


## A Demand Driven Web Service Lifecycle

Zhaohao Sun

Graduate School of Information Technology and Mathematical Sciences, University of Ballarat

View metadata, citation and similar papers at [core.ac.uk](http://core.ac.uk)

brought to you by  CORE

Dong Dong

School of Computer Science and Technology, College of Mathematics and Information Science  
Hebei Normal University, Shijiazhuang, 050016 China  
e-mail: donald.ddong@gmail.com

Jun Han

School of Computer Science and Engineering, Beihang University  
Beijing, 100191 China  
e-mail: jun\_han@buaa.edu.cn

**Abstract** - Web service lifecycle is a fundamental topic for web services and service computing. This article will examine web service lifecycle and propose a demand-driven model for web service lifecycle taking into account the demand of the main parties in web services. The proposed approach will facilitate the research and development of web services, e-services, service intelligence and service computing.

**Keywords** - web services; web service lifecycle; service computing; service intelligence; multiagent systems.

### I. INTRODUCTION

Web services are Internet-based application components published using standard interface description languages and universally available via uniform communication protocols [4][40]. With the dramatic development of the Internet and the web in the past decade, web services have been flourishing in e-commerce, artificial intelligence (AI), service computing and soft computing because they offer a number of strategic advantages such as mobility, flexibility, interactivity and interchangeability in comparison with traditional services [10].

The fundamental philosophy of web services is to meet the requirements of users precisely and thereby increase market share and revenue [23]. Web services have helped users reduce the cost of information technology (IT) operations and allow them to closely focus on their own core competencies [10]. At the same time, for business marketers, web services are very useful for improving interorganizational relationships and generating new revenue streams [29]. Furthermore, web services can be considered a further development of e-commerce or e-business, because they are service-focused business

paradigms that use two-way dialogues to build customized service offerings, based on knowledge and experience about users to build strong customer relationships [23]. It implies, however, that one of the intriguing aspects of web services is that any web service cannot avoid similar challenges encountered in traditional services such as how to meet the customer's demands in order to attract more customers.

Web service lifecycle is a fundamental topic for web services and service-oriented computing (SOC). Web service lifecycle is the basis for engineering and managing activities in web services. For example, many techniques, approaches, methods have been proposed to facilitate or support the main stages of the entire web service lifecycle [37]. Many web service lifecycles have also been proposed to improve web services with their applications. However, the proposed models for web service lifecycles have not paid sufficient attention to the main parties in web services and the demand of the main parties for web services. If the main parties and their demands are ignored in web services, the healthy development of web services might be problematic, because ignorance of demands in economy and business will lead to economic crisis, like the current global financial crisis. Therefore, this article will alleviate the above mentioned issue by examining demands of the main parties in web services and proposing a demand-driven web service lifecycle for each of the main parties in web services respectively. To this end, the remainder of this article is organized as follows. Section II. looks at parties in web services. Section III. examines the web service lifecycle. Section IV. to 6 examines the demand-driven web service lifecycle for each party mentioned in Section II.. Section VII. provides a unified perspective on demand-driven web service lifecycles. The

final section ends the article with some concluding remarks and future work.

## II. PARTIES IN WEB SERVICES

This section will look at the parties involved in web services and corresponding architectures.

There are mainly three parties related to web services: web service requesters, web service brokers, and web service providers [6][29], as shown in Figure 1. Web service requesters denote web service users, buyers, customers, consumers, receivers, and their intelligent agents. Web service brokers denote web service intermediaries, middle agents and their intelligent agents. Web service providers denote web service owners, sellers, senders and their intelligent agents. Singh and Huhns mentioned these three parties in their simple service oriented architecture (SOA) for web services [25] (p. 20). In this architecture, web service providers create web services and advertise them to potential web service requesters by registering the web services with web service brokers, or simply offers web services [5]. The web service provider also needs to describe the web service in a standard format, and publish it in a central service registry. The service registry contains additional information about the service provider, such as address and contact of the providing company, and technical details about the service. Web service providers may integrate or compose existing services [16] using intelligent techniques such as case-based reasoning (CBR) [28]. They may also register descriptions of services they offer, monitor and manage service execution [5]. Web service requesters retrieve the information from the registry and use the service description obtained to bind to and invoke the web service. Web service brokers maintain a registry of published web services and might introduce web service providers to web service requesters. They use universal description discovery integration (UDDI) to find the requested web services, because UDDI specifies a registry or “yellow pages” of services [25] (p. 20). They also provide a searchable repository of service descriptions where service providers publish their services, service requesters find services and obtain binding information for these services.

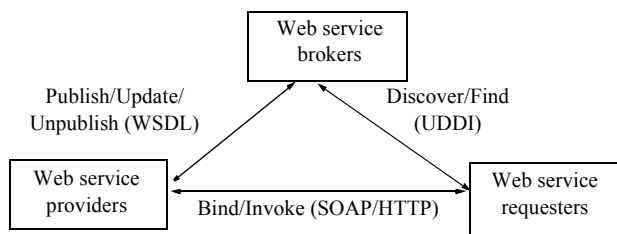


Figure 1. Parties in web services

This architecture is simple because it only includes three parties (as mentioned above) and three basic opera-

tions: publish, find and bind. In fact, some behaviors of web service agents are also fundamentally important in order to make web services successful. These fundamental behaviors at least include communication [25][30], interaction [25][30], collaboration [25][30], cooperation [25][30], coordination [25][30], negotiation [25][30], trust [25] and deception [30].

Papazoglou [17] proposes an extended service-oriented architecture. The parties involved in this architecture are more than that in the simple SOA, because it includes a service provider, a service aggregator, a service client, a market maker, and a service operator.

A service aggregator is a service provider that consolidates services provided by other service providers into a distinct value-added service [17]. Service aggregators develop specifications and/or code that permit the composite service to perform functions such as coordination, monitoring quality of service (QoS) and composition. In our view, a service aggregator should be differentiated from a service provider. We can use *web service recommender* to replace service aggregator, because recommendation is one of the most important activities in web services.

The main task of web market makers is to establish an efficient service-oriented market in order to facilitate the business activities among service providers to service brokers and service requesters. In the traditional market, the service broker is working in the market, while the market maker makes the market operating.

The web service operator is responsible for performing operation management functions such as operation, assurance and support [17].

From a viewpoint of multiagent systems [35], there are still other parties involved in web services, such as web service advisor, web service manager, and web service composer and so on. Further, an activity of web services usually is implemented by a few intelligent agents in a multiagent web services system [30], therefore, more and more intelligent parties or agents will be involved in web services with the development of automating activities of web services. Some of these will be mentioned in the later sections.

## III. WEB SERVICE LIFECYCLE

This section mainly reviews a lifecycle and web service lifecycles and discusses the corresponding issues.

From a perspective of computer science, the concept of lifecycle originated from software engineering [22]. It describes the life of a software product from its conception, to its implementation, delivery, use, and maintenance [21]. A traditional software development lifecycle mainly consists of seven phases: planning, requirements analysis, systems design, coding, testing, delivery and maintenance. Based on this, a web service lifecycle consists of

the start of a web service, the end of web service and its evolutionary stages that transform the web service from the start to the end.

There have been a number of attempts to address web service lifecycle in the web service community [2]. For example, Leymann [15] discusses a lifecycle of a web service based on explicit factory-based approach, in which a client uses a factory to create “an instance” of a particular kind of service; the client can then explicitly manage the destruction of such an instance, or it can be left to the Grid environment. Sheth [24] proposes a semantic web process lifecycle that consists of web description (annotation), discovery, composition and execution or orchestration. Wu and Chang consider service discovery, service invocation and service composition as the whole lifecycle of web services [37]. Zhang and Jeckle propose a lifecycle for web service solutions that consists of web service modeling, development, publishing, discovery, composition, collaboration, monitoring and analytical control from a perspective of web service developers [39]. Kwon proposes a lifecycle of web services consisting of four fundamental steps: web service identification, creation, use and maintenance [13]. Narendra and Orriens [19] consider a web service lifecycle consisting of web service composition, execution, midstream adaptation, and re-execution etc. Tsalgatidou and Pilioura [31] propose a web service lifecycle that consists of two different layers: a basic layer and a value-added layer. The basic layer contains web service creation, description, publishing, discovery, invocation and unpublishing, all of these activities are necessary to be supported by every web service environment. The value-added layer contains the value-added activities of composition, security, brokering, reliability, billing, monitoring, transaction handling and contracting. These activities bring value-added functionality and better performance to any web service environment. They acknowledge that some of these activities take place at the web service requester’s site, while others take place at the web service broker’s or provider’s site. They also explore technical challenges related to each activity in the web service lifecycle. However, they have not classified the proposed activities of stages in their lifecycle based on web service requesters, providers, and brokers in detail. Some companies and organizations also propose their own web service lifecycle. For example, W3C proposes a service lifecycle for web service management, which is expressed as state transition diagrams [36]. Sun Microsystems considers the lifecycle of web services consisting of four stages: design/build, test, deploy/execute, and manage [27], which can be considered a model for web service developers. From a market perspective, web services mainly consist of three parties: Service providers, service requesters and service brokers [32]. Different parties require different web service lifecycles. Therefore, what is

a web service lifecycle from the viewpoint of web service providers, brokers and requesters respectively? How many stages does a web service lifecycle consist of?

Further, demand is an important factor for market and economy development [11]. The decrease of demand is an implication for economic recession, as happens in the current global financial crisis. Different parties generally have different demands for web services, different demands have also different web service lifecycles. Therefore, what are the demand-driven web service lifecycle from the viewpoint of web service providers, brokers and requesters respectively?

These issues still remain open in web services. The following sections will address these issues by examining the web service lifecycle from a demand viewpoint.

It should be noted that everybody, whether an application user, developer, financier, businessman, or an e-commerce manager, has enjoyed or will enjoy some tangible benefits from web services [7] such as searching information using Google and doing business online. At the same time, he or she demands more and more from web services with the development of the Internet. Therefore, we do not examine the demand of everybody for web services, but the demand of the main parties in web services in what follows, that is, we will look at demand-driven web service lifecycles for web service providers, requesters and brokers respectively.

#### IV. PROVIDER’S DEMAND DRIVEN WEB SERVICE LIFECYCLE

From a web service provider’s demand perspective, a web service lifecycle mainly consists of web service identification [13][14][32], description/representation, creation (design/build, test, deploy) [13][32], publishing [32], composition [16][32], invocation, use and reuse [13], execution or orchestration, management and monitoring [5][32], maintenance [13], billing and security [32].

Web service identification aims to identify appropriate services [14]. Web service invocation is to invoke the discovered web service interface [14]. Web services are published to intranet or the Internet repositories for potential users to locate [32]. Web service unpublishing is sometimes no longer available or needed, or it has to be updated to satisfy new requirements [32].

Web service composition primarily concerns requests of web service users that cannot be satisfied by any available web service [19]. One of the simple Web service compositions is to combine a set of available web services to obtain a composite service that might be recommended to the users. More strictly, web service composition refers to the process of creating customized services from existing services by a process of dynamic discovery, integration and execution of those services in order to satisfy user requirements [16][34]. Web service composition is a

key challenge to manage collaboration among web services [16]. It refers to intelligent techniques and efficient mechanisms of composing arbitrarily complex services from relatively simpler services available over the Internet. Service composition can be either performed by composing elementary or composite services. Composite services in turn are recursively defined as an aggregation of elementary and composite services [5][28]. Web service composition is becoming an important topic for service computing, because composing web services to meet the requirement of the web service requester is the most important issue for web service providers and brokers.

There are many techniques existing for web service composition. For example, Tang et al [32] propose an automatic web service composition method taking into account both services' input/output type compatibility and behavioral constraint compatibility. Further, Dustdar and Schreiner [5] discuss the urgent need for service composition and the required technologies to perform service composition as well as present several different composition strategies.

#### **V. REQUESTER'S DEMAND DRIVEN WEB SERVICE LIFECYCLE**

From a web service requester's demand perspective, a web service lifecycle mainly consists of web service consultation, search [14], matching [14], discovery [14][32], composition, mediation [14], negotiation [30], evaluation and recommendation [29].

Web service discovery is a process of finding the most appropriate web services needed by a web service requester [25]. It identifies a new web service and detects an update to a previously discovered web service [14]. Services may be searched, matched, and discovered by service requesters by specifying search criteria and then be invoked [5][32]. Service invocation is restricted to authorised users [5]. There have been a variety of techniques and approaches developed for web service discovery. For example, OWL-S (of W3C) provides classes that describe what the service does, how to ask for the service, what happens when the service is carried out, and how the service can be accessed [14].

Web service mediation is to mediate the request of web service from the web service requester. Web service negotiation consists of a sequence of proposal exchanges between the two or more parties with the goal of establishing a formal contract to specify agreed terms on the service [38]. Through negotiation, web service requesters can continuously customize their needs, and web service providers can tailor their offers. In particular, multiple web service providers can collaborate and coordinate with each other in order to satisfy a request that they can't process alone.

However, a web service requester might not need to know how the web services are retrieved, discovered and composed internally. Therefore, web service search, matching, and composition might be less important for a web service requesters.

#### **VI. BROKER'S DEMAND DRIVEN WEB SERVICE LIFECYCLE**

Brokering is the general act of mediating between requesters and providers in order to match service requester's needs and providers' offerings [30]. It is a more complicated activity than discovery [32]. A broker should enable universal service-to-service interaction, negotiation, bidding and selection of the highest quality of service (QoS) [25] (p.345-46). Brokering is supported by HP web services platform as a HP web intelligent broker [31]. After discovering web service providers that can respond to a user's service request, HP web services platform negotiates between them to weed out those that offer services outside the criteria of the request.

From a web service broker's demand perspective, a web service lifecycle mainly consists of web service consultation, personalization, search, matching, discovery, adaptation, composition, negotiation, recommendation, contracting and billing.

We propose web service consultation as the start of the web service lifecycle, because the web service customer provides a request for a web service so that the web service broker begins to consultation. In order to provide a service consultation, the web service broker has to conduct web service search, like Google does. During the web service search, the web service broker uses any techniques of web service matching such as CBR [30]. After discovering a number of web services, the web service broker can select one of them to recommend it to the web service customer. If the customer accepts the recommended web service, then the web service can be considered as a web service use/reuse; that is, the existing web service has been reused by customers.

Web service recommendation is to help web service requesters in selecting web services more suitable to their needs [28]. Web service recommendation is a significant challenge for web service industry, in particular for web service brokers. Web service recommendation can be improved through optimization, analysis, forecasting, reasoning and simulation [13]. Recommender systems have been studied and developed in e-commerce, e-business and multiagent systems [26][30]. Sun and Lau has examined case based web service recommendation [29]. However, how to integrate web service recommendation, composition and discovery in a unified way is still a big issue for web services.

Different web service customers have different preferences. Therefore, a web service broker has to personalize

web services in order to meet the requirement of the web service customer satisfactorily. It is necessary to compose web services based on the requirement of customers in order to personalize the web service. At the same time, web service composition allows web service broker to create a composite web service for customers rapidly [32].

Billing concerns service brokers and service providers [32]. Service brokers create and manage taxonomies, register services and offer rapid lookup for services and companies. They might also offer value-added information for

services, such as statistical information for the service usage and QoS data.

## VII. DEMAND DRIVEN WEB SERVICE LIFECYCLE: A UNIFIED PERSPECTIVE

Based on the above discussion, the stages involved in the demand-driven web service lifecycle for web service provider, requester and broker can be summarized as in Table 1. Some of the detailed activities have not been listed in the table because of space limitation. From Table

TABLE 1. DEMAND DRIVEN WEB SERVICE LIFECYCLES: A UNIFIED PERSPECTIVE

Parties in web services	identification	representation	search/matching	match	discovery	consultation	personalization	composition	recommendation	adaptation	mediation	negotiation	invocation	billing	contract
Provider	X	X	X					X					X	X	X
Requester	X		X	X	X	X	X	X	X	X	X	X	X	X	X
Broker	X		X	X	X		X	X	X	X	X			X	X

1, we can intuitively find that service requesters and brokers are the dominant force for developing web service, because they have more demands than service providers, which will be examined in more detail in another paper. In what follows, we discuss the above proposed demand-driven web service lifecycles from a unified perspective.

Some activities in web services are common demands of the main parties: service providers, brokers, and requesters. This means that they share the same web service activity. However, different parties in web services demand the same activity in a different way. For example, the service provider demands “web service search” means that s/he asks web service developers or her/his technology agents to provide efficient web service search function for his or her business. On the other hand, the service requester demands “web service search” means that s/he requires a fast search function from the service provider or broker in order to obtain the most satisfactory web services as soon as possible.

Search and matching are not unique activities related to web services, because they are also involved in database and CBR. For example, Google uses search and matching to provide web services. In fact, search can be considered a common demand for everyone who accesses the Internet or the web. Adaptation, retrieval, classification [14], use/reuse [13], retention or feedback are not unique activities related to web services either, because

they are also stages of CBR cycle [30]. Web service invocation, binding, billing, contract [32] can be considered the common features for any commercial activities. Therefore, we need not discuss each of them in detail in the context of web services. Based on the above discussion, the most important activities in web services can be web service discovery, composition and recommendation: The service requestors demand the service providers and brokers for web services discovery and recommendation; The service brokers demand the service providers for web services discovery and composition; the service providers demands up-to-date techniques and tools for web services discovery, composition and recommendation.

It should be noted that the activities in web services should be classified in a hierarchical way (main services and subservices). For example, identification, search and matching can be subactivities of web service discovery. Then we can handle the detailed activities in web service discovery, composition and recommendation respectively. In addition, demand is a 3-ary relation, that is, service requesters demands service brokers or providers to provide certain web services, and vice versa. However, what properties this 3-ary relation having in the context of web services remains open. Some stages in the web service lifecycle may be absent since the demand disappears. There also are many situations resulting in the demand cancellation.

## VIII. CONCLUSIONS AND FUTURE WORK

This article analyzed the main parties in web services and then proposed a demand-driven model for web service lifecycle taking into account the demand of web services providers, brokers and requestors. The proposed approach will facilitate the research and development of web services, e-services, service intelligence and service-oriented computing.

The proposed web service lifecycle is still in a linear form. In future work, we will develop demand-driven models for web service lifecycle in a spiral and iterative way, as done in software engineering [22].

Demand is an important concept in economics [11]. However, there is less attention in web services although increasing demand of the customers are one of the main tasks for the government of many countries facing the global financial crisis. Therefore, we will investigate the computing basis of demand and then improve the above-mentioned web service lifecycle.

## ACKNOWLEDGEMENTS

This research is supported by the Ministry of Education Hebei, China under Grant No. ZH200815. The authors would like to thank Ms Yanxia Wang and Mr. Zhengke Kang from Hebei Normal University, China, and Dr Andrew Stranieri of the University of Ballarat for providing many useful materials and discussions which contributed in the elaboration of this article.

## REFERENCES

- [1] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, *Web Services: Concepts, Architectures and Applications*. Berlin: Springer-Verlag, 2004.
- [2] A. Bucchiarone, and S. Gnesi, "A Survey on Services Composition Languages and Models," *Proc. the International Workshop on Web Services – Modeling and Testing (Web service-MaTe 2006)*, Palermo, Sicily – June 9, 2006, pp. 51-63.
- [3] J.Y. Chung, K.J. Lin, and R.G. Mathieu, "Web services computing: Advancing software interoperability." *Computer* 36(10), 2003, pp. 35-37.
- [4] H.M. Deitel, P.J. Deitel, B. DuWadt, and L.K. Trees, *Web Services: A technical Introduction*. Upper Saddle River, NJ: Prentice Hall, 2004.
- [5] S. Dustdar, and W. Schreiner W, "A survey on web services composition." *Intl J. Web and Grid Services*, 1(1), 2005, pp. 1-30.
- [6] T. Erl, *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Upper Saddle River, NJ: Prentice Hall, 2006.
- [7] A. Guruge, *Web Services: Theory and Practice*. Amsterdam: Elsevier Inc, 2004.
- [8] B. Henderson-Sellers, and P. Giorgini, Eds, *Agent-Oriented Methodologies*, Hershey: Idea Group Publishing, 2005.
- [9] P.C.K. Hung, H. Li, and J.J. Jeng, "Web service negotiation: An overview of research issues," *Proc 37th Hawaii Intl Conf on System Sciences*, 2004, pp. 1-10.
- [10] K.D. Hoffman, "Marketing + MIS = E-Services," *Comm. of the ACM*, 46(6) 2003, pp. 53-55.
- [11] J. Jackson, and R. McIver, *Microeconomics (7th Edn)*. Australia. McGraw-Hill, 2004.
- [12] U. Küster, B. König-Ries, M. Stern, and M. Klein, "DIANE: An integrated approach to automated service discovery, matchmaking and composition" *Proc. Intl Conf on World Wide Web (WWW2007)*, May 8–12, 2007, Alberta, Canada, pp. 1033-1041.
- [13] O.B. Kwon, "Meta web service: building web-based open decision support system based on web services," *Expert Systems with Applications vol. 24*, 2003, pp. 375–389.
- [14] R. Ladner, et al. "Soft computing techniques for web service brokering," *Soft Computing*, 12, 2008, pp. 1089-1098.
- [15] F. Leymann F. "Web Services: Distributed Applications without Limits: An Outline," *Proc. Database Systems For Business, Technology and Web BTW*, Leipzig, Germany, Feb. 26 – 28, 2003, Springer, 2003. <http://doesen0.informatik.uni-leipzig.de/proceedings/paper/keynote-leymann.pdf>.
- [16] B. Limthanmaphon, and Y. Zhang, "Web service composition with case-based reasoning," *ACM Intl Conf. Proc. Series; Vol. 143, Proc. 14th Australasian Database Conf.*, Adelaide, Australia, 2003, pp. 201-208.
- [17] P. Mike, "Papazoglou Service -Oriented Computing: Concepts, Characteristics and Directions," *Proc. 4th Intl Conf. on Web Information Systems Engineering (WISE2003)*. 10-12 Dec. 2003, pp. 3-12.
- [18] G. Miller, .NET vs. J2EE, *Comm. of the ACM*, 48(7), 2005, pp. 64-67.
- [19] N.C. Narendra, and B. Orriens, "Requirements-driven modeling of the web service execution and adaptation lifecycle," *In Madria S. et al, Eds, ICDCIT, LNCS 4317*, 2006, pp. 314-324.
- [20] C. Petrie, M. Genesereth M, et al, "Adding AI to web services," *In van Elst, Dignum, and Abecker, Eds, AMKM 2003, LNAI 2926*, 2003, pp. 322-338.
- [21] S.L. Pfleeger, and J.M. Atlee, *Software Engineering: Theory and Practice (3rd Edn)*, Pearson Education, Inc. 2006.
- [22] R.S. Pressman, *Software Engineering: A Practitioner's Approach (5th edn)*. Boston: McGraw-Hill, 2001.
- [23] R.T. Rust, and P.K. Kannan, "E-service: A new paradigm for business in the electronic environment," *Comm. of the ACM*, 46(6) 2003, pp. 37-42.
- [24] A. Sheth, "Semantic web process lifecycle: Role of Semantics in Annotation, Discovery, Composition and Orchestration," *Invited Talk, WWW 2003 Workshop on E-Services and the Semantic Web*, Budapest, Hungary, May 20, 2003.
- [25] M.P. Singh, and M.N. Huhns, *Service-oriented computing: Semantics, Processes, and Agents*, Chichester: John Wiley & Sons, Ltd, 2005.
- [26] Z. Sun, and G. Finnie, "A unified logical model for CBR-based e-commerce systems," *Intl J Intelligent Syst*, 20(1), 2005, pp. 29-46.
- [27] Sun Microsystem, *Web services life cycle: Managing enterprise Web services*, White Paper, October 2003, <http://www.sun.com/software>, accessed on 24 Dec 2008.

- [28] Z. Sun, J. Han, Ma D. "A unified CBR approach for web services discovery, composition and composition," Proc. IC-MLC 2009 Perth, Australia, July 10-12, in press.
- [29] Z. Sun, and S.K. Lau, "Customer experience management in e-services," In Lu, Ruan, and Zhang, Eds, E-Service Intelligence: Methodologies, Technologies and applications, Springer, Berlin, 2007, pp. 365-388.
- [30] Z. Sun, and G. Finnie, Intelligent Techniques in E-Commerce: A Case-based Reasoning Perspective. Berlin, Heidelberg: Springer-Verlag, 2004.
- [31] A. Tsalgatidou, T. Pilioura, "An Overview of Standards and Related Technology in web Services," Distributed and Parallel Databases, 12, 2002, pp. 135-162.
- [32] X.F. Tang, C.J. Jiang, Z.J. Ding, and C. Wang, "A Petri net-based semantic web services automatic composition method," Journal of Software (in Chinese), 18(12), 2007, pp. 2991-3000
- [33] M. Wang, W.K. Cheung, J. Liu, X. Xie, and Z. Lou, "E-service/process composition through multi-agent constraint management," Proc. Intl Conf. on Business Process Management (BPM 2006), Vienna, Austria, LNCS 4102, Berlin: Springer, 2006, pp. 274-289.
- [34] M. Wang, J. Liu, H. Wang, W.K. Cheung, and X. Xie, "On-demand e-supply chain integration: A multi-agent constraint-based approach," Expert Systems with Applications, 34 (4), 2008, pp. 2683-2692.
- [35] G. Weiss, Ed, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. Cambridge, Massachusetts, USA: MIT Press, 1999.
- [36] W3C, Web Service Management: Service life cycle, 2004, <http://www.w3.org/TR/2004/NOTE-web-service-lc-20040211/>. Retrieved Dec. 24 2008
- [37] C. Wu, E.A. Chang, "Conceptual Architecture of Distributed Web Services for Service Ecosystems," In Proc. 18th International Conf. on Computer Applications in Industry and Engineering (CAINE 2005), pp. 209-214.
- [38] Y. Yao, F. Yang, and S. Su, "Flexible decision making in web services negotiation," In Euzenat and Domingue, Eds, AIMS 2006, LNAI 4183, Berlin: Springer, 2006, pp. 108-117.
- [39] L.J. Zhang, and M. Jeckle, "The next big thing: Web services composition," In Jeckle, and Zhang, Eds, ICWS-Europe, LNCS 2853, Berlin: Springer, 2003, pp. 1-10.
- [40] <http://conferences.computer.org/icws/2009/>.