Demand driven web services

Sun, Zhaohao, Dong, Dong and Yearwood, John 2010, Demand driven web services. In Leung, Ho-fung, Chiu, Dickson K. W. and Hung, Patrick C. K. (ed), *Service intelligence and service science: evolutionary technologies and challenges*, IGI Global, Hershey, Pa., pp.35-55.

DOI: 10.4018/978-1-61520-819-7.ch003

Chapter 3
Demand Driven Web Services

Zhaohao Sun
University of Ballarat, Australia

Dong Dong
Hebei Normal University, China

John Yearwood
University of Ballarat, Australia

ABSTRACT

Web services are playing a pivotal role in e-business, service intelligence, and service science. Demand-driven web services are becoming important for web services and service computing. However, many fundamental issues are still ignored to some extent. For example, what is the demand theory for web services, what is a demand-driven architecture for web services and what is a demand-driven web service lifecycle remain open. This chapter addresses these issues by examining fundamentals for demand analysis in web services, and proposing a demand-driven architecture for web services. It also proposes a demand-driven web service lifecycle for the main players in web services: Service providers, service requestors and service brokers, respectively. It then provides a unified perspective on demand-driven web service lifecycles. The proposed approaches will facilitate research and development of web services, e-services, service intelligence, service science and service computing.

INTRODUCTION

Web services are Internet-based application components published using standard interface description languages and universally available via uniform communication protocols (ICWS, 2009). With the dramatic development of the Internet and the web in the past decade, web services have been flourishing in e-commerce, e-business, artificial intelligence (AI), and service computing. They have also offered a number of strategic advantages such as mobility, flexibility, interactivity and interchangeability in comparison with traditional services (Hoffman, 2003).

The fundamental philosophy of web services is to meet the needs of users precisely and thereby increase market share and revenue (Rust & Kannan, 2003). Web services have helped users reduce the cost of information technology (IT) operations and allow them to closely focus on their own core

DOI: 10.4018/978-1-61520-819-7.ch003

competencies (Hoffman, 2003). At the same time, for business marketers, web services are very useful for improving interorganizational relationships and generating new revenue streams (Sun & Lau, 2007). Furthermore, web services can be considered a further development of e-business (Gottschalk, 2001), 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 (Rust & Kannan, 2003). However, 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.

Service-oriented architecture (SOA) is an important topic for service computing, service science and service intelligence (Singh & Huns, 2005). The special form of SOA in web services is Web service architectures. Web service architectures are the basis for engineering many activities in web services. Therefore, there are many web service architectures proposed in the web service community (Erl, 2006; Alonso, et al, 2004). Papazoglou (2003) proposes a hierarchical service-oriented architecture (SOA) for web services. Burstein, et al. (2005) propose a semantic web services architecture. However, the existing web service architectures are mainly from the perspective of implementation (Benatallah, et al, 2006) rather than from a demand perspective. It seems that demand-driven web service architecture and corresponding web services have not yet received any attention, to our knowledge, although demand is a critical force for developing web services, just as demand is a driving factor for microeconomics.

Demand is a fundamental concept of economics. Demand refers to "the quantities that people are or will be willing to buy at different prices during a given time period provided that other factors affecting these quantities remain the same" (Wilkinson, 2005, p. 75). The demand for a firm's products determines its revenues and also enables the firm to plan its production (Wilkinson, 2005, p. 71). Then demand is a decisive factor for market. Demand theory is an important part in microeconomics and managerial economics (Wilkinson, 2005, pp. 73-120). Demand theory in managerial economics examines demand curves, demand equations, demand analysis, demand chain, impact factors on demand, demand estimation and so on. Demand analysis is a factor driving e-marketing and e-business strategy objectives (Chaffey, 2007, p. 344). Demand analysis assesses current and projected demand for e-commerce services amongst existing and potential customer segments (Chaffey, 2007, p. 344). Demand chain has drawn attention in the field of supply chain management and customer relationship management since the end of last century (Walters, 2006). The demand chain can be defined as "The complex web of business processes and activities that help firms understand, manage, and ultimately create consumer demand" (Rainbird, 2006). The following problems arise in web services: what is the demand of main service players in web services? what is the demand theory for web services? what is demand analysis? What is a demand chain in web services? what is the mathematical analysis of demand in web services? These problems remain open in web services. This chapter addresses these issues by providing a mathematical analysis for web services taking into account the demand of service players in web services.

The web service lifecycle (WSLC) is a fundamental topic for web services and service computing. The web service lifecycle is also the basis for engineering and managing web services. However, the existing models for web service lifecycles have not paid sufficient attention to the main players in web services and the demand of the main players for web services. If the main players and their demands are ignored in web services, then the healthy development of web services might be problematic, because ignorance of demand in economy and business will lead to economic recession. Therefore, this chapter will address the above mentioned issues by examining fundamentals for demands in web services, and proposing a demand-driven architecture for web services. It also reviews the existing web service lifecycles and proposes a demand-driven web service lifecycle for the main players in web services: Service providers, service requestors and service brokers, respectively. The key ideas behind this chapter are that SOA is fundamental for service intelligence (SI) and service science (SS). Web services are an important application field of SI and SS. Web service architecture is a logical realization of SOA in web services. The demand of main players and their intelligent agents is a central part for web services. Mathematical analysis of demands in web services is a basis for developing demand analysis and demand theory of web services. A WSLC can be considered as a logical implementation of web service architecture. The demand-driven WSLCs are a logical realization of the WSLC. The proposed approach will facilitate the research and development of web services, e-services, service intelligence, service science and service computing.

To this end, the remainder of this chapter is organized as follows. First of all, we review SOA, Web services architecture, web services life cycle and classify main players in web services. Then we analyse demands in web services mathematically, which leads to a new classification for e-commerce. We also examine demand relationships among service providers, brokers and requestors in web services and demand chain in web services. Then we examine web service architectures and provide a demand-driven architecture for web services (DWSOA). We also examine web service lifecycles, propose the demand-driven web service lifecycle for the main players in web services respectively and then discuss the demand-driven web service lifecycles in a unified way. Finally we end the chapter with discussing some future research directions and providing some concluding remarks.

BACKGROUND

Service-oriented architecture (SOA) has been extensively studied in the fields of web services and service-oriented computing (Atkinson, et al, 2004; Singh and Huns, 2005). SOA is a conceptual architecture for implementing e-business, e-services, leaving the networking, transport protocol, and security details to the specific implementation (Gisolfi, 2001). SOA consists of three principal participants: a service provider, a service requestor, and a service broker. These three SOA participants interact using three fundamental operations: publish, find and bind: Service providers *publish* services to one or more service brokers or discovery agencies (Ferris & Farrell, 2003; Burstein, et al, 2005). Service requestors *find* required services via a service broker or a discovery agency and *bind* to them (Gottschalk, et al, 2000; Ferris & Farrell, 2003).

A concrete form of SOA in web services is web service architecture. A web service architecture is a conceptual architecture for implementing web services, which is free of concrete implementation of a web service system owing to its conceptual nature. There are a number of different web service architectures proposed in the web service community. For example, Gottschalk, et al. (2000) propose an IBM web service architecture. This might be the first web service architecture, which is then called a SOA (Gisolfi, 2001). In other words, the web service architecture is the same as SOA in web services. This can be considered as the simplest SOA. Kreger (2001) developed Gottschalk's web service architecture proposed in 2000 by adding artifacts, which mainly consist of service and service description. He uses service registry to replace the service broker in the previous architecture to fulfil the role of discovery agencies (Ferris & Farrell, 2003). Kreger (2001) also provides a business perspective on service provider, requestor and registry. He considers the above-mentioned three fundamental operations: publish, find and bind as the interactions between service provider, service requestor, and service broker. Therefore, Kreger's web service architecture is a further development of web service architectures or SOAs. Talia (2002) explores the open grid services architecture to fully integrate grids and web services, and defines grid as a geographically distributed computation platform composed of a set of heterogeneous machines that users can access via a single interfaces. Burstein, et al. (2005) propose a semantic web services architecture, and consider semantic web services as web services in which semantic web ontologies ascribe meanings to published service descriptions so that software systems representing prospective service clients can interpret and involve them. They also examine the main interactions of web services between service providers and service requestors: service discovery, engagement, and enactment. However, they have not paid much attention to the role of service broker in interactions of web services. They have not focuses on the activities of web services from a viewpoint of web services lifecycle either.

Numerous techniques, approaches, methods have been proposed to facilitate or support the main stages of the entire web service lifecycle (Wu & Chang, 2005). A large number of web service lifecycles have also been proposed to improve web services with their applications. For example, Atkinson, et al. (2004) propose a process model for a typical service, which consists of resources, service logic, and a message-processing layer that deals with message of exchanges. In this model, messages arrive at the service and are acted on by the service logic, utilizing the service's resources as required. This model can be considered as an anatomy of a service in SOA, because it only focuses on the processing of a service rather than the interactions of service providers, requestors and brokers. Kreger (2001) considers web services development lifecycle as the design, deployment, and runtime requirements for each of the players in web services: service registries, providers, and requestors. Each player has specific requirement for each phase of four phases in the development lifecycle: build, deploy, run and manage. Benatallah, et al. (2006) implement a model-driven framework for web service development lifecycle in a prototype platform, Service Mosaic, to model, analyze, and manage service models including business protocols, and adaptors. This framework at least includes protocol definition, protocol analysis, and protocol data management, which are fundamental issues that affect the web service development lifecycle from an implementation perspective. However, they have focused on neither WSLC nor the interoperations or interactions of service providers, brokers and requestors from a demand perspective. Narendra and Orriens (2006) consider a web service lifecycle consisting of web service composition, execution, midstream adaptation, and re-execution. We will turn to web services life cycle once again later when we propose demand-driven web services lifecycle.

Humans are one of the most important decisive forces for development of web services. From a viewpoint of multiagent systems (Weiss, 1999; Sun and Lau, 2007), various intelligent agents are also a decisive force for developing intelligent web services. However, few studies have paid sufficient attention to the main players in web services, to our knowledge. In the next section we will first examine main players in web services.

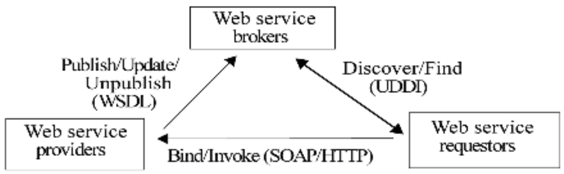
MAIN PLAYERS IN WEB SERVICES

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

There are mainly three players related to web services: web service requestors, web service brokers, and web service providers (Sun & Lau, 2007; Singh & Huhns, 2005), as shown in Figure 1.

Web service requestors also denote web service users, buyers, customers, consumers, receivers, clients, and their intelligent agents. Web service brokers denote web service intermediaries,

Figure 1. Main players in web services



middle agents, registry (Kreger, 2001), discovery agency (Ferris & Farrell, 2003) and their intelligent agents (Burstein, et al, 2005). Web service providers (Kreger, 2001) denote web service owners, sellers, senders and their intelligent agents. Web service requestors, brokers, and providers are the most integral players in web services transactions (Deitel, et al, 2004, p. 52). Gisolfi (2001) mentioned these three players in the simple service oriented architecture (SOA) for web services. In this architecture, web service providers create web services and advertise them to potential web service requestors by registering the web services with web service brokers, or simply offers web services (Dustar & Schreiner, 2005). The web service provider needs to describe the web service in a standard format, and publish it in a central service registry. The service registry or broker 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 (Limthanmaphon & Zhang, 2003) using intelligent techniques. They may also register descriptions of services they offer, monitor and manage service execution (Dustar & Schreiner, 2005). Web service requestors 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 requestors. They use universal description discovery integration (UDDI) to find the requested web services, because UDDI specifies a registry or "yellow pages" of services (Singh & Huhns, 2005, p. 20). They also provide a searchable repository of service descriptions where service providers publish their services, service requestors find services and obtain binding information for these services.

This architecture is simple because it only includes three players (as mentioned above) and three basic operations for web services: 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, interaction, collaboration, cooperation, coordination, negotiation, trust and deception (Singh & Huhns, 2005; Sun & Finnie, 2004; Burstein, et al, 2005).

Papazoglou (2003) proposes an extended service-oriented architecture. The players involved in this architecture are more than that in the simple SOA, because it includes service provider, service aggregator, service client, market maker, and service operator.

A service aggregator is a service provider that consolidates services provided by other service providers into a distinct value-added service (Papazoglou, 2003). Service aggregators develop specifications and/or codes that permit the composite service to perform functions such as coordination, monitoring quality of service (QoS) (Burstein, et al, 2005) and composition. In our view, a service aggregator should be differentiated from a service provider. We can use *web service recommender* or web service composer to replace service aggregator, because recommendation and composition are 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 requestors. 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 (Papazoglou, 2003).

From the viewpoint of multiagent systems (Weiss, 1999; Henderson-Sellers & Giorgini, 2005), there are still other players involved in web services, such as web service advisors, web service managers, web service composers, web service recommenders, web service consultants, and so on. Further, an activity of web services usually is implemented by a few intelligent agents within a multiagent web service system (Sun & Finnie, 2004). Therefore, more and more intelligent players or agents will be involved in web services with the development of automating activities of web services. Although some of these will be mentioned in the later sections, we mainly focus on service providers, brokers and requestors in what follows.

MATHEMATICAL ANALYSIS OF DEMAND IN WEB SERVICES

This section will analyze demands in web services from a mathematical perspective and then discuss the demand relationships in web services.

Demand is an important concept in microeconomics. Jackson and Mclver (2004, p. 74) defines demand as "a schedule that shows the amounts of a product that consumers are willing and able to purchase at each specific price in a set of possible prices during some specified period of time". The basic law of demand is "All else being constant, as price falls, the corresponding quantity demanded rises". Alternatively, the higher the price is, the less corresponding quantity demanded.

Demand analysis has drawn attention in e-business. Chaffey (2007) defines demand analysis as "assessment of the demand for e-commerce services among existing and potential customer segments" (p. 218). He then analyzes the factors that affect demand for e-commerce services (Chaffey, 2007, pp. 150-60.) and uses demand analysis to examine current projected customer use of each digital channel with different markets (Chaffey, 2007, p. 344).

Demand, in particular "on-demand" (Dan, et al, 2004), has also drawn some attention in web services. For example, Burstein, et al. (2005) examine functional and architectural demands or requirements for service discovery, engagement and enactment in terms of the semantic web service architecture. However, in the above-mentioned discussion, it seems that the subject of the demand and its objective are ignored to some extent. For example, who demands what from where is usually unclear. It may not be critical for traditional economics and e-commerce. However, it is useful for web services to know who, what and where exactly for web services, which can be seen in the examination of web service lifecycle. Further there has not been a mathematical theory or analysis of demand in e-commerce and web services. In what follows, we examine the mathematical foundation of demand in order to fill this gap and then use it to develop demand-driven web services.

We can analyze "demand" mathematically as follows. A man M demands something S provided by N. In other words, from a mathematical viewpoint, demand is a 3-ary relation that can be denoted as *Demand* (m, n, s). In the context of web services, we can explain *Demand* (m, n, s) as: a player m demands web service s provided by player n. For example, "service requestor r demands web service consultation c provided by service broker b" can be denoted as *Demand* (r, b, c). More generally, demand as a 3-ary relation can be denoted as: Let M, N, and S be a non-empty set respectively, $M = \{m_1, m_2 ... m_I\}$, $N = \{n_1, n_2 ..., n_J\}$, $S = \{s_1, s_2, ... s_K\}$, then any subset D^3 of $M \times N \times S$, $D^3 \subseteq M \times N \times S$, is a demand relation. In web services, N and M can denotes all the service requestors and all the service providers or brokers respectively. S represents all the web services provided on the web.

In business practice, this 3-ary demand relation is usually simplified as a binary relation D^2 or a unary relation D^1: For example, in B2C e-commerce, we only focus on: who demands what, that is, $D^2 \subseteq M \times S$ represents "customers *m* demands a good *s*," where *M* denotes all the service requestors or all the service providers. Further, in B2C e-commerce, we usually do not care about "who demands what" but only care about "what that are demanded", that is, $D^1 \subseteq S$ represents the good that is demanded. Therefore, from a demand's perspective, there are three different types of e-commerce: D^1 e-commerce, D^2 e-commerce and D^3 e-commerce.

- D^1 e-commerce only focuses on the goods that are transacted. Such an e-commerce is usually used for statistical analysis.
- D^2 e-commerce only focuses on the customer and the goods that are purchased by the customer or the seller, and that the goods that are sold. Therefore, D^2 e-commerce corresponds to a B2C e-commerce.
- D^3 e-commerce focuses on all the service providers, requestors, and goods that are transacted. Therefore, D^3 e-commerce is an organization that oversees the activities in web services or e-commerce.

In fact, taking into account the amount and payment associated with demand relation, we introduce demand functions respectively for D^1 e-commerce, D^2 e-commerce and D^3 e-commerce. For example, let *A* and *P* be non-empty sets, then any function $d^1 : S \to A \times P,\ d(s) = a \cdot p$, is a demand function taking into account the amount *A* of demand and the corresponding price *P* per a unit demand. For example, in D^1 e-commerce, a customer demands 100 textbooks on e-commerce, and the price is AUD$100.00 per textbook. Then, the corresponding demand function value is

$$d^1(book) = 100 \cdot 100 = 10000$$

where the customer and provider are technically ignored. This demand function represents the total price for the demanded 100 textbooks.

Similarly, in D^2 e-commerce, $d^2(David, book) = 100 \cdot 100 = 10000$ represents that David demands 100 textbooks on e-commerce with the price of AUD$10000.00, where the providers are technically ignored.

In D^3 e-commerce, $d^3(David, book) = 100 \cdot 100 = 10000$ represents that David demands 100 books on e-commerce provided by Amazon.com with the price of AUD$10000.00. This is a complete form for demand in a transactional web service.

From the above discussion, we can see that there is an inclusion relationship among D^1 e-commerce, D^2 e-commerce and D^3 e-commerce, as illustrated in Figure 2.

In demand-driven web services, we need a 3-ary demand relation taking into account the main players in web services (see the previous section). This demand relation can be illustrated in Demand relations in Web services.

In D^3 web services, demand is a 3-ary relation, that is, service requestors demands service brokers or providers to provide certain web services, and vice versa. However, what properties this 3-ary relation has in the context of web services remains open. Some stages in the web lifecycle may be absent since the demand disappears. There are also many situations resulting in demand cancellation.

In Table 1, the first column consists of a service provider, a service requestor, and a service broker.

Figure 2. Interrelationship among D^1, D^2, and D^3 e-commerce

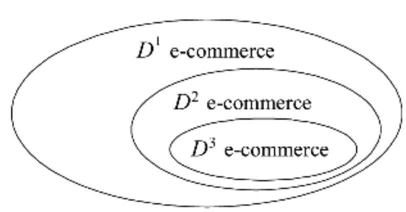


Table 1. Demand relations in web services

WS activities	Provider	Requestor	Broker
Provider	N/A	Binding, discovery, negotiation, invocation, billing, contract	discovery, recommendation, invocation, billing, contract
Requestor	QoS, description, representation, identification, search, match, discovery, negotiation, invocation, contract	N/A	Finding, consultation, personalization, recommendation, adaptation, mediation negotiation,
Broker	Publication, management search, match, Discovery, billing, contract	invocation, billing, contract	N/A

The first row lists a service provider, a service requestor, and a service broker. WS activities in the top left cell denote all the web service activities (or operations) in all the cell (m, n), where m an n denote rows and columns respectively. This implies that cell (m, n) contains the web service activities that are demanded. For example, cell (1, 2) contains web service binding, discovery, negotiation, invocation, billing, contract, all of which are demanded by the web service provider to the web requestor, where cell (2, 1) contains web service QoS, description, representation, identification, search, match, discovery, negotiation, invocation, contract, all of which are demanded by the web service requestor to the web service provider.

Even if the web service requestor demands the web service provider to represent web services (web service representation), the web service provider might not represent web services by himself. Instead, he may demand others to do so. In this way, a web service demand chain is formed in web services. For example, the web service requestor demands service consultation from a service broker, while the service broker demands web service representation and publication from a service provider. The service provider demands the most powerful web service tools from the ICT developer to realize the web service representation and publication, as shown in Figure 3. The extended form of a web service demand chain is a web service demand network in web services, just as there are supply chain networks in e-commerce (Chaffey, 2007).

It should be noted that supply chain management and demand chain management have been seriously studied in business, marketing and management (Chaffey, 2007, pp. 266-300; Rainbird, 2004; Walters, 2006), whereas demand chain and demand chain management have not drawn significant attention in e-business and e-commerce, to our knowledge. This situation might be changed in web services, because service customers' demands play a vital role in web services.

Figure 3. A demand chain in web services

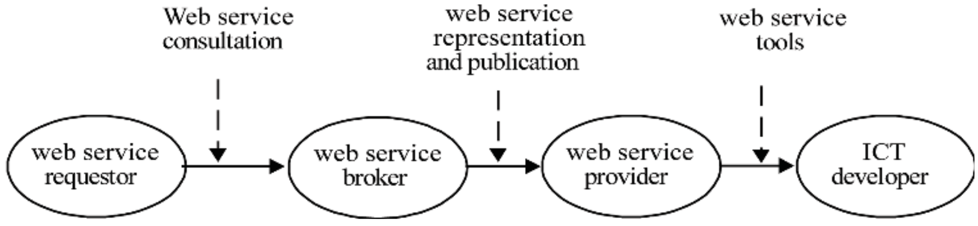
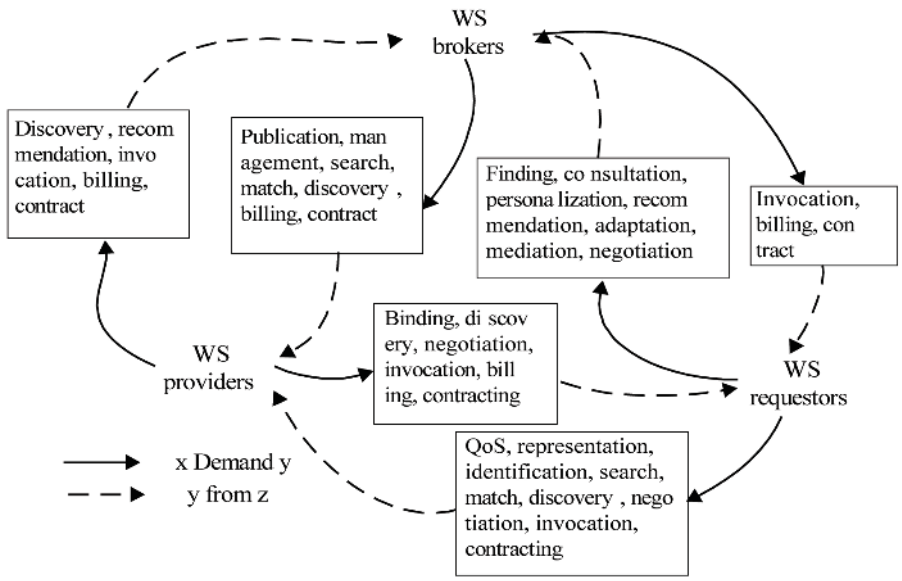


Figure 4. DWSOA: A demand-driven SOA for web services



What are the majority of customers' demands as well as their expectation and propensity in web services? and how do demand chain and demand network as well as demand network management impact on web services? How can we integrate demand chain analysis with the proposed mathematical analysis of demand? These problems remain open. We will address them in another article, and will not look at them any more because they are beyond the scope of this chapter.

DWSOA: A DEMAND DRIVEN ARCHITECTURE FOR WEB SERVICES

This section will review a few web service architectures and then provide a demand-driven architecture for web services.

We have briefly mentioned different service-oriented architectures (SOA) for web services in the section "Background". In fact, web service architectures have been a research topic for engineering web services (Alonso, et al, 2004). There have been a large number of web service architectures proposed in the past years. For example, Wu and Chang (2005) provide a conceptual architecture of web services for service ecosystems. Garcia and de Toledo (2006) propose an extended web service architecture providing QoS management for web services. However, the existing web service architectures are mainly from the perspective of developers rather than from a demand perspective. Based on the above demand analysis for web services, we can propose a demand-driven SOA for web services (DWSOA), as shown in Figure 4. Note that in Figure 4, x and z denote either WS providers, brokers or requestors, y denotes WS activities.

In the DWSOA, WS (web service) providers demand WS discovery, recommendation, invocation, billing, and contract from WS brokers; WS brokers demand WS invocation, billing, contract from WS requestors; WS requestors demand WS finding, consultation, personalization, recommendation, adaptation, mediation, negotiation from WS brokers; and so on. From the viewpoint of previously mentioned demand relation in web services, the above-mentioned demand relations are 3-ary, and corresponds to

1. Demand (WS providers, WS brokers, WS activities) = {WS discovery, recommendation, invocation, billing, and contract};
2. Demand (WS brokers, WS requestors, WS activities) = {WS invocation, billing, contract};
3. Demand (WS requestors, WS brokers, WS activities) = {WS finding, consultation, personalization, recommendation, adaptation, mediation, negotiation};

and so on. It should be noted that there are inter-demands among the WS providers, brokers and requestors in the DWSOA, although the demands among them are not symmetric. In other words, the service requestor demands the service consultation from the service broker, whereas the service broker does not demand service consultation from the service requestor.

We do not believe that all activities in the web services have been covered in the DSWOA. For example, WS engagement, enactment, and management (Burstein, et al, 2005) are not included in the DSWOA. The key idea behind it is that we expose the demand relationship among WS providers, brokers and requestors. This is the basis for demand-driven web service lifecycles, which will be discussed in later sections.

WEB SERVICE LIFECYCLE

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

From the perspective of computer science, the notion of lifecycle originated from software engineering (Pressman, 2001). It describes the life of a software product (development) from its conception, to its implementation, delivery, use, and maintenance (Pfleeger & Atlee, 2006). 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 large number of attempts to address web service lifecycle in the web service community, as discussed in the section "Background". Further, Glatard, et al. (2008) examine a SOA enabling dynamic service grouping for optimizing distributed workflow execution. Leymann (2003) 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 (2003) proposes a semantic web process lifecycle that consists of web service description (annotation), discovery, composition and execution or orchestration. Wu and Chang (2005) consider service discovery, service invocation and service composition as the whole lifecycle of web services. Zhang and Jeckle (2003) 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. Kwon (2003) proposes a lifecycle of web services consisting of four fundamental steps: web service identification, creation, use and maintenance. Tsalgatidou and Pilioura (2002) 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 (Gottschalk, et al, 2000), 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 requestor'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 requestors, providers, and brokers in detail. Some 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 (W3C, 2004). Sun Microsystems considers the lifecycle of web services consisting of four stages: design/build, test, deploy/execute, and manage (Sun Microsystems, 2003), which can be considered a model for web service developers.

Based on the above analysis, we can find that there are at least the following activities of web services that have been mentioned in the reviewed web service lifecycles, all of these can be classified into two categories: development-oriented web service activities and business-oriented web service activities.

- Development-oriented web service activities: Web service modeling, representation, design/building, test, publishing, unpublishing, deployment, execution, re-execution, orchestration, collaboration, monitoring, analytical control, maintenance, and management,
- Business-oriented web service activities: Web service creation, identification, description (annotation), publishing, finding, discovery, use, invocation and binding (Gottschalk, et al, 2000), adaptation, composition, security, brokering, recommendation, reliability, billing, monitoring, transaction handling and contracting.

This classification does not produce a crisp mathematical partition, because, some activities such as web service representation, management, and creation can be considered in both kinds of web service activities. However, such a classification reflects the fact that some existing web service lifecycles are proposed from the implementation perspective of the web service developers (Burstein, et al, 2006), whereas other web service lifecycles are proposed from the perspective of business. Generally speaking, web service providers pay more attention to development-demanded web service activities than web service brokers and requestors, whereas web service brokers and requestors pay more attention to Business-demanded web service activities than web service providers.

From a market perspective, web services mainly consist of three kinds of players: Service providers, service requestors and service brokers (Tang, et al, 2007). Different players require different web service lifecycles. Therefore, what is a web service lifecycle from the demand viewpoint of web service providers, brokers and requestors respectively? How many stages does a web service lifecycle consist of? These problems are interesting for examining demand-driven web services.

Further, demand is an important factor for market and economy development (Chaffey, 2007, p. 150). The decrease of demand is an implication for economic recession. Different players generally have different demands for web services, different demands have also different web service lifecycles. Therefore, what is the demand-driven web service lifecycle from the viewpoint of web service providers, brokers and requestors 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 (Guruge, 2004) such as searching information using Google and doing business online. At the same time, he or she demands more and more 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 players in web services in what follows, that is, we will look at demand-driven web service lifecycles for web service providers, requestors and brokers respectively.

A PROVIDER'S DEMAND DRIVEN WEB SERVICE LIFECYCLE

In web services, a web service provider usually demands a web service requestor to commit some web service activities and also demands a web service broker to commit some web service activities. Therefore, based on the above demand analysis for web services, a provider's demand-driven web service lifecycle mainly consists of web service finding, identification (Kwon, 2003), description/representation (Burstein, et al, 2005), creation (Kwon, 2003; Tang, et al, 2007), discovery, composition (Limthanmaphon & Zhang, 2003; Tang, et al, 2007; Burstein, et al, 2005), recommendation (Sun & Lau, 2007), negotiation (Hung, et al, 2004; Burstein, et al, 2005), invocation (Burstein, et al, 2005), contracting, use and reuse (Kwon, 2003), execution or orchestration, management and monitoring (Dustar & Schreiner, 2005), maintenance (Kwon, 2003), billing and security (Tang, et al, 2007).

Web service identification aims to identify appropriate services (Ladner, 2008). Web service invocation is to invoke the discovered web service interface. Web services are published to the intranet or the Internet repositories for potential users to locate (Tang, et al, 2007). Web service unpublishing is sometimes no longer available or needed, or it has to be updated to satisfy new requirements (Tang, et al, 2007)

Web service composition primarily concerns requests of web service users that cannot be satisfied by any available web service (Bucchiarone & Gnesi, 2006; Narendra & Orriens, 2006). One form of simple web service composition is to combine a set of available web services to obtain a composite service that might be recommended to the users. In other words, web service composition is a process in which a single web service requested from a service requestor can be satisfied by an aggregation of different services provided by several independent web services providers. More strictly, web service composition refers to the process of creating customized or personalized services from existing services by a process of dynamic discovery, integration and execution of those services in order to satisfy user requirements (Wang, et al, 2008). Web service composition is a key challenge to manage collaboration among web services (Limthanmaphon & Zhang, 2003). 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 (Dustdar & Schreiner, 2005).

There are many techniques existing for web service composition. For example, Tang, et al (2007) propose an automatic web service composition method taking into account both services' input/output type compatibility and behavioral constraint compatibility. Cheng, et al. (2006) use case-based reasoning (CBR) to support web service composition. Further, Dustdar and Schreiner (2005) discuss the urgent need for service composition and the required technologies to perform service composition as well as present several different composition strategies. Web service composition is becoming an important topic for service computing, because composing web services to meet the requirement of the web service requestor is the most important issue for web service providers and brokers.

REQUESTOR'S DEMAND DRIVEN WEB SERVICE LIFECYCLE

In web services, a web service requestor usually demands a web service provider and a web service broker to commit some web service activities respectively. Therefore, based on the above demand analysis for web services, a requestor's demand-driven web service lifecycle mainly consists of web service consultation, creation (Burstein, et al, 2005), representation, search (Ladner, 2008), matching (Ladner, 2008), finding, discovery (Tang, et al, 2007; Burstein, et al, 2005), identification (Burstein, et al, 2005), composition, mediation (Ladner, 2008; Burstein, et al, 2005), personalization, adaptation, negotiation (Gottschalk, et al, 2000), evaluation (Burstein, et al, 2005) and recommendation, QoS (Burstein, et al, 2005), invocation (Burstein, et al, 2005), contracting (Burstein, et al, 2005).

Web service discovery is a process of finding the most appropriate web service needed by a web service requestor (Singh & Huhns, 2005; Burstein, et al, 2005). It identifies a new web service and detects an update to a previously discovered web service (Ladner, 2008). Services may be searched, matched, and discovered by service requestors by specifying search criteria and then be invoked (Dustdar & Schreiner, 2005; Tang, et al, 2007). Service invocation is restricted to authorized users (Dustdar & Schreiner, 2005). There have been a variety of techniques and approaches developed for web service discovery. For example, OWL-S (an OWL-based Web service ontology 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 (Ladner, 2008).

Web service mediation aims to mediate the request of web service from the web service requestor. 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 web service (Yao, et al, 2006). Through negotiation, web service requestors 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 requestor might not need to know how the web services are retrieved, discovered and composed internally. Therefore, search, matching, and composition might be less important for a web service requestor.

A BROKER'S DEMAND DRIVEN WEB SERVICE LIFECYCLE

Brokering is the general act of mediating between requestors and providers in order to match service requestor's needs and providers' offerings. It is a more complicated activity than discovery (Tang, et al, 2007). A broker should enable universal service-to-service interaction, negotiation, bidding and selection of the highest quality of service (QoS) (Singh & Huhns, 2005, pp. 345-346). Brokering is supported by HP web services platform as a HP web intelligent broker (Tsalgatidou & Pilioura, 2002). 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.

In web services, a web service broker usually demands a web service provider and a web service requestor to undertake some web service activities respectively. Therefore, based on the above demand analysis for web services, a broker's demand-driven web service lifecycle mainly consists of web service consultation, publication, search, matching (Burstein, et al, 2005), discovery, personalization, adaptation, composition, negotiation, recommendation, management, invocation, contracting and billing.

We propose web service consultation as the start of the broker's demand-driven web service lifecycle, because the web service requestor 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, by using a search tool/engine such as google.com and beidu.com. During the web service search, the web service broker uses any techniques of web service matching such as CBR (Sun & Finnie, 2004). After discovering a number of web services, the web service broker can select one of them to recommend it to the web service requestor. If the requestor 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 aims at helping web service requestors in selecting web services more suitable to their needs. 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 (Kwon, 2003). Recommender systems have been studied and developed in e-commerce, e-business and multiagent systems (Sun & Finnie, 2004). Sun and Lau has examined case based web service recommendation based on the analysis of customer experience and experience-based reasoning (Sun & Lau, 2007). 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 requestors have different preferences and expectations. Therefore, a web service broker has to personalize web services in order to meet the requirement of the web service requestor satisfactorily. It is necessary to compose web services based on the requirement of requestors in order to personalize the web service. At the same time, web service composition allows web service brokers to create a composite web service for requestors rapidly (Tang, et al, 2007).

Billing concerns service brokers and service providers (Tang, et al, 2007). 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.

A UNIFIED PERSPECTIVE ON DEMAND DRIVEN WEB SERVICE LIFECYCLES

Based on the above discussion, the stages involved in the demand-driven web service lifecycle for web service providers, requestors and brokers can be summarized in Figure 5. Some of the detailed activities have not been listed in the table because of space limitations. From Figure 5, we can intuitively find that service requestors and brokers are the dominant force for developing web services, 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 players: service providers, brokers, and requestors. This means that they share some common web service activities. However, different players in web services demand the same activity in a different way. For example, the service provider demands "web services search" also means that s/he asks web services developers or her/his technology agents to provide efficient web services search function for his or her business. On the other hand, the service requestor demands "web services 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.

Finding, search and matching are not unique activities or operations related to web services,

Figure 5. Demand driven web service lifecycles: A unified perspective

Players in web services	identification	representation	creation/publishing	search/matching	finding/discovery	consultation	personalization	composition	recommendation	adaptation	mediation	negotiation	invocation	binding	billing	contract	security	maintenance	evaluation/QoS	management
Provider	X	X	X		X			X	X		X	X	X	X	X	X	X	X		X
requestor	X	X	X	X	X	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	X		X	X		X		X

because they are also involved in database and case based reasoning (CBR). For example, Google uses search and matching to provide web services. In fact, search can be considered the most common demand for everyone who accesses the Internet or the web. Adaptation, retrieval, classification (Ladner, 2008), use/reuse (Kwon, 2003), retention or feedback are not unique activities related to web services either, because they are also stages of CBR cycle (Sun & Finnie, 2004). Web service invocation, binding, billing, contract (Tang, et al, 2007) can be considered as 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. In a more general sense, all the above-mentioned activities of web services can be considered as a demand from web services to all the stakeholders of web services. This demand asks web service developers to provide services with high QoS and advanced tools for all the activities of web services. Therefore, these services can be considered as meta-web services and we will examine the hierarchy of demands in web services in future work.

It should be noted that the activities in web services should be classified in a hierarchical way (main services and subservices). For example, identification, finding, search and matching can be subactivities of web service discovery (Burstein, et al, 2005). Dan, et al. (2004) argues that the subactivities of web service contract consist of offering creation, customer order and negotiation, monitoring, billing and reporting. Burstein, et al. (2005) examines the subactivities of web service discovery, engagement, enactment and management. Then we can examine the hierarchical structure of activities or interoperations in web services assuming that publish, find, and bind are the fundamental activities of web services.

FUTURE RESEARCH DIRECTIONS

Understanding the demand of stakeholders of web services is a critical factor for further development of web services. This chapter only focuses on demand-driven web services from a demand perspective of the main players in web services: service providers, brokers, and requestors. In fact, these demands from the web service providers, brokers, and requestors not only require to be met from themselves but also from more stakeholders of web services, in particular, the web service developers with the strong background of information communication technology (ICT). They will provide technological solution for the main activities of web service lifecycle such as web service description and discovery (Garcia & Toledo, 2006), composition (Papazoglou, et al, 2006), billing, contracting. For example, the engineering of web service composition and recommendation are a research direction (Papazoglou, et al, 2006). In future work, we will explore implementation issues for engineering of web service composition and recommendation.

The proposed demand-driven web service lifecycles are still in a linear form. Providing other forms of demand-driven web service lifecycle is also a research direction. In future work, we will develop demand-driven models for web service lifecycle in a spiral and iterative way with corresponding diagrams, as done in software engineering.

Applying intelligent techniques to web services and automating the process stages in the demand-driven web service lifecycle is another research direction (Petrie & Genesereth, 2003). In future work, we will integrate web service discovery, composition and recommendation using soft case based reasoning.

Demand is an important concept in economics. However, there is less attention in web services. In future work, we will investigate the computing basis of demand and then improve the above-mentioned web service lifecycle. For example, we will examine demand as a 4-ary relation (seller, buyer, service, price) from a mathematical and business viewpoint and propose a hierarchical structure for demand-driven e-commerce and demand-driven web services. We will also further analyse the demands between service players, and provide the related instruction/guidance for web service design and development in order to develop demand-driven framework for Web services.

CONCLUSION

This chapter first looked at main players in web services, provided a mathematical analysis of demand in web services. It also examined the demand relationship among service providers, brokers and requestors in web services and the demand chain in web services. Then the chapter reviewed web service architectures and provided a demand-driven architecture for web services (DWSOA). It also reviewed web service lifecycles, proposed the demand-driven web service lifecycle for the main players in web services respectively and then discussed the demand-driven web service lifecycles in a unified way. The proposed approach in this chapter can facilitate the engineering and management of web services, and the research and development of web services, e-services, service intelligence and service science. In the future work, besides above-mentioned future research directions, we will develop demand-driven framework for Web services by extending Table 2 to include as many stages or activities of web services life cycle as possible. We will also use the proposed approaches to study business models further and try to apply them to Web services design.

ACKNOWLEDGMENT

This research is partially supported by the Ministry of Education Hebei, China under a key research Grant No. ZH200815 and other research grants provided by the College of Mathematics and Information Science, Hebei Normal University, China.

REFERENCES

W3C (2004). Web service management: Service life cycle. Retrieved December 26, 2008, from http://www.w3.org/TR/2004/NOTE-web servicelc-20040211/

Alonso, G., Casati, F., Kuno, H., & Machiraju, V. (2004). *Web Services: Concepts, Architectures and Applications*. Berlin: Springer-Verlag.

Atkinson, M., DeRoure, D., Alistair Dunlop, A., et al. (2004). Web service grids: An evolutionary approach. *UK e-Science Technical Report Series*. Retrieved July 6, 2009 from http://www.nesc.ac.uk/technical_papers/ UKeS-2004-05.pdf

Benatallah, B., Reza, H., & Nezhad, M. (2006, July/August). A model-driven framework for web services life-cycle management. *IEEE Internet Computing*, 55–63. doi:10.1109/MIC.2006.87

Bucchiarone, A., & Gnesi, S. (2006). A survey on services composition languages and models. In *Proceedings of the International Workshop on Web Services – Modeling and Testing* (Web service-MaTe 2006), Palermo, Sicily, June 9 (pp.51-63).

Burstien, M. (2005). A semantic Web services architecture. *IEEE Internet Computing*, *9*(5), 72–81. doi:10.1109/MIC.2005.96

Chaffey, D. (2007). *E-Business and E-Commerce Management* (3rd ed.). Harlow, UK: Prentice Hall.

Cheng, R., & Su, S. Yang, F., & Li, Y. (2006). Using case-based reasoning to support web service composition. In V.N. Alexandrov et al. (Eds.), *ICCS 2006, Part IV* (LNCS 3994, pp. pp. 87-94). Berlin, Heidelberg: Springer-Verlag.

Dan, A., Kuebler, D., & Davis, D. (2004). Web services on demand: WSLA-driven automated management. *IBM Systems Journal*, *43*(1), 136–158. doi:10.1147/sj.431.0136

Deitel, H. M., Deitel, P. J., DuWadt, B., & Trees, L. K. (2004). *Web Services: A technical Introduction*. Upper Saddle River, NJ: Prentice Hall.

Dustdar, S., & Schreiner, W. (2005). A survey on web services composition. *International Journal Web and Grid Services*, *1*(1), 1–30. doi:10.1504/IJWGS.2005.007545

Erl, T. (2006). *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*. Upper Saddle River, NJ: Prentice Hall.

Ferris, C., & Farrell, J. (2003). What are web services? *Communications of the ACM*, *46*(6), 31. doi:10.1145/777313.777335

Garcia, D. Z. G., & de Toledo, M. B. F. (2006). A web service architecture providing QoS management. In *Proceedings of the Fourth Latin American Web Congress* (LA-Web '06) (pp. 189-198).

Gisolfi, D. (2001). Web services architect: Part 1: An introduction to dynamic e-business. Retrieved July 8, 2009 from http://www.ibm.com/developerworks/webservices/library/ws-arc1

Glatard, T., Montagnat, J., Emsellem, D., & Lingrand, D. (2008). A service-oriented architecture enabling dynamic service grouping for optimizing distributed workflow execution. *Future Generation Computer Systems*, *24*(7), 720–730. doi:10.1016/j.future.2008.02.011

Gottschalk, K., et al. (2000). Web Services architecture overview. Retrieved July 15, 2009 from http://www.ibm.com/developerworks/webservices/library/w-ovr

Guruge, A. (2004). *Web Services: Theory and Practice*. Amsterdam: Elsevier Inc.

Henderson-Sellers, B., & Giorgini, P. (Eds.). (2005). *Agent-Oriented Methodologies*. Hershey, PA: Idea Group Publishing.

Hoffman, K. D. (2003). Marketing + MIS = E-Services. *Communications of the ACM*, *46*(6), 53–55. doi:10.1145/777313.777340

Hung, P. C. K., Li, H., & Jeng, J. J. (2004). Web service negotiation: An overview of research issues. In *Proc of 37th Hawaii Intl Conf on System Sciences* (pp. 1-10).

ICWS. (2009). http://conferences.computer.org/icws/2009/.

Jackson, J., & Mclver, R. (2004). *Microeconomics* (7th ed.). Australia: McGraw-Hill.

Kreger, H. (2001). Web services conceptual architecture (WSCA 1.0), IBM Software Group. Retrieved March 28, 2009, from http://www.cs.uoi.gr/~zarras/mdw-ws/WebServicesConceptualArchitectu2.pdf

Kwon, O. B. (2003). Meta web service: building web-based open decision support system based on web services. *Expert Systems with Applications*, *24*, 375–389. doi:10.1016/S0957-4174(02)00187-2

Ladner, R. (2008). Soft computing techniques for web service brokering. *Soft Computing*, *12*, 1089–1098. doi:10.1007/s00500-008-0277-0

Leymann, F. (2003). Web Services: Distributed Applications without Limits: An Outline. In *Proceedings Database Systems for Business, Technology and Web BTW*, Leipzig, Germany, Feb 26 – 28, 2003, Springer. Retrieved April 1, 2208, from http://doesen0.informatik.uni-leipzig.de/proceedings/paper/keynote-leymann.pdf

Limthanmaphon, B., & Zhang, Y. (2003). Web service composition with case-based reasoning. In *ACM Intl Conf Proc Series (Vol. 143), Proc. 14th Australasian Database Conf*, Adelaide, Australia (pp. 201-208).

Miller, G. (2005). NET vs. J2EE. *Communications of the ACM*, *48*(7), 64–67.

Narendra, N. C., & Orriens, B. (2006). Requirements-driven modeling of the web service execution and adaptation lifecycle. In S. Madria et al. (Eds.), *ICDCIT* (LNCS 4317, pp. 314-324).

Papazoglou, M. P. (2003). Service -Oriented Computing: Concepts, Characteristics and Directions. In *Proceedings of 4th Intl Conf on Web Information Systems Engineering* (WISE2003) (pp. 3-12).

Papazoglou, M.P., Traverso, P, Dustdar, S. & et al. (2006). Service-oriented computing research roadmap. Retrieved April 4, 2009, from http:drops.dagstuhl.de/opus/volltexte/2006/524/

Petrie, C., Genesereth, M., et al. (2003). Adding AI to web services. In L. van Elst, V. Dignum, & A. Abecker (Eds.), *AMKM 2003* (LNAI 2926, pp. 322-338).

Pfleeger, S. L., & Atlee, J. M. (2006). *Software Engineering: Theory and Practice* (3rd ed.). Beijing: Pearson Education, Inc.

Pressman, R. S. (2001). *Software Engineering: A Practitioner's Approach* (5th ed.). Boston: McGraw-Hill.

Rainbird, M. (2004). Demand and supply chains: The value catalyst. *International Journal of Physical Distribution & Logistics Management*, *34*(3/4), 230–250. doi:10.1108/09600030410533565

Rust, R. T., & Kannan, P. K. (2003). E-service: A new paradigm for business in the electronic environment. *Communications of the ACM*, *46*(6), 37–42. doi:10.1145/777313.777336

Schneider, G. P. (2003). *Electronic Commerce* (4th ed.). Boston, MA: Thomson Course Technology.

Sheth, A. (2003). 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.

Singh, M. P., & Huhns, M. N. (2005). *Service-oriented Computing: Semantics, Processes, and Agents*. Chichester: John Wiley & Sons, Ltd.

Sun, Z., & Finnie, G. (2004). *Intelligent Techniques in E-Commerce: A Case-based Reasoning Perspective*. Berlin, Heidelberg: Springer-Verlag.

Sun, Z., & Finnie, G. (2005). A unified logical model for CBR-based e-commerce systems. *International Journal of Intelligent Systems*, *20*(1), 29–46. doi:10.1002/int.20052

Sun, Z., & Lau, S. K. (2007). Customer experience management in e-services. In J. Lu, D. Ruan, & G> Zhang (Eds.), *E-Service Intelligence: Methodologies*, Technologies and Applications (pp. 365-388). Berlin, Heidelberg: Springer-Verlag.

Sun Microsystems. (2003). Web services life cycle: Managing enterprise Web services, White Paper. Retrieved December 18, 2008, from http://www.sun.com/software

Talia, D. (2002). The open grid services architecture: Where the grid meets the web. *IEEE Internet Computing*, *6*(6), 67–71. doi:10.1109/MIC.2002.1067739

Tang, X. F., Jiang, C. J., Ding, Z. J., & Wang, C. (2007). A Petri net-based semantic web services automatic composition method. [in Chinese]. *Journal of Software*, *18*(12), 2991–3000.

Tsalgatidou, A., & Pilioura, T. (2002). An overview of standards and related technology in web services. *Distributed and Parallel Databases*, *12*, 135–162. doi:10.1023/A:1016599017660

Wang, M., Cheung, W. K., Liu, J., Xie, X., & Lou, Z. (2006). E-service/process composition through multi-agent constraint management. In *Intl Conf on Business Process Management* (BPM 2006), Vienna, Austria (LNCS 4102, pp. 274-289).

Wang, M., Liu, J., Wang, H., Cheung, W. K., & Xie, X. (2008). On-demand e-supply chain integration: A multi-agent constraint-based approach. *Expert Systems with Applications*, *34*(4), 2683–2692. doi:10.1016/j.eswa.2007.05.041

Waters, D. (2006). Demand chain effectiveness-supply chain efficiencies: A role for enterprise information management. *Journal of Enterprise Information Management*, *19*(3), 246–261. doi:10.1108/17410390610658441

Weiss, G. (Ed.). (1999). *Multiagent Systems: A modern approach to Distributed Artificial Intelligence*. Cambridge, MA: MIT Press.

Wiki (2009). Retrieved April 2, 2009, from http://en.wikipedia.org/wiki/Demand_(economics).

Wilkinson, N. (2005). *Managerial Economics: A Problem-Solving Approach*. Cambridge: Cambridge University Press.

Wu, C., & Chang, E. (2005). A conceptual architecture of distributed web services for service ecosystems. In S. Dascalu (Ed.), *18th International Conf on Computer Applications in Industry and Engineering* (CAINE 2005) (pp. 209-214).

Yao, Y., Yang, F., & Su, S. (2006). Flexible decision making in web services negotiation. In J. Euzenat & J. Domingue (Eds.), *AIMSA 2006* (LNAI 4183, pp. 108-117).

Zhang, L. J., & Jeckle, M. (2003). The next big thing: Web services composition. In M. Jeckle & L.J. Zhang (Eds.), *ICWS-Europe* (LNCS 2853, pp. 1-10).

ADDITIONAL READING

Bell, M. (2008). *Service-Oriented Modeling (SOA): Service Analysis, Design, and Architecture*. Chichester, UK: John Wiley and Sons Ltd.

Breu, R. Breu1, M., Hafner, M., & Nowak, A. (2005). Web service engineering advancing a new software engineering discipline. In D. Lowe & M. Gaedke (Eds.), *ICWE 2005* (LNCS 3579, pp. 8-18).

Brown, A., Johnston, S., & Kelly, K. (2002). Using Service-Oriented Architecture and Component-based Development to Build Web Services. Retrieved 10 July 2009, from http://www.ibm.com/developerworks/rational/library/510.html#download

Chen, L., & Tao, F. (2008). An intelligent recommender system for web resource discovery and selection. In Ruan, D., Hardeman, F., & van der Meer, K. (Eds.), *Intelligent Decision and Policy Making Support Systems* (pp. 113–140). Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-78308-4_7

Cheng, M.-Y., Tsai, H.-C., & Chiu, Y.-H. (2009). Fuzzy case-based reasoning for coping with construction disputes. *Expert Systems with Applications*, *36*(2), 4106–4113. doi:10.1016/j.eswa.2008.03.025

Cheng, R., & Su, S. Yang, F., & Li, Y. (2006). Using case-based reasoning to support web service composition. In V.N. Alexandrov et al. (eds.), *ICCS 2006, Part IV* (LNCS 3994, pp. 87-94). Berlin Heidelberg: Springer-Verlag.

Chung, J. Y., Lin, K. J., & Mathieu, R. G. (2003). Web services computing: Advancing software interoperability. *Computer*, *36*(10), 35–37. doi:10.1109/MC.2003.1236469

Costantini, S. (2008). Agents and Web Services. Retrieved April 04, 2009, from http://www.cs.nmsu.edu/~epontell/backbone/aug08/content/Articles/sadri2/paper.pdf

Han, W., Xingdong Shi, X., & Chen, R. (2008). Process-context aware matchmaking for web service composition. *Journal of Network and Computer Applications*, *31*(4), 559–576. doi:10.1016/j.jnca.2007.11.008

Jeong, B., Cho, H., & Lee, C. (2009). On the functional quality of service (FQoS) to discover and compose interoperable web services. *Expert Systems with Applications*, *36*(3), 5411–5418. doi:10.1016/j.eswa.2008.06.087

Küster, U., König-Ries, B., Stern, M., & Klein, M. (2007). DIANE: An integrated approach to automated service discovery, matchmaking and composition. In *Intl Conf on World Wide Web* (WWW2007), May 8–12, 2007, Alberta, Canada (pp. 1033-1041).

Lau, R. Y. K. (2007). Towards a web services and intelligent agents-based negotiation system for B2B eCommerce. *Electronic Commerce Research and Applications*, *6*(3), 260–273. doi:10.1016/j.elerap.2006.06.007

Lawler, J. P. (2007). *Service-Oriented Architecture: SOA Strategy, Methodology, and Technology*. Hoboken: Taylor & Francis Ltd. doi:10.1201/9781420045017

Lu, J., Ruan, D., & Zhang, G. (Eds.). (2006). *E-Service Intelligence*. Berlin, Heidelberg: Springer Verlag.

Madhusudan, T., & Uttamsingh, N. (2006). A declarative approach to composing web services in dynamic environments. *Journal of Decision Support Systems*, *41*(2), 325–357. doi:10.1016/j.dss.2004.07.003

Park, C.-S., & Park, S. (2008). Efficient execution of composite Web services exchanging intensional data. *Journal of Information Science*, *178*(2), 317–339. doi:10.1016/j.ins.2007.08.021

Psaila, G., & Wagner, R. R. (Eds.). (2008). *E-Commerce and Web Technologies*: *9th International Conference, EC-Web 2008*, Turin, Italy, September 3-4, 2008, Proceedings (LNCS). Berlin Heidelberg: Springer Verlag.

Richter, M. M. (2009). The search for knowledge, contexts, and case-based reasoning. *Engineering Applications of Artificial Intelligence*, *22*(1), 3–9. doi:10.1016/j.engappai.2008.04.021

Rust, R. T., & Kannan, P. K. (Eds.). (2002). *E-Service: New Directions in Theory and Practice*. Armonk, NY: M.E. Sharpe.

Sahai, A., & Graupner, S. (2005). *Web Services in the Enterprise: Concepts, Standards, Solutions, and Management (Network and Systems Management)*. New York: Springer.

Song, R., Korba, L., & Yee, G. (Eds.). (2007). *Trust in E-services: Technologies, Practices and Challenges*. Hershey, PA: IGI Global.

Talukder, A. K., & Yavagal, R. R. (2007). *Mobile Computing: Technology, Applications and Service Creation*. New York: McGraw-Hill Inc.

Tatemura, J., & Hsiung, W. P. (2006). Web service decomposition: Edge computing architecture for cache-friendly e-commerce applications. *Electronic Commerce Research and Applications*, *5*(1), 57–65. doi:10.1016/j.elerap.2005.08.001

Ting, I.-H., & Wu, H. J. (Eds.). (2009). *Web Mining Applications in E-commerce and E-services*. Berlin, Heidelberg: Springer Verlag. doi:10.1007/978-3-540-88081-3

Web Services Activity. (n.d.). Retrieved April 3, 2009, from http://www.w3.org/2002/ws/arch/.

Wooldridge, M. (2002). *An Introduction to Multiagent Systems*. Chichester, UK: John Wiley & Sons Ltd.

Zhao, J. L., & Cheng, H. K. (2005). Web services and process management: a union of convenience or a new area of research? *Journal of Decision Support Systems*, *40*(1), 1–8. doi:10.1016/j.dss.2004.04.002

KEY TERMS AND DEFINITIONS

Web Services: Generally speaking, web services are all the services available on the Web or the Internet from a business perspective. The first web services were information sources (Schneider, 2003). From a technological perspective, web services are Internet-based application components published using standard interface description languages and universally available via uniform communication protocols. This definition is currently used in the web service community.

Web Service Discovery: The process of searching, matching a machine-processable description of a Web service. It aims to find appropriate web services that meet the requirement of the customers.

Web Service Lifecycle (WSLC): It 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. Many activities are included in a WSLC such as web service discovery, composition, recommendation and management.

Demand Theory: A part of microeconomics. It examines demand curves, demand equations, demand analysis, demand chain, impact factors on demand, demand estimation and so on. Demand analysis assesses current and projected demand for e-commerce services amongst existing and potential customer segments.

Service Computing: A research field about service science, science intelligence, service technology, service engineering, service management, and service applications. It is the most general representation form of studying service in computing discipline. Service computing and service-oriented computing are used interchangeably.

Web Service Architecture: A Web service architecture is a high level description for web services, which is free of concrete implementation of a web service system.

A Web Service Demand Chain: A chain linking players related to web services, similar to supply chain in e-commerce. For example, the web service requestor demands service consultation from service broker, while the service broker demands web service representation and publication from service provider. The service provider demands the most powerful web service tools from the ICT developer to realize the web service representation and publication.

Multiagent Systems: An intelligent system consisting of many intelligent agents. An intelligent agent can be considered as a counterpart of a human agent in intelligent systems.

Intelligent System: A system that can imitate, automate some intelligent behaviors of human being. Expert systems and knowledge based systems are examples of intelligent systems. Currently intelligent systems is a discipline that studies the intelligent behaviors and their implementations as well as impacts on human society.