation algorithms. There are several resource allocation protocols been developed, for example, IETF Service Location Protocol (SLP) [8] and Jini [25] software package from Microsystems. However, these protocols address how to find the resources in wired networks, not in wireless networks. Maab [15] develops a location information server for location-aware applications based on the X.500 directory service and the lightweight directory access protocol LDAP [26]; while it does not cover some important issues about the movements of mobile hosts, for example, how to generate a new directory service and how a host gets the new services, when a directory agent moves away its original region. In an Ad-Hoc network, system structure is dynamic and hosts can join or leave any time. Therefore, how to provide load services and how to find available hosts providing load services become importance issues in an Ad-Hoc network system. The goal of this paper

is that users can easily find and share resources based on the concepts of “service reusability” and “service innovation”

Based on the population of Web Services techniques, this paper discusses a new architecture which uses SOA model with Web 2.0 [8, 11] for web services. By using Web 2.0 with SOA, the network resources should be easily found by the hosts which need services. Based on XML, the SOA load service system can be used in any computer system platform [1, 2, 3]. This is a very important characteristic for hosts to share or request services in different systems. With the help of Web 2.0, hosts can find the required services easily from the Internet [4].

Figure 1 shows the basic SOA structure [5, 6, 7], which is built by three major components – the Directory, Service consumer, and Service provider. SQA is operated by the following: The Directory provides a platform for information that a service provider can register in the Directory for providing services; a service consumer can find its desired service it needs in the Directory. Once the Directory finds services that a service consumer needs, it sends a query response back to the service consumer to notify it the result. At this time, the service consumer has the information about the hosts which can provide services; therefore the consumer contacts the service provider directory by sending requests. The service provider now will send responses back to the consumer for the services the consumer needs. This is also called the “invoke” process.

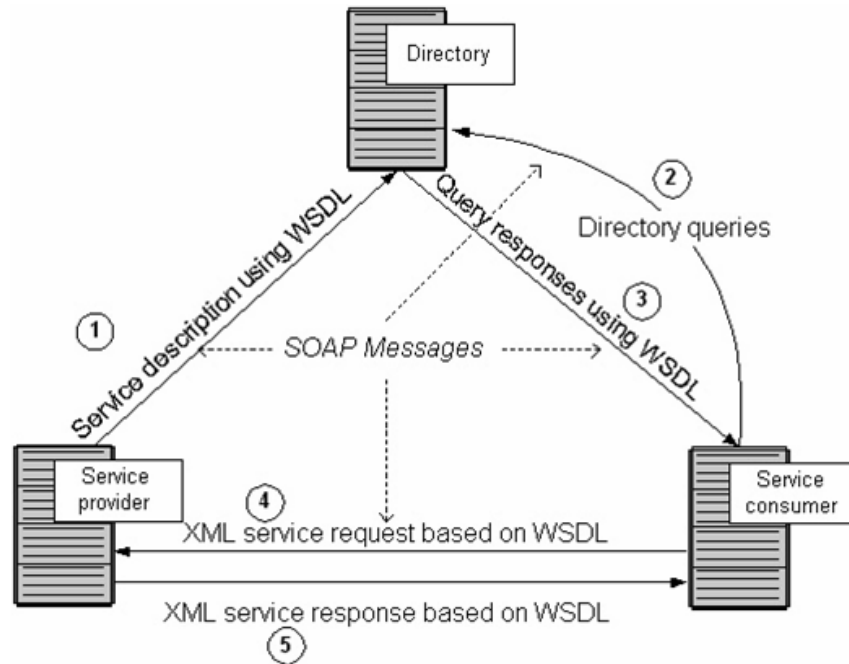


Figure 1: SOA structure

2. System structure

The system structure for the SOA model is illustrated in Figure 2. There are two layers in this structure – the Service Network layer, and the Service Logic layer.

The Service Network layer is the main network that connects to the internet using the regular network protocols. It receives requests from internet and forwards requests to the Service Logic layer in Web Service (WS) object forms. Each WS object is based on the SOA model which can communicate each other in social network way. The Service Logic layer is the main layer that uses WS objects to communicate each other in the sub-network under the Service Network layer. Different WS object has different objectives, for example, some WS objects are used for social network communication, while other WS objects are used for accessing contents in Content Networks. Since they are in SOA form, it is easy for them to find the resources they need for different purposes. Inside the Service Logic layer, there are sub-networks for different purposes and functions. For example, nodes can form social networks; storage devices can also form a content network for data accessing. All these operations are

managed by the WS objects under the SOA model. For the service reusability purpose, most WS objects are generated by the Service Network layer for data and object consistency.

In Figure 2, all the services and requests are in the forms of Web Service objects which are defined and implemented by XML. For users who need services, requests are sent by the users to the Service Network Layer. The Service Network is the gateway for accepting requests and sending back requesting results to the requesters.

All the requests are processed by the Service Logic Layer, which finds the required information and applications for requests. The Content Network is a network which communicates databases. The Social Network contains the relations for social communities. The following procedure illustrates how it works.

1. Users send requests to a Service Network.
2. The Service Network forwards requests to Service Logic Layer via Web service functions.

In this step, requests are transferred to objects that can communicate with the Service Logic layer.

3. When users send requests, Service Network has the ability to generate the desired WS objects according to the requests forwarded by the Service Network.
4. Service Logic Layer performs the required functions for the requests. It accesses data and information from Content Network using Web service functions. A User can also contact other users using WS objects under the Service Logic Layer.
5. After the Service Logic Layer has the results for the requests, it sends the results back to the Service Network using Web service functions.
6. Service Network then sends the results back to the users who sent initial requests.

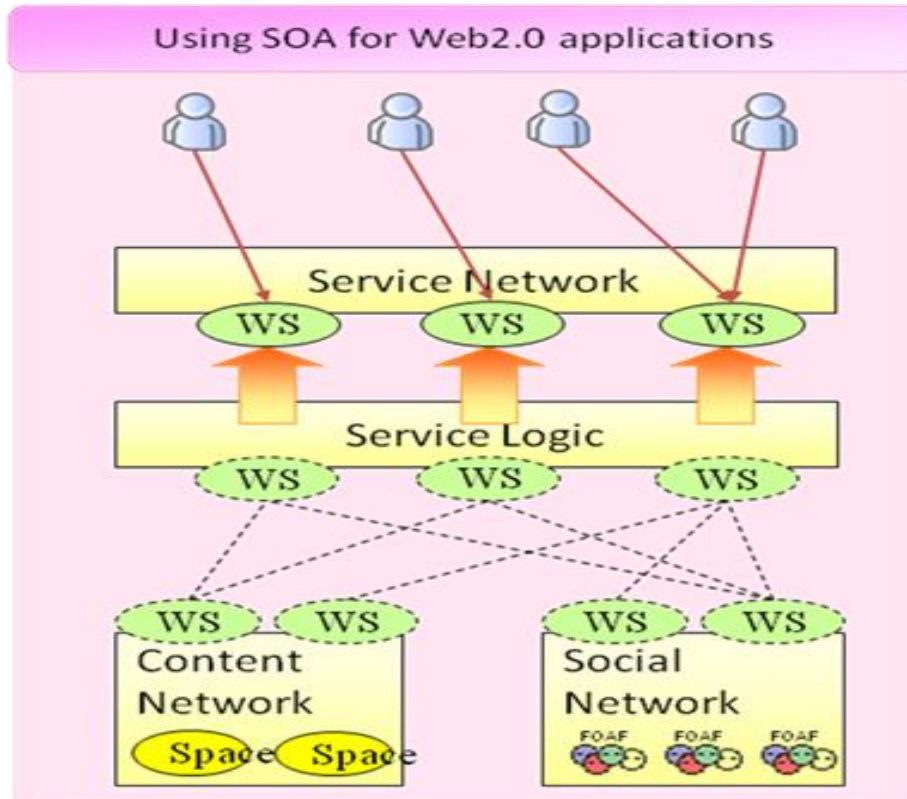


Figure 3: Using SOA for Web2.0 applications

There are several advantages with the design structure.

1. With the system structure, users can join the desired networks anywhere once they connect to the Internet.
2. With the characteristics of Web 2.0 with SOA model, users can join the desired networks they need to share or find resources easily they need.
3. Using WS objects for the communication makes it easy for service reusability and service innovation. Users do not need to construct special system or programs for data accessing and analysis.
4. Different Service Networks communicate with each other to find and share available resources.

3. Implementation and Simulation

Based on the structure of SOA model with Web 2.0 application, the system can be built in a three-tier structure. The lowest level is the sub-networks including Content Network, Social Network, which provides resources for data sharing. The middle level is the Service Logic layer, which provides WS objects. The top level is the Service Network layer that accepts requests or sends request results back to the users.

This paper is going to construct a virtual community, and work on a simulation for data generation and analysis. Ten thousands nodes with a thousand Web services will be used in the simulation. The simulation will compare the performance for data sharing and load transferring to the system without using SOA structure.

4. Conclusion

Usually that it is hard to find the required network resources in the Internet for load balance and load service purposes. Because of this, a new structure is proposed for hosts in the computer networks to find the resources for web services. This new structure provides new way that finds resources easily for web services. Especially, when a user needs services which are not very commonly provided in the Internet, with the help of Web 2.0 and SOA, users can find what they need because of the “long tail” property of Web 2.0 and the platform free property of SOA.

This system performance will be evaluated by using a simulation. Usually it is very hard to evaluate the performance for Web 2.0; therefore this project should have great helps in finding the performances for the usage of Web 2.0 and SOA.

References

- [1] Ross J. Anderson, "The eternity service," *Pragocrypt* 1996, 1996.
- [2] Stephanos Androutsellis-Theotokis and Diomidis Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys*, 36(4):335–371, December 2004. doi:10.1145/1041680.1041681.
- [3] Yvonne Balzer, "Improve your SOA project plans," *IBM*, 16 July 2004
- [4] Biddle, Peter, Paul England, Marcus Peinado, and Bryan Willman, "The Darknet and the Future of Content Distribution," *2002 ACM Workshop on Digital Rights Management*, 18 November 2002.
- [5] Jason Bloomberg, "Mashups and SOBAs: Which is the Tail and Which is the Dog?," *Zaphink*
- [6] Channabasavaiah, Holley and Tuggle, "Migrating to a service-oriented architecture," *IBM DeveloperWorks*, 16 Dec 2003
- [7] Thomas Erl, "About the Principles," *Serviceorientation.org*, 2005-2006
- [8] E. Guttman, C. Perkins, J. Veizades and M. Day, "Service Location Protocol," Version 2, IETF, RFC 2165, November 1998.
- [9] Dion Hinchcliffe, "Is Web 2.0 The Global SOA?," *SOA Web Services Journal*, 28 October 2005
- [10] Huang, M., S. H. Hosseini, and K. Vairavan, *Load Balancing in Computer Networks*, Proceedings of ISCA 15th International Conference on Parallel and Distributed Computing Systems (PDCS-2002), Special session in Network Communication and Protocols. Held in the GALT HOUSE Hotel, Louisville, Kentucky, Sep. 19 - 21.
- [11] Kim, A.J. (2000), "Community Building on the Web: Secret Strategies for Successful Online Communities," *London: Addison Wesley (ISBN 0-201-87484-9)*.
- [12] Christopher Koch, "A New Blueprint For The Enterprise," *CIO Magazine*, Mar 1 2005
- [13] Peter Kollock (1999), "The Economies of Online Cooperation: Gifts and Public Goods in Cyberspace," *Communities in Cyberspace*, Marc Smith and Peter Kollock (editors). London: Routledge.
- [14] Paul Krill, "Make way for SOA 2.0," *InfoWorld*, May 17, 2006
- [15] H. Maab, "Location-Aware Mobile Application Based on Directory Services," *MOBICOM 97*, pp23-33.
- [16] Joe McKendrick, "Anti-SOA 2.0 petition nears 400," *ZDNet.com*, June 29, 2006

- [17] Yefim Natis & Roy Schulte, "Advanced SOA for Advanced Enterprise Projects," *Gartner*, July 13, 2006
- [18] Andy Oram et al., "Peer-to-Peer: Harnessing the Power of Disruptive Technologies," *Oreilly* 2001
- [19] Howard Rheingold (2000), "The Virtual Community: Homesteading on the Electronic Frontier," *London: MIT Press. (ISBN 0-262-68121-8)*.
- [20] Ross, T. J., *Fuzzy Logic with Engineering Applications*, McGraw Hill, 1995.
- [21] Antony Rowstron and Peter Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," *Proceedings Middleware 2001: IFIP/ACM International Conference on Distributed Systems Platforms*, Heidelberg, Germany, November 12-16, 2001. *Lecture Notes in Computer Science*, Volume 2218, Jan 2001, Page 329.
- [22] Ralf Steinmetz, Klaus Wehrle (Eds), "Peer-to-Peer Systems and Applications," *ISBN: 3-540-29192-X*, *Lecture Notes in Computer Science*, Volume 3485, Sep 2005
- [23] Smith, M. "Voices from the WELL: The Logic of the Virtual Commons," *UCLA Department of Sociology*.
- [24] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *Proceedings of SIGCOMM 2001*, August 2001.
- [25] J. Waldo, "The Jini Architecture for network-centric computing," *Communication of the ACM*, pp 76-82, July 1999.
- [26] W. Yeong, T. Howes, and S. Kille, "Lightweight Directory Access Protocol," *RFC 1777*, March 1995.
- [27] <http://opengroup.org/projects/soa/doc.tpl?gdid=10632>