

## Auction-Based Pricing Model for Web Service Providers

Jia Zhang<sup>\*1</sup>, Ning Zhang<sup>\*\*</sup>, Liang-Jie Zhang<sup>\*\*\*</sup>

<sup>\*</sup>Department of Computer Science, Northern Illinois University, DeKalb, IL 60173

<sup>\*\*</sup>Department of Policy Analysis and Management, Cornell University, Ithaca, NY 14850

<sup>\*\*\*</sup>IBM T.J. Watson Research, Hawthorne, NY 10532

<sup>\*</sup>jjazhang@cs.niu.edu, <sup>\*\*</sup>nz33@cornell.edu, <sup>\*\*\*</sup>zhanglj@us.ibm.com

### ABSTRACT:

Applying auctions to Web services selection and invocation calls for examination due to the unique features of Web services, such as interoperable machine-to-machine interactions and re-entenable bargaining services. In this paper we propose a formal model for Web services-based auctions. Examining one-sided sealed auction type, we prove mathematically that service requestors' risk preferences could lead to different pricing strategies for service providers towards higher profit. We argue that Service Level Agreement (SLA) documents can be used to analyze service requestors' preferences. On top of WS-Agreement, we propose a basic service requestor risk preference elicitation algorithm, as well as a historical data-based service requestor risk preference prediction model. Guidelines are provided to iteratively approach the learning rate of the proposed risk preference prediction model. The methods and techniques presented in this paper can be reused to investigate and examine more facades of services-oriented auctions, towards establishing a new research realm on comprehensive services-oriented auctions.

### KEY WORDS:

*Services-oriented auction, Web services, Service Level Agreement*

## INTRODUCTION

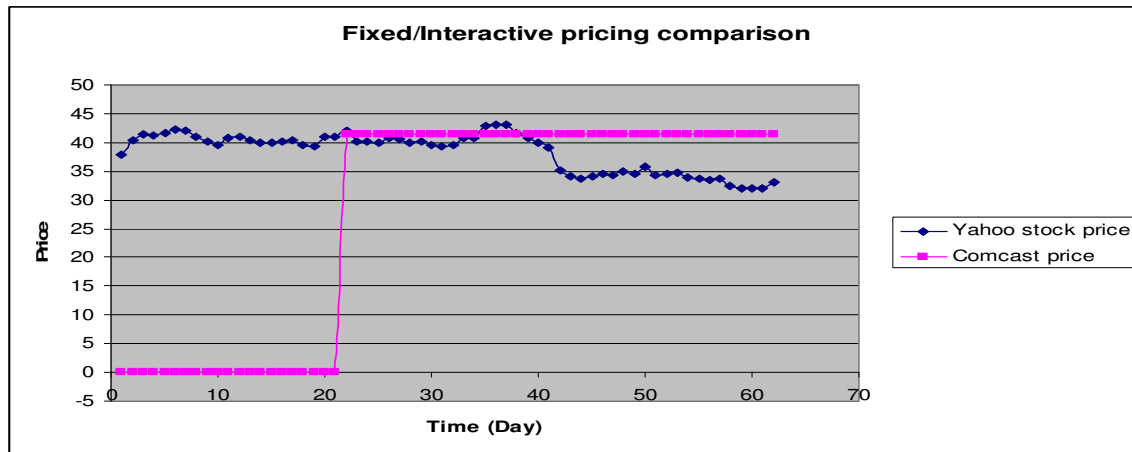
The paradigm of Web services has opened a new era for business service providers. This new model not only allows easier management and maintenance for provider-hosting services, but also creates a lot more potential business opportunities for service providers. Gartner Group, a leading industry analyst firm, predicted that by 2008 more than 60 percent of businesses would adopt Web services and transform into new types of enterprises (Gartner 2003).

As for traditional server providers, how to establish appropriate pricing models to pursue the highest profits is an essential concern for Web service providers. To date there are two major pricing models in the field of Web services that are derived from traditional business: periodic pricing and fluctuant pricing. Periodic pricing means that a service provider predefines a fixed price for a period of time, when every service requestor subjects to the same price value. In recent years, in order to attract more customers, many service providers offer a variant of the periodic pricing model, called staging price model. As shown in Figure 1, Comcast, a cable company, offers a \$0 first-month trial benefit for new customers, and a fixed \$40/month service fee afterwards. Fluctuant pricing means that the price of a service is ever changing based on marketing situations. Stock pricing is a typical example of the fluctuant pricing model. The price of a stock symbol constantly changes depending on the number of transactions at the moment.

---

<sup>1</sup> Jia Zhang is currently with BEA Systems, Inc., and a Guest Scientist of National Institute of Standards and Technology (NIST).

Figure 1 shows the stock price curve of Yahoo (Nasdaq: YHOO) from 11/15/2005 to 2/15/2006 according to published data from E\*TRADE financial (ETrade 2006).



**Figure 1. Illustration of fixed/interactive pricing model**

As shown in Figure 1, the fluctuant pricing model automatically adjusts the service price subject to market demand, activity capacity, and changing environment. It obviously promises higher flexibility and larger optimization space for service providers. However, to date most service providers merely adopt traditional fluctuant pricing models without adaptation. There is rarely research focusing on justification and verification of traditional pricing models in the context of Web services paradigm. We argue that because of the unique features of Web services, such as dynamic service discovery and invocation and heterogeneous interactions, traditional pricing models deserve re-examination.

Auction, as one of the fundamental business operation approaches, can be used as one fluctuant pricing model. However, both traditional physical auctions and recently emerged electronic auctions (e-auctions) are basically conducted by human beings, although computer technology is adopted in e-auctions to facilitate auction processes. Conducting auctions in the context of Web services poses significant challenges, because the unique features of Web services implies machine-to-machine interactions, re-enterable bargaining services, and a more ideally independent bidding, etc.

To our best knowledge, to date there has no published research formally modeling Web services-oriented auctions. Some reported works adopted the basic format of auction as a negotiation method between Web service providers and requestors (Esmailsabzali and Larson 2005; Huang, Chen et al. 2005). Some other researchers adopted the Web services technology to implement e-auctions, such as eBay (EBay 2005). Contrast with their work, our research aims at establishing a fundamental model for adopting auction mechanism to facilitate Web services negotiation and interactions.

The ultimate goal of our research is to examine the feasibility of applying the traditional auction theories to the field of Web services and establish corresponding Web services-specific auction models. In detail, we first examine the specific features of Web services towards applying auctions. Then we establish Web services-oriented auction model. We focus on investigating and mathematically proving how service providers can decide different service auction strategies to

obtain higher profit. Furthermore, we propose to utilize the technique of Service Level Agreement (SLA) documents as resources to deduce service requestors' preferences.

The remainder of the paper is organized as follows. After preparing readers for the basic concepts of traditional auctions, we motivate our research. Then we construct a formal model of Web services-based one-sided sealed auction, and mathematically prove that different auction strategies bring different profits to service providers. We propose our technique of eliciting service requestor's risk preferences from SLA documents. Basic risk preferences eliciting algorithm and historical data-based risk preferences prediction model are presented. We then discuss the implementation details of a prototype as an intelligent engine that helps service providers select optimal auction strategy. Afterwards we discuss related work. Finally, we make conclusions.

## **BASIC AUCTION CONCEPTS**

In this section, we will briefly introduce the basic auction concepts to prepare readers for the rest of the paper. McAfee formally defined an auction as "a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants." (McAfee and McMillan 1987) Auction has been used extensively as one of the fundamental bargaining forms in human society since antiquity for the sale of objects as early as 500 B.C (Krishna 2002). An auction event may involve independent auction activities on multiple auction items. A specific auction activity typically includes one or more identical auction items, one or more sellers, and one or more bidders. Bidders compete for the auction item(s); and the bidders bidding for the highest prices will be granted the auction items.

Electronic auctions, or so-called e-auctions, refer to the virtual auctions conducted via the Internet. E-auction differs from spot auctions in several significant ways. First, the average time limit is longer than spot auctions. Second, electronic bidders may participate in a common auction via the Internet at different times and different places. Third, e-auctions need to serve a scalable and growing Internet population. Fourth, bidders may obtain more information during the bid as they can surf online or do some research on the object at the same time as they bid. However, both physical auctions and e-auctions are conducted by human beings. E-auctions have been typically applying traditional auction theories and methodologies in practices and have been successful. During the past two decades, a variety of popular Web sites were established to support electronic auction activities, such as eBay (EBay 2005), onSale (OnSale), AuctionNet (AuctionNet), NETIS auction Web (AuctionWeb), etc.

## **RESEARCH MOTIVATION**

The rapidly emerging Web services paradigm has been catching significant momentum from both academia and industry in recent years. Simply put, a Web service is a programmable Web application that is universally accessible through standard Internet protocols (Ferris and Farrell 2003). Web services typically adopt a Service Oriented Architecture (SOA)-based operational model (Roy and Ramanujan 2001). Three types of components (i.e., roles) are identified: service provider, service requestor, and service broker. Service providers develop, deploy, and host Web services on their own sites. Then the service providers register with public or private service brokers (or so-called service registries) the Web services of their metadata descriptions, e.g., their service types, capabilities, and network addresses. The services brokers subsequently publish

registered Web services to the world. Service requestors describe to public service brokers the demand for specific kinds of services; then the service brokers search the registered service metadata and deliver back the results that match the requests. Using the network addresses retrieved from the service brokers, the service requestors bind to the located service providers. After negotiating with the service providers as appropriate, the service requestors access the required Web services and execute the services from the sites of the service providers.

In this negotiation process, theoretically any traditional business negotiation approaches could be used for Web services bargaining. Auction, one of the fundamental business negotiation methods, can be considered as a good candidate in such a context. Here we informally define a Web services-based auction. A service provider, or auctioneer, acts as a seller who hosts a Web service; multiple service requestors bid for the service. The service requestor, or bidder, who bids the highest wins the service. Without losing generality, we follow the way of traditional auction research to consider only one type of bidding object in such an auction. If the service provider can support multiple service requestors simultaneously, multiple service requestors can be chosen. As shown in this definition, the form of auction fits well in the Web services negotiation context; thus, it obviously could be applied to the field of Web services. For the rest of this paper, we exchangeably use the terms “Web services-based auction”, “Web services-oriented auction”, “Web services auction”, and “service auction”.

However, the unique features of Web services decide that a Web services-oriented auction bear several significant differences compared with the traditional auctions. First, bidders in a traditional auction bid for goods; while service requestors in a service auction bid for Web services. In a traditional auction, the winner wins the bidding goods and owns the goods. In a service auction, the winner wins the service in the sense that the service provider needs to provide the service to the winner only once (or multiple times if it is defined so). Therefore, in a tradition auction, the ownership of the bidding object belongs to the winner; whereas in a service auction, the ownership of the bidding service is still possessed by the service auctioneer such that the same service can be put for another bid after the winner finishes using it. In one word, a service auction is a re-enterable auction type.

Second, Web services interactions imply machine-based interactions, instead of human-based interactions. In the traditional auction research, one typical mechanism is to investigate optional bidders' strategies based upon all participants' preferences. These preferences are normally formed from auction participants' personal background information, e.g., household income, personal taste, education levels, reputation, and other generic characteristics. In a machine-to-machine service auction, a participant's preference can only be deduced from published machine-readable documents available. These documents include the Web Services Description Language (WSDL) specifications from service providers and the Service Level Agreement (SLA) documents from both service providers and service requestors. These two kinds of documents are *ad hoc* industry standard languages specifically coined for Web services communications. A WSDL document allows a service provider to define the functionality of a Web service and the way how to invoke the service. A SLA document defines a formal contract between a service requestor and a service provider, aiming at specifying quantifiable issues under varying contexts based upon mutual understandings and expectations (WS-Agreement 2005). In other words, a SLA document defines the agreements to govern service provision and receipt.

Third, the essential aspect of SOA model is the concept of dynamic discovery and invocation. Service requestors dynamically search and locate services from service brokers, and invoke the Web services from the service providers over the Internet upon an on-demand basis. Therefore, it is unlikely that the service requestors in one Web service auction would ally to bargain a better

price. As a matter of fact, under most circumstances the only reason for the bidders (i.e., service requestors) to come to the same service auction is that they happen to discover the same service from the same service provider. Hence compared to the human-based auction, it is more reliable to believe that bidders are independent in information collecting, price bidding, and private value of the service. We will come to this point in later sections.

Due to these apparent differences, the mature research results on traditional auctions deserve to be re-examined in the context of service auctions. Meanwhile, computing technologies should be utilized to facilitate service auctions. Note that the term *Web services-based auction* marries the Web services technology and the auction model. It can refer to two directions. On one side, it refers to conducting auctions utilizing the Web services technology as vehicles and tools; on the other hand, it refers to conducting inter-Web services communication utilizing the auction as approach. In this research, we refer to the latter meaning. Throughout the rest of this paper, we use the term “traditional auctions” to refer to human-based auctions including the traditional physical auctions and e-auctions, as opposite to “Web services-based auctions” that are machine-based auctions.

## WEB SERVICES AUCTION MODEL

In this section, we will present our Web services auction model. First we formally define a generic service auction problem; second we mathematically prove that different auction strategies may bring different profits to service providers based upon service requestors’ risk preferences.

**Definition 1.** A Web service auction problem can be defined as a six-element tuple  $SA = (ws, WS, AM, SR, R_c, R_s)$  where:

- $ws = (ws_{wsdl}, ws_{impl}, ws_{sla}, sp)$  is a four-element tuple that represents the Web service to be bid on, where:
  - $ws_{wsdl}$  is the interface description of the Web service in WSDL;
  - $ws_{impl}$  is the executable implementation of the Web service;
  - $ws_{sla}$  is the SLA document associated with the Web service;
  - $sp$  is the service provider who acts as the *seller* in the action;
- $WS = \{ws_1, ws_2, \dots, ws_m\}$  is the set of identical Web services that the service requestors face. In other words, the service provider may be able to simultaneously support multiple service requestors.
- $AM = \{am_1, am_2, \dots, am_n\}$  is a set of alternative auction methods (i.e., pricing strategies) that can be either statically or dynamically applied over the auction.
- $SR = \{sr_1, sr_2, \dots, sr_p\}$  is a set of service requestors acting as *bidders* that bid for the Web services.  $p = |SR|$  is the total number of bidders, which is an unpredictable finite natural number. This means that a Web service provider cannot set up the number of bidders, and both service provider and requestors do not know how many requestors altogether participate in the auction before the auction is completed. Each service requestor is a tuple  $sr_k = (sr_{k,contact}, sr_{k,sla})$ , where:
  - $sr_{k,info}$  is the unique information of the *kth* service requestor (will be discussed in detail in Definition 6);

- $sr_{k,sla}$  is the SLA document associated with the  $k$ th service requestor;
- $R_c = \{r_{c1}, r_{c2}, r_{cc}\}$  is the innate common set of rules that governs the service auction, and
- $R_s = \{r_{s1}, r_{s2}, \dots, r_{ss}\}$  is a set of rules that is defined for the specific service auction and governs the auction.

**Definition 2.** The unique concept of Web services implies the following three fundamental rules to a Web service auction:

- $r_1$  (Private value): The private information of a Web service requestor is her own value for the service, and it does not depend on what other requestors know.
- $r_2$  (Independent service requestor): All Web service requestors are independent in terms of what they need, when and where they find the service site, whether to participate in the auction, and what to bid for the service. In addition, there is no communication between them.
- $r_3$  (Symmetry): Requestor's private value comes from the same distribution function that is commonly known by all service requestors and the corresponding service seller.

**Rationale:** In order to enable auction research, a set of definitions or pre-defined assumptions are commonly established to simplify auction environments (Fudenberg and Tirole 1991). Examining the common auction assumptions in the context of Web services, the above three rules are established as the fundamental rules to delimit a quantifiable Web services-based auction environment. In Rule  $r_1$ , the private value that a requestor bids is the maximum amount of money she is willing to pay for the service, or called, willingness-to-pay for the service. Rule  $r_2$  states that the requestor's value is known only to herself, regardless of what others think. Rule  $r_2$  also implies that the requestor would not change his value even if she learns others' private information.

Web services are executable and programmable Web applications published on the Internet; they can be discovered from common public registries (here we omit the private service registry) and retrieved using standard Internet protocols. Thus, it is reasonable to consider that service requestors are independent from each other, and that there is no coalition among them. They come to the same service auction just because they happen to discover the same Web service from public registries and they are guided by those service registries to the same service provider. In addition, this rule can be further supported by the targeted scenario of Web services: service requestors dynamically discover the Web service and request for service in an on-demand manner. Therefore, it is reasonable to expect no assembly among service requestors. Since service requestors do not form any assembly, they only know of their own expected value for the bid service, i.e., price range that they are willing to pay for the bid service (as specified in the second common Rule  $c_2$ ).

**Definition 3.** Without losing generality, we predefine a set of rules that governs all service auctions to simplify the Web services auction. In a Web service auction:

- $r_4$ : No entry fee is needed for service requestors.
- $r_5$ : The service provider bears no operation cost in selling services.
- $r_6$ : Only the service provider knows the true value of the bidding service.

- $r_7$  : Only the winner requestor needs to pay for the bid service; other bidders who lose the auction pay nothing.
- $r_8$  : All service requestors in the same service auction adopt the same strategy.
- $r_9$  : Service requestors always adopt a symmetric Nash equilibrium strategy in auctions.
- $r_{10}$  : The service provider is aware of the service requestors' strategy, i.e., Nash equilibrium.

**Rationale:** Rules  $r_4 \sim r_{10}$ , along with the Rules  $r_1 \sim r_3$ , are acceptable for service auctions. Rules  $r_4 \sim r_7$  are widely adopted in the traditional auctions and present online e-auctions; therefore, it is reasonable to introduce them as pre-defined rules into generic Web services auctions without enforcing unrealistic limitations. Rule  $r_8$  aims to simplify Web services auction by assuming that all participants bear the same strategy, as adopted by the game theory. It can also be stated as, for example in a two-bidder auction, that “I know that you know what I know; you know that I know what you know; both of us know that what we adopt a strategy”.

Rules  $r_9$  and  $r_{10}$  intend to simplify Web services auction by applying widely adopted auction assumptions. These two rules state that we focus on equilibrium state in the Web service auction. The Nash equilibrium is the set of strategies that no player can benefit by changing the strategy while the other players keep their strategies unchanged. For the sake of completeness, here we provide the definition of the Nash equilibrium. For detailed information please refer to (Fudenberg and Tirole 1991). A strategy profile  $\hat{a} = (\hat{a}_1, \hat{a}_2, \dots, \hat{a}_p)$  is a Nash equilibrium if for each player  $i$  and  $\hat{a}_i \in A_i$ ,

$$u_i(a) \geq u_i(\hat{a}_1, \hat{a}_2, \dots, \hat{a}_p)$$

where  $\hat{a}_i$  is the strategy that person  $i$  takes,  $A_i$  is alternative strategy sets. For  $a = (a_1, a_2, \dots, a_p) \in \prod_{j=1}^i A_j$ ,  $u_i(a)$  is  $i$ 's payoff if the strategy profile  $a$  is played. In one word, a Nash equilibrium strategy is a strategy profile in which each agent is as good a response to what others do as any other strategy alternatives.

Furthermore, since our main purpose in this research is to prove that service providers can adopt different auction strategies to obtain different profits under different auction scenarios, we introduce these reasonable assumptions to narrow down to some auction scenarios to facilitate our discussions, instead of exhausting all auction scenarios.

## Auction Strategies

As we specified in Definition 1:  $AM = \{am_1, am_2, \dots, am_n\}$  refers to a set of alternative auction methods (i.e., pricing strategies) that can be either statically or dynamically applied over a service auction. In this section, we will discuss the available auction strategies that can be applied to Web services auctions.

Looking inside the traditional auctions and e-auctions, several types of taxonomy (Klemperer. 1999; McAfee and McMillan 1987; Wurman, Wellman et al. 1998) were identified regarding auction seller strategies. Four major taxonomies are summarized in Table 1 as follows.

**Table 1. Auction categories**

Price trend	Seller request		Bidder price		Marketable
English	One-sided	First-price	Sealed	Single-item	Marketable
		Second-price		Discriminatory	
Dutch	Two-sided		Outcry	One price	Non-marketable
				All prices	

The first categorization is based upon the trend of the auction prices: whether the bidder prices increase or decrease in the process of an auction. Two categories are identified, namely, English and Dutch. In an English auction, auctioneers start at a low price and incrementally increase the price; in a Dutch auction, auctioneers start at a high price and incrementally decrease the price.

The second categorization is based upon seller's requests: whether only bids are permitted. Two categories are identified, namely, one-sided and two-sided. In a one-sided auction, only bids are permitted; sellers are not allowed to "ask". In a two-sided auction, not only bids are permitted, sellers can "ask" prices at the same time. As shown in Table 1, a one-sided auction can in turn be divided into two sub-categories, based upon the "winner payment" method. In both types, the bidder with the highest price will win the auction. However, in a first-price auction, the winner pays the highest bidding price; in a second-price auction, the winner pays the second highest bidding price.

The third categorization is based upon how bidders submit bidding prices: whether the bids are published or kept secret. Two categories are identified, namely, sealed and outcry. In a sealed auction, all bidding prices are kept secret to other bidders; in an outcry auction, some bidding price can be viewed by other bidders. As shown in Table 1, a sealed auction can in turn be divided into two sub-categories, based upon the number of bidding items. In a discriminatory auction, more than one bidding items exist. An outcry auction can be in turn divided into two sub-categories, based upon whether part or all of bidding prices are available to other bidders. In a one-price auction, only the upper-bound or lower-bound bidding price at the checking moment will be visible, based upon whether the auction is an English or Dutch one. In an all-price auction, all bidding prices are available to all bidders.

The fourth categorization is based upon whether a seller has to sell a bidding item; in other words, whether a seller can cancel a bidding item in the process of an auction. In a marketable auction, as long as a bidding item is published, a final winner will be granted and the seller has to give the bidding item to the winner. In an unmarketable auction, a seller can decide to cancel an auction if she believes that the prices that the bidders are willing to pay are lower than her predefined lowest price.

## **Auction Strategies under Investigation**

After carefully examining the possible auction strategies identified from the traditional auctions and e-auctions, we found that some auction strategies are symmetric to others, such as English auctions and Dutch auctions; and some auction strategies are extensions to others, such as one-price outcry auctions and all-price outcry auctions. Therefore, we need to investigate only one auction type; the results can be applied to the other ones.



In our research, we decided to focus on English auctions, because the English auction type is more popular in traditional auctions. In addition, how single-item sealed auction type and discriminatory sealed auction type lead to different seller strategies is rather a generic economics research topic, instead of one limited to Web services auctions. Therefore, we only consider single-item auction. In other words, using the notations in the Definition 1, the set of Web services to be bid on contains only one Web service instance:  $WS = \{ws\}$ . Sealed auctions and outcry auctions are both possibly applicable to Web services auctions. However, outcry auctions require a standard way for service providers to publish bidding prices and to communicate with service requestors automatically. Although the requirement is possible to be implemented in the future, the solution may require extra standardization efforts. Therefore, we only consider sealed auction type. Finally, whether a service provider allows herself to cancel an auction in the process is rather related to her advertisement profile than a scientific decision. It is more likely to depend on her own business requirements predefined. Therefore, we do not consider the marketable/non-marketable auctions.

As a result, the auction categories under investigation in this research are summarized in Table 2, with excluded categories grey out.

**Table 2. Auction categories under investigation**

Price trend	Seller request		Bidder price		Marketable
English	Single-sided	First-price	Sealed	Single-item	Marketable
		Second-price		Discriminatory	
Dutch	Double-sided	Outcry		One price	Non-marketable
				All prices	

Using our notations in Definition 1:  $SA = (ws, WS, MM, SR, R_c, R_s)$ , each auction category above predefines some rules into  $R_s = \{r_{s1}, r_{s2}, \dots, r_{sq}\}$ .

## Decision on Auction Strategies

Economists have been carefully examining the auctioners' behaviors and auctioneer's decision making process. Their research results, summarized in the well-known framework game theory (Fudenberg and Tirole 1991; Mas-Colell, Whinston et al. 1995), provide a guideline and starting point for Web services auction practices. According to the game theory, buyers' strategy is different from auction rules set by sellers; thus, it is critical for sellers to choose an optimal auction type to stimulate bidders. In order to decide the best auction type, sellers need to predict the behaviors of bidders and their expected profits thereafter. Hence, our goal is embodied into two directions from the service providers' perspective: how to predict the behaviors of service requestors, and how to decide on auction strategies (types). Our approach starts from the latter direction and is two-fold: first, we examine the seller's process of deciding auction types, thus extract the criteria of predicting service requestor behaviors; second, we propose how to extract and predict service requestor behaviors, which will be discussed in detail in the section of SLA-based service requestor preference deduction.

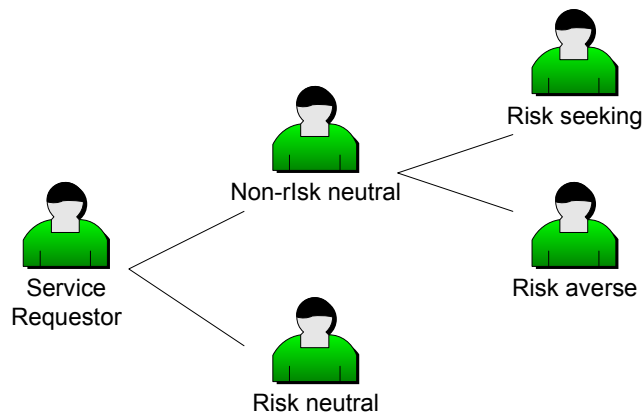
## Service Requestor Information

According to the game theory, auction sellers, or, auctioneers, need to understand and examine bidder behaviors based upon bidder types, goals, and preferences. Applying this theory, we introduce the following formal definition of service provider preferences:

**Definition 4.** In a Web service auction, a utility function  $u$  is a numerical value to a service provider or service requestor when she chooses a strategy:  $WS = \{ws\}$  , and  $sr \in SR = \{sr_1, sr_2, \dots, sr_p\}$  . The function value represents her preference. If  $u(x) \geq u(y)$ , if and only if the party prefers strategy  $x$  to strategy  $y$ . A von Neumann-Morgenstern (vNM) utility function is that a function has the expected utility property: the service provider or service requestor is indifferent between receiving a given bundle or a gamble with the same expected value.

**Definition 5.** Consider that in a Web service auction, a service provider/requestor, or called, an agent chooses a strategy  $x$  . It bears a vNM utility function  $U$  over  $x$  as  $U(x)$  and a mathematical mean function  $E$  over  $x$  as  $E[x]$  . The agent  $x$  is risk averse if  $E[U(x)] < U(E[x])$  . The agent is risk neutral if  $E[U(x)] = U(E[x])$  . The agent is risk proclivity (or, risk-loving or risk-seeking) if  $E[U(x)] > U(E[x])$ .

As shown in Figure 1, a service requestor's preference towards a service auction is defined in two categories: risk neutral and non-risk neutral. A service requestor is considered as risk neutral to a service auction if she is indifferent to bidding for it. Otherwise, the service requestor is considered as non-risk neutral. A non-risk neutral service requestor can be further defined as risk averse if she refuses the bidding service, or risk seeking if she happily accepts the bidding service.



**Figure 1. Categorization of service requestors in a service auction**

**Definition 6.** The unique information of the  $i$ th service requestor  $sr_i$  in a service auction is represented by a tuple  $sr_{i,info} = (t_i, u_i)$  where:

- $t_i = (t_{i,1}, t_{i,2}, \dots, t_{i,o}), t_{i,a} \in R, o \in I$  represents the type of service requestor  $sr_i$  , which contains a list of attributes specific to the auction and the service requestor;  $t_i$  is the *type* of bidder  $i$ , or we can call requestor  $i$  a *type  $t$  requestor*.

- By rules  $r_1 \sim r_3$ ,  $t_i$  is uniformly distributed, i.e.,  $t_i \propto \cup[0, V]$ .
- $u_i : AM \times t_i \rightarrow R$  is the real value of the utility function applied on the service requestor  $sr_i$  based on her type. (Recall that  $AM$  represents the alternate auction strategies that can be applied onto the service auction.)
- Every service requestor  $sr_i$  is an expected utility maximizer, i.e.,  $u_i = \underset{u_i}{MAX} Eu_i(t_i)$ .

In a traditional auction, what a seller cares most is the bidders' risk preferences: the bidders' strategies are different under the conditions whether the bidders really want to win the auction or not. If a seller in some way knows, or assumes, the bidders' risk preferences, she could adjust the auction strategy accordingly so as to gain the most profit. Let us examine in a Web service auction, how a service provider should adjust auction strategy based upon service requestors' various types of preferences.

**Definition 7.** In the process of a service auction, each service requestor  $sr_i$  specifies five values  $(e_i, l_i, h_i, b_i, B_i)$  to a service on bid:

- $e_i \in R$  represents the private value that the requestor holds for the service;
- $l_i \in R$  represents the lowest value that the service requestor would pay for the service;
- $h_i \in R$  represents the highest value that the service requestor would pay for the service;
- $b_i \in R$  represents the requestor's bidding price during a service auction; and
- $B_i \in R$  represents the value that the service requestor finally pays for the service. This paying price, or requestor's payment, depends on the auction type.

If the service requestor wins the bid, she will pay  $B_i$ ; otherwise, she will pay nothing.

**Definition 8.** A service requestor  $sr_i$ 's utility function  $u_i$  is defined as a function over her risk preference  $rp_i$ :  $u_i = f(rp_i)$ .  $u_i$  can exhibit in one of the two forms: either as a linear function yielded by risk neutral preference, or a non-linear function yielded by non-risk neutral preference:

$$u_i = \begin{cases} e_i - B_i, & \text{if } p_i \text{ is risk neutral;} \\ (e_i - B_i)^\theta, & \theta \in (0,1) \cup (1, \infty), \text{ if } p_i \text{ is non-risk neutral;} \end{cases} \begin{cases} \theta \in (0,1), & \text{if } sr_i \text{ is risk averse;} \\ \theta \in (1, \infty), & \text{if } sr_i \text{ is risk seeking} \end{cases}$$

where  $e_i$  is the private value service requestor  $sr_i$  assigns to the service,  $B_i$  is the final payment of  $sr_i$ .

In other words, if the service requestor  $sr_i$  wins the bid, she will obtain the value of  $e_i - B_i$ , and 0 otherwise.

Since a service requestor's utility function can exhibit in two forms due to different preference types, now we examine the service provider's expected profits with respect to the two forms.

## Service Provider Strategy Decision Deduction

With the previous notations and preparation, in this section, we prove mathematically how a service provider should select optimal auction strategy towards higher profit. We utilize the categorization information in Table 2 to guide our discussions.

As shown in Table 2, we will consider in a one-sided sealed auction, how a service provider will decide whether a first-price or a second-price strategy should be used. The decision should be made based upon different utility functions adopted by the corresponding service requestors. As we discussed in the last section, there are two types of utility functions: linear or non-linear. We will discuss each type in detail as follows.

### Risk Neutral Bidders

When service requestors are risk neutral, their utility functions exhibit as linear functions. As specified in Rule  $r_8$  in Definition 2, all service requestors in the same auction always adopt the symmetric Nash equilibrium strategy (e.g., see Mascale et al 1995):

$$b_i = b_i(e_i) = \frac{p-1}{p} e_i,$$

where  $p$  represents the total number of service requestors in the bid at the moment,  $e_i \in R$  represents the private value the service requestor  $sr_i$  assigns to the service.

Now let us compare the service provider's profit under first-price and second-price strategy. First, we calculate the service provider's expected profit (revenue) under first-price strategy. Recall that under uniform distribution on (0,1), for which:

$$F(u) = u, \text{ and } f(u) = 1, \text{ for every } u \sim (0,1)$$

The expected profit for the service provider is the mean of what she gets from the winner, i.e. :

$$\begin{aligned} r_{\pi 1} &= E[\text{winner's payment}] = E[b_i | i \text{ wins the bid}] \\ &= \frac{p-1}{p} E \max\{e_1, e_2, \dots, e_p\} \\ &= \frac{p-1}{p} \int_0^1 u p f(u) [F(u)]^{p-1} du \\ &= \frac{p-1}{p} \int_0^1 u p u^{p-1} du \\ &= \frac{p-1}{p+1} \end{aligned} \tag{1}$$

Then let us consider the service provider's profit under the second-price auction:

$$\begin{aligned} r_{\pi 2} &= E[\text{winner's payment}] = E[b_j | i \text{ wins, } b_j < b_i, \text{ for } i \neq j, \text{ and } b_j > b_k, \text{ for all } j \neq k \text{ and } k \neq i] \\ &= E[2^{\text{nd}} \text{ highest}\{e_1, e_2, \dots, e_p\}] \end{aligned}$$

$$\begin{aligned}
 &= \int_0^1 up(p-1)f(u)[1-F(u)](F(u))^{p-2} du \\
 &= p(p-1)\int_0^1 u(1-u)u^{p-2} du \\
 &= p(p-1)\int_0^1 (u^{p-1} - u^{p-2})du \\
 &= p(p-1)\left(\frac{1}{p} - \frac{1}{p+1}\right) \\
 &= \frac{p-1}{p+1}
 \end{aligned} \tag{2}$$

Comparing (1) and (2),  $r_{\pi 1} = r_{\pi 2}$ . This means that if service requestors are risk neutral, the service provider can choose either the first-price or second-price strategy, without affecting their profit.

### Non-risk Neutral Bidders

In most cases, however, service requestors do have preferences over bidding service. In other words, service providers need to consider non-linear utility functions for participating service requestors. Taking the functional form as defined in Definition 8, we will come to the symmetric Nash equilibrium strategy that a service requestor has for bidding (Mascale et al 1995):

$$b_i(e_i) = \frac{p-1}{p-1+\theta} e_i$$

The service provider's expected revenue can be deduced as:

$$\begin{aligned}
 r_{\pi 3} &= E[\text{winner's payment}] = E[b_i | i \text{ wins the bid}] \\
 &= \frac{p-1}{p-1+\theta} P(i \text{ wins}) E[b_i | b_i \geq b_j, \forall i \neq j] \\
 &= \frac{p-1}{p-1+\theta} P(t_i \geq t_j) E[b_i | b_i \geq b_j, \forall i \neq j] \\
 &= \frac{p-1}{p-1+\theta} \int_0^1 [F(u)]^{p-1} uf(u) pdu \\
 &= \frac{(p-1)p}{(p+1)(p-1+\theta)}
 \end{aligned} \tag{3}$$

Under the second price auction setup, the bidder's dominant Nash equilibrium strategy is the same as if she were risk neutral, i.e.,  $b_i(t_i) = t_i$ . Therefore, the service provider's expected profit will not change,

$$r_{\pi 4} = \frac{p-1}{p+1} \tag{4}$$

Comparing (3) and (4), we can see that  $r_{\pi 3} \neq r_{\pi 4}$ . If  $\theta \in (0,1)$ ,  $r_{\pi 3} > r_{\pi 4}$ ; if  $\theta \in (1,\infty)$ ,  $r_{\pi 3} < r_{\pi 4}$ . In either way, the service provider expects different profits under different auction strategies. Intuitively, if the service seller believes that bidders are risk averse ( $\theta \in (0,1)$ ), she

prefers to conduct the first-price auction to the second-price auction. On the other hand, if the service seller believes that bidders are risk seeker ( $\theta \in (1, \infty)$ ), she prefers the second price auction.

In summary, we have proved that a service provider should adopt different auction strategies to obtain high profit if she knows corresponding service requestors' risk preferences. Thus, our research challenge has turned into how to discover and predict service requestors' risk preferences. In traditional auctions, bidders' risk preferences are typically predicted based upon bidders' background information such as personal taste and household wealth, as well as bidding object's perspectives, such as its price range, durability, functions, and personality (McAfee and McMillan 1987). These information are normally obtained from survey investigation and bidding objects' selling experience (Zhang, Zhang et al. 2004). However, a Web service auction is dynamically formed with unpredictable service requestors; these normally used information cannot be easily found. Therefore, a service provider needs a new method to predict service requestors' risk preferences. Our approach is to utilize service requestors' Service Level Agreement (SLA) documents in addition to the service provider's SLA document.

## **SLA-BASED SERVICE REQUESTOR PREFERENCE DEDUCTION**

### **Introduction of SLA**

The term Service Level Agreement (SLA) was coined to define a formal contract between a Web service requestor and service provider, aiming at specifying quantifiable issues under various contexts on the grounds of mutual understandings and expectations (WS-Agreement 2005). In other words, an SLA intends to produce an agreement regarding the guarantees of a Web service. It should be noted that SLA can be used as an approach to formally define any service-related issue, not limited to the final mutual agreement. An SLA can facilitate Web services adoption and distribution by benefiting both service providers and requestors. A published SLA from a service provider will not only assure potential service requestors that they will get the service they pay for; but also obligate herself to achieve the service promises as well. Meanwhile a service requestor can use an SLA to express her requirements regarding to a service, so that a service provider can adjust service quality and quantity accordingly. For example, consider a service provider is facing two service requestors, one requiring service delivery within 10 seconds while the other requiring 30 seconds. Based upon the different SLAs, the service provider can assign more resources to the service requestor requiring faster service than the other one; thus to satisfy both requestors.

In theory, the quantifiable issues could cover whatever each or both sides are interested in, including Quality of Service (QoS), prices, constraints, and any other requirements or preferences. An interested issue is described as a single or multiple SLA parameters, which are normally based on domain-specific vocabularies. A Web service provider may define one or more SLAs for one Web service in order to serve different service requestors. For example, United Airline may offer different SLAs with different qualities to serve different categories of clients, e.g., business organizations, frequent flyers, and casual customers, etc.

To date, several SLA specifications and proposals have emerged. The two causing most attentions are Web Services Agreement Specification (WS-Agreement) from Global Grid Forum

(WS-Agreement 2005) and Web Service Level Agreement (WSLA) from IBM (IBM 2003). Their goals are to standardize the terminology, concepts, agreement structure, agreement terms, agreement templates, as well as WSDL for defining state and message exchanges.

WS-Agreement defines a language and a protocol for service providers to advertise Web services. It allows business parties to use extensible XML language to specify the nature of the agreement and agreement templates, so as to facilitate business match and discovery and emphasize flexibility, extensibility and compatibility. A WS-Agreement-based specification generally contains three parts: (1) the schema of the agreement, (2) the schema of the agreement template, and (3) agreement-specific life-cycle management port types and operations. Such an agreement defines one to many service level state-dependent requirements as expressions of resource availabilities (e.g., memory, CPU, disk space) and service qualities (e.g., response time).

WSLA also defines a language primarily for service providers based upon XML. As a matter of fact, it is defined as an XML schema. WSLA provides a technique to define service parameters and algorithms that specify how to measure and decide the deviation and failure of a promised service guarantees, and a technique that monitors and governs services. In addition, WSLA provides an extensible mechanism to allow users to specify domain-specific vocabularies, which largely increases the extensibility of WSLA to be applied in different domains.

## **SLA-based Preference Elicitation**

In our research, we decide to deduce service requestor's risk preferences from service requestor SLA documents in the WS-Agreement format. The main reason of our selection of WS-Agreement is its popularity. WS-Agreement is proposed by Global Grid Forum (GGF) (GGF), which community consists of thousands of individuals in both research and industry, and represents over 400 organizations in more than 50 countries. WS-Agreement has been gaining significant attention from both Web services field and Grid computing field. It is a pure XML-based approach to formally describe service-level requirements as well as service provider or requestor's information. It should be noted that our approach of SLA-oriented preference elicitation is not limited to WS-Agreement. Instead, it can be easily applied to any XML-based SLA standards, such as WSLA, with limited changes.

It should be noted that the term *risk preference* in traditional economics is an abstract concept representing a buyer's tendency of participating in a lottery and of paying the price. As described above, it is one characteristic that an agent process by disclosing features of utility function. The comparison of risk preference is mainly on who is more risk averse with using risk averse ratio (Arrow 1963; Arrow 1965). In other words, risk reference, *per se*, is not numerically representable. However, we believe that certain types of risk preference lead to some certain behaviors, which fact gives us the idea that "risk preference" can be deduced, or revealed, from observed behaviors. The rest of this paper discusses how we conduct an analysis to reveal a service requestor's risk preference from some of its behaviors. For example, SLA documents define some agreement that is related to risky behaviors, such as delivery time, penalty-care if failing to fulfill the agreement, urgency of the service, confidentiality of participants, etc. A service requestor ranks each of the criteria, which by combination, expose her risk preference. We provide a formal definition of this "revealed" risk preference idea later in this paper (see Definition 10). On all accounts, in the machine-to-machine only, re-enterable, dynamic services auctions, abstract concepts have to be translated into quantifiable concepts that are numerical representative, before they can be used in the process. This paper is the first attempt to extend and

digitize the traditional risk preference concept into calculable and comparable numbers on top of Web services specific and available machine-understandable SLA documents.

Recall that in the last section, we conclude that a service provider can adopt different auction strategies to gain higher (expected) profit, by analyzing and predicting service requestors' risk preferences. In an SLA document, both service providers and requestors can decide to specify their background information and specific requirements on the bidding service. These specifications can assist each part to understand the other part's basic information and requirements related to the service requesting or provision. On one hand, as we state above, an SLA from a service provider can document the service description, features, and delivery parameters. On the other hand, an SLA document from a service requestor can reveal how urgently she needs a service, how serious she is, how much she plans to pay for the service, what her risk preference type regarding to the service, etc. Those serious requestor, or those who need the service urgently, or both, will more likely specify detailed and specific quality requirements on a bidding service, and are more likely to compete for the bidding service if possible. If a requestor is risk loving for one type of service, she may be highly willing to pay more to obtain a service, regardless of other market environment. Furthermore, it has been extensively accepted that SLA is an indispensable element in Web services paradigm; thus, it is reasonable to assume that service providers and requestors will provide more or less SLA documents. Consequently, we believe that SLA documents are appropriate resources for service providers to elicit bidding service requestors' risk preferences. Our research challenge has turned into how to uncover and predict service requestors' preferences from published SLA documents written in WS-Agreement standard.

## Basic risk Preference Deduction Model

Our fundamental idea is that the definitions in an SLA document of a service requestor can be used to compare with the corresponding definitions in the SLA document of the corresponding service provider, to deduce the service requestor's risk preference. An SLA document from a service provider describes the basic information of a Web service, which is a guideline for service requestors to prepare their SLA documents related to that service. Note that we only consider interested service requestors as (potential) service bidders. If a service requestor provides completely unrelated or unmatched SLA documents compared with the service provider's SLA document, she can be considered as not serious about the bidding service; thus, she can be ignored from the corresponding service auction. This means that the relatedness of a requestor's SLA to a provider's produces the service bidder base. We thus delimit the temporal relationship between the SLA documents in a Web service auction as follows: a service provider (i.e., seller) publishes her SLA document to be associated with her published Web service; interested service requestors (i.e., bidders) submits their own SLA documents defining their specific requirements based upon the service provider's SLA document. These specific requirements may strengthen or emphasize some quality requirements, or release some.

A WS-Agreement-based SLA document is a pure XML-based document, which defines a set of service-related attributes as XML tags. For example, a service provider may declare in her SLA document that the service is guaranteed to be delivered in 10 seconds, by specifying a user-defined tag `<DeliveryTime = 10 Unit = s>`. The composer of an SLA document owner defines any number of attributes based upon specific service logic. Some of the defined attributes are more important than others if they are shared and used by both the service provider and corresponding requestors. We call these attributes *primary attributes*, such as delivery time,



response time, reliability, and all other attributes defined by the service provider<sup>2</sup> by default. Other attributes are defined by some service requestor alone. For example, a service requestor may independently set a requirement on a QoS attribute “fault tolerance”. The corresponding service provider may not have a mutual agreement on the attribute. These service requestor-specific attributes are called *secondary attributes*. We formally define an SLA document in Definition 9.

**Definition 9.** An SLA document  $ws_{sla}$  associated with a Web service in a service auction  $SA$  is defined as a tuple:

$$ws_{sla} = (PATT, SATT, ATTV), \text{ where:}$$

- $PATT = \{att_1, att_2, \dots, att_{N_p}\}$  is a list of primary attributes that are used by both the service provider ( $sp$ ) and service requestors ( $SR$ ),  $N_p \in I$  is the number of primary attributes defined;
- $SATT = \{att_1, att_2, \dots, att_{N_{SA}}\}$  is a list of secondary attributes that are defined by either the service provider  $sp$  or a service requestor  $sr_j \in SR$ ,  $N_{SA}$  is the number of secondary attributed defined;
- $ATTV = \{ \langle att_1, v_1 \rangle, \langle att_2, v_2 \rangle, \dots, \langle att_{N_p+N_{SA}}, v_{N_p+N_{SA}} \rangle \}$  is a list of name/value pairs containing the attribute (both primary attributes and secondary attributes) names and corresponding values specified.

We now define the risk preference of an interested service requestor on a specific primary service attribute. We propose to compare the corresponding specifications on the same attribute from both the service provider and a requestor. For example, a service provider may declare in her SLA document that the service is guaranteed to be delivered in 10 seconds, while one service requestor may ask for 5 seconds as upper bound, yet another service requestor allows 20 seconds response time frame. For a specific service QoS attribute, if a service requestor sets higher or tighter requirements than set by the service provider, the service requestor is considered as risk loving on the attribute; if a service requestor sets lower requirements, she is considered as risk averse on the attribute; if a service requestor sets the same value or does not set a value for the attribute, she is considered to accept the default value and thus can be viewed as risk neutral on the attribute. Using the example we discussed before, since the first service requestor sets more stringent requirements (5 seconds) on the response time, it is more likely that it is willing to bid higher price for the service than the latter requestor. In other words, the first service requestor is more risk seeking, while the second one is more risk averse.

It is possible that a service requestor sets acceptance ranges on some attributes than others. Consider the previous example when the service provider sets the default values of response time as 10 seconds and a service requestor asks for 5 seconds. According to our previous definition the service requestor should be considered as risk seeking on the attribute. However, if the service requestor declares that it does not matter if the time frame is met, then its risk preference should be adjusted to risk neutral. Therefore, we introduce a concept of *acceptance range* parameter to represent this constraint. The formal definition of the risk preference of a service requestor is summarized in Definition 10.

---

<sup>2</sup> Except service provider personal information such as contact address.

**Definition 10.** The revealed risk preference of a related service requestor  $sr_j \in SR$  on a specific service attribute  $att_i \in MATT = \{att_1, att_2, \dots, att_{N_A}\}$  is defined as follows:

$$rp_{j,i} = \alpha_{j,i} p_{j,i} = \begin{cases} = 1 & \text{if it is risk seeking;} \\ = 0 & \text{if it is risk neutral;} \\ = -1 & \text{if it is risk averse;} \end{cases} \quad (5)$$

where  $\alpha_{j,i}$  is the acceptance range of  $sr_j$  on attribute  $att_i$ ,  $p_{j,i}$  is the risk preference of  $sr_j$  on attribute  $att_i$ .

In order to further illustrate our consideration and facilitate our discussion, let us consider binary acceptance range parameter. In more detail, there are two acceptance settings associated with each service attribute, *urgent* or *non-urgent*. The crossover of isolated requirements levels (tighter requirements, looser requirements, same requirements) and the acceptance range parameters leads to  $3 \times 2 = 6$  combinations as shown in Table 3:

**Table 3. Risk preference decision table**

	Tighter requirements	Same requirements	Looser requirements
Urgent	Risk seeking	Risk neutral	Risk averse
Non-Urgent	Risk neutral	Risk neutral	Risk averse

As shown in Table 3, if a service requestor specifies that the request on the service attribute is urgent, and defined values are tighter, the same, or looser requirements than the defined values of the service provider, its risk preference is risk seeking, risk neutral, or risk averse, respectively. If she specifies that the request is not urgent, and her defined values are tighter, the same, or looser requirements, her risk preference becomes risk neutral, risk neutral, or risk averse, respectively. Note that it may sound like a trivial situation when a service requestor specifies a highly tighter requirement on a service attribute but declares that she can accept any service not meeting its requirement; for completeness reason, we still list the possibility.

As shown in Table 3, binary acceptance rate situations can be resolved using a matrix sufficiently. It is possible that a service requestor specifies more than two levels of acceptance rate, e.g., a vector of acceptance rates. We will consider the more complicated cases in our future research.

Risk preference of a service requestor is not easy to be observed directly. It is also unreliable by being revealed from only one SLA document because even for requestors with the same type of risk preferences, they may have different satisfaction levels with one requirement. For example, one risk averse service requestor may consider response time more important than fault tolerance; while another risk averse requestor may consider in the opposite way. In other words, the importance level of a certain service attribute is arbitrary and may differ across service requestors. Therefore, any information on an isolated attribute of a service is not sufficient for the service provider to induce a requestor’s risk preference. Rather, a better way is that risk preference is statistically induced from the requirements of a variety of attributes. We use a simple example to explain this idea. Suppose there is a person-based, self-reported survey. We do not know a person’s actual risk preference, and unable to get the reliable and true information even if we ask them. We then set up a series of hypothetical questions related to risk preferences that people can

easily answer. For example, are you going to buy a new product that no one else whom you know around ever heard of, the rank of new product in a wish-list of a recent shopping, intentions of being the volunteer user of a new product, etc. At the end, we can come up with the likelihood of a risk preference type that a person can be, through all answers to those questions. Similarly a service provider can only analyze a service requestor's risk preference, through all requests that include different valuation of various service attributes. The conclusion that the provider draws out is the form of possibility in percentage point that a service requestor is any of the type of preferences.

The formal definition of the risk preference of a service requestor is listed in Definition 11 as follows.

**Definition 11.** The risk preference of a related service requestor  $sr_j \in SR$  is defined as:

$$rp_j = \sum_{i=1}^{N_A} \omega_i q_i + \gamma = \begin{cases} > \varepsilon \Rightarrow rp_j = 1 \text{ is risk seeking;} \\ \in [-\varepsilon, \varepsilon] \Rightarrow rp_j = 0 \text{ is risk neutral;} \\ < -\varepsilon \Rightarrow rp_j = -1 \text{ is risk averse;} \end{cases} \quad (6)$$

where  $\sum_{i=1}^{N_A} \omega_i = 1$ ,  $N_A$  is the number of attributes specified,  $\varepsilon$  and  $\gamma$  are two predefined configurable small real numbers.

$\omega_i$  are decided based upon each specific service requestor. Various algorithms can be adopted. We will discuss one algorithm in the section of implementation details.

As shown in Definition 11, we introduce a small number  $\varepsilon$  to add precision. Because a service may include a number of attributes, combined with the corresponding weights make it difficult to lead to exact value 0. By introducing this small number  $\varepsilon$ , if the result is close enough to 0, we consider that the service requestor is risk neutral. Similar conclusions can be made to risk seeking and risk averse.

Furthermore, as shown in Definition 11, we introduce another small number  $\gamma$  to represent the impact from secondary attributes. It is up to the service provider to decide whether to consider secondary attributes and associated values from service requestors. Their influences are absolutely smaller than the primary attributes. For example, it is possible that a service requestor asks for slightly higher quality of a Web service. Consider again the example we used before, a service provider promises 10 seconds of response time, and a service requestor asks for 8 seconds. If the service provider considers that it can satisfy the requirement, it could notify the service requestor, and considers that the service requestor is risk seeking. On the other hand, if the requirement cannot be fulfilled, it could notify the service requestor and removes the requestor from further consideration. Since this exception can be handled accordingly, in this paper we do not consider this complicity for simplicity.

Finally we define the risk preference of the whole body of service requestors that a service provider is facing.

**Definition 12.** The risk preference of a body of related service requestors  $SR$  is defined as:

$$rp = \sum_{j=1}^{N_R} rp_j / N_R = \begin{cases} > \varphi \Rightarrow rp = 1 \text{ is risk seeking;} \\ \in [-\varphi, \varphi] \Rightarrow rp = 0 \text{ is risk neutral;} \\ < -\varphi \Rightarrow rp = -1 \text{ is risk averse;} \end{cases} \quad (7)$$

where  $N_R \leq |SR|$  is the number of related service requestors, and  $\varphi$  is a predefined configurable small real number.

As shown in Definition 12, another small number  $\varphi$  is introduced to decide when the result is not equal to value 0 but very close to it. If the result is close enough to 0, we consider that the base of service requestors is risk neutral. Similar conclusions can be made to risk seeking and risk averse.

As we have prepared all algorithms for calculating risk preferences, our basic risk preference deduction algorithm is summarized as follows:

### Basic Risk Preference Deduction Algorithm

1. Obtain the SLA document of the service provider;
2. Analyze the document {
  - 2.1. Parse the document
  - 2.2. Extract a list of analysis attributes and default values;
  - 2.3. Normalize the SLA attributes;
}
3. Obtain the SLA documents of all participating service requestors;
4. Iterate through the SLA documents from all participating service requestors {
  - 4.1. Parse the document
  - 4.2. Use normalized primary attributes from Step 2.3 to extract corresponding values required;
  - 4.3. Iterate through each attribute from Step 2.3 {
 

If the specified value is semantically more stringent than the default value and is urgent {  
The service requestor is set to be risk seeking on the attribute;

} else if the specified value is semantically looser than the default value {  
The service requestor is set to be risk averse on the attribute;

} else {  
The service requestor is set to be risk neutral on the attribute;
  - 4.4. Calculate the risk preference of the service requestor with weighted attributes
  - 4.5. Obtain secondary attributes and required values;
  - 4.6. Normalize the secondary SLA attributes;
  - 4.7. Use predefined secondary SLA attributes weight to adjust risk preference from Step 4.4;
}
5. Predict the risk preference of the base of service requestors

In Step 1, the service provider obtains her own SLA document. By analyzing the document in Step 2, she first parses the XML document and extracts a list of primary attributes and secondary

attributes associated with their values defined. These attributes need to be normalized using word normalizing technique, which will be discussed in detail in the section of implementation details.

In Step 3, the service provider fetches the SLA documents of all participating service requestors. How to fetch the SLA document of a service requestor is out of the domain of this paper. Then in Step 4, the service provider iterates through all SLA documents. First, she parses the document, uses the normalized primary attributes from her own SLA document to extract corresponding attribute values specified by the service requestor. For each primary attribute, Formula (1) is used to decide whether the service requestor is risk seeking, risk neutral, or risk averse on the attribute. After the risk preference value on each primary attribute is obtained, the risk preference of the service requestor on the bidding Web services is calculated using Formula (2).

Then the service provider analyzes secondary attributes and values defined. Specific algorithms can be applied to adjust the risk preference value obtained above based upon the secondary attributes, based upon the service provider's strategy. After getting the adjusted risk preference values of each service requestor, finally in Step 5, Formula (3) can be used to predict the risk preference of the whole base of related service requestors.

## Historical Data-based Risk Preference Prediction Model

In the last section, we propose a basic risk preference deduction model based upon SLA documents. When the service is published for a Web service auction for the first time, it is impossible for the service provider to predict the risk preference of the body of service requestors she is facing. Then the basic deduction model is the only way for the service provider to analyze and set up her pricing strategy after all service requestors report in the service auction. After the first time, the service provider can predict the risk preference of the service requestors in the next coming auction, since she has some historical data as the SLA documents of the services requestors from the previous service auctions,. Therefore, she can establish and publish a proper pricing strategy before she opens a new auction cycle of the same service (recall that a Web service is re-enterable auctions). Thus, we propose a historical data-based polynomial risk preference prediction model as follows:

$$rp^p(t+1) = \sum_{i=1}^t \frac{(1-\eta^i)rp^p(i) + \eta^i rp^r(i)}{e^{t-i}} \quad (8)$$

where  $rp^p(t+1)$  denotes the predicted auction requestors' risk preference at the  $(t+1)$  th auction cycle,  $rp^p(i)$  is the predicted risk preference at the  $i$ th auction cycle,  $rp^r(i)$  is the actual risk preference at the  $i$ th auction cycle, and  $\eta$  is the learning rate.

**Rationale:** The time point when we try to predict is right before the  $(t+1)$  th auction cycle, while no service requestors' SLA documents are available. Formula (8) predicts the risk preference in a coming auction cycle using the historical data from all of the previous auction cycles. As shown in Formula (8), more recent auction cycles weigh more than earlier ones due to the setting of the denominator. Our model utilizes both the previous prediction data and actual data, so as to gradually approach to an appropriate learning rate. The learning rate is obtained from every auction cycle iteration between predicted risk preference and actual risk preference as follows:

$$rp^p(t) = (1 - \eta^t)rp^p(t) + \eta^t rp^r(t) \quad (9)$$

The initial learning rate can start from a randomly assigned number such as 0.1. According to statistics theorem, when  $t \rightarrow \infty, \eta^t \rightarrow 0$ . Thus,  $rp^p(t) \rightarrow rp^r(t)$  after a large number of iterations (i.e., auction cycles).

The actual risk preference of a service requestor is calculated as follows. Based upon all the actual bid prices, we calculate a mean value. If a service requestor's bidding price is higher than the mean value, its actual risk preference is recorded as risk seeking; if its bidding price is lower than the mean value, its actual risk preference is recorded as risk averse; and if its bidding price is close to the mean value by a predefined small number, its actual risk preference is recorded as risk neutral.

## SLA Document Parser

We constructed an SLA document parser to automate the process of SLA documents analysis and risk preference elicitation. An SLA parser in our research takes an SLA document as an input, and generates a structured file as an output. The output document contains three parts: (1) SLA document goal, (2) SLA owner information, and (3) SLA requirements. Each part is a list of (name, value) pairs, with the name extracted from SLA document tags. The first part describes the intention of the SLA document. The second part describes who writes the SLA document, as well as some owner's background information, e.g., owner address. The last part lists a set of quality requirements over the targeted Web service, such as its availability, response time, quality, etc. Here we will focus on discussing how to obtain SLA requirements.

SLA documents following WS-Agreement use XML tags to define service level requirements. The same terms with different associated tags may generate different affections to the semantics of the documents. Since we are interested in SLA definitions regarding to services quality requirements, we focus on the following two tag types: `<wsag:Serviceproperties>` and `<wsag:GuaranteeTerm>`. In detail, we do a global search over an SLA document for the starting points of the two tags. If found, the total tree of the nodes in the SLA document with one of the two tags above will be fetched in a whole for further investigation.

The tag *ServiceProperties* is used to define service-oriented quantifiable properties, such as response time and throughout. It normally includes a sub-tag `<wsag:VariableSet>` to define a list of customized attributes and corresponding values. In the following very simplified example, the `<wsag:ServiceProperties>` tag defines one service level requirement *ResponseTime* delimited by the tag `<wsag:Variable>`.

```
<wsag:ServiceProperties wsag:Name="xs:AName" wsag:ServiceName="xs:MyService">
  <wsag:Variable name="ResponseTime" metric="job:ResponseTimeCount">
    <wsag:Location>
      //TaskDescription/Resources/IndividualResponseTime/Exact
    </wsag:Location>
  </wsag:Variable>
</wsag:ServiceProperties>
```

The tag *GuaranteeTerm* is constructed to define quantifiable assurance of service level quality, associated with the service described by the service level bounds, or Service Level Objectives (SLO). An SLA document may contain zero or more guarantee terms. We need to find all of these guarantee terms delimited by the tag `<wsag:GuaranteeTerm>`. In the following very simplified example, the `<wsag:GuaranteeTerm>` tag defines a requirement from a service requestor. The service requestor requests that the requirement be assured on any operations of the Web service *MyService*. A precondition (a date) is also defined as delimited by the tag `<wsag:QualifyingCondition>`.

```
<wsag:GuaranteeTerm Obligated="wsag:ServiceConsumer">  
  <wsag:ServiceScope ServiceName="xsd:MyService">  
    xsd:any  
  <wsag:QualifyingCondition>"01302006"</wsag:QualifyingCondition>  
</wsag:GuaranteeTerm>
```

Our SLA document parser fetches the segments, parses the content, and generates a tree-like structure with SLA requirements as internal nodes with associated values. In general, any XML parser-based tool can be used to fulfill the task. Our previous research yields a Web application code generator WebGen (Zhang and Chung 2003), which we used in this work as a document parser, so that we can easily add features and custom interface code to facilitate the SLA document analysis.

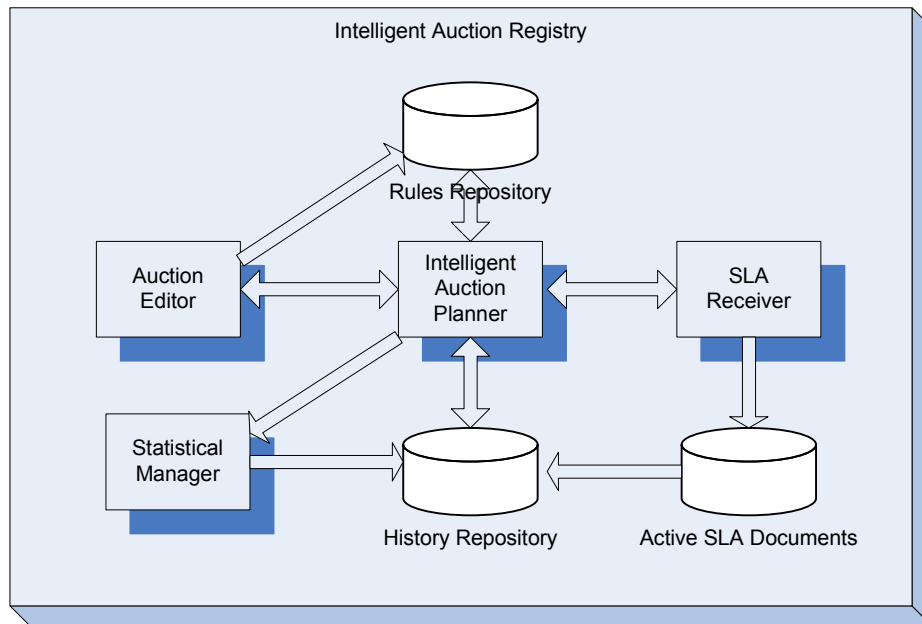
The SLA document developers may use different variants of a keyword to define the same meaning. Plurals, gerund forms, and past tense suffixes are examples of syntactical variants that may lead to divergences of one semantic concept. For instance, an SLA document may define a requirement on a service availability using different forms, e.g., “availability”, “available”, “availabilities”, etc. This problem can be partially resolved by substituting words with their respective stems. A stem is the portion of a word that is left after the removal of its affixes - prefixes and suffixes. For example, “availability” is the stem of “availabilities”. Comparatively, suffix removal is more important because most variants of a word are generated by the introduction of suffixes. There have been many algorithms in the literature regarding affixes removal. We decided to adopt the Porter stemming algorithm (Porter 1980) in our research due to its popularity, simplicity, and efficiency. The Porter algorithm, or so-called ‘Porter stemmer’, removes common morphological and in-flexional endings from English words for the purpose of term normalization. After the word stemming process, the tags of an SLA document are changed into a normalized form, all variants of a word are represented by its root. Thus, it is easier to capture the semantics of service specification from an SLA document. Note that the word stemming process should be applied to both service provider SLA documents and service requestor SLA documents, for consistency and precision.

## PROTOTYPE IMPLEMENTATION

We constructed an Intelligent Auction Registry (IAR) engine that is a prototype to help service auction providers make proper pricing decisions. Figure 2 illustrates the architecture and the components of our IAR engine. Four components are set up in the IAR engine, namely, auction editor, intelligent auction planner, statistical manager, and SLA document receiver. Three repositories are included in the engine: rules repository, active SLA repository, and history

repository. In order to help service providers better describe the bidding services they want to register, the auction editor provides a user interface to prompt service providers to input WSDL document and SLA document. The rule repository stores both the auction selling price selection algorithm and corresponding information, such as the parameters needed. Service providers can also define specific rules for a service auction and store them into the rule repository.

The intelligent auction planner component has five functionalities. First, it coordinates with the auction editor to get bidding service and associated service provider-side SLA document. Second, it manages the SLA receiver to monitor incoming service requestors and obtain their SLA documents. Third, it queries the history repository for the previous predicted risk preferences and actual risk preferences of each service requestor. Fourth, it starts the statistical manager to calculate and predict service requestor risk preferences. Fifth, it coordinates the auction process using the policies from the rules repository. IAR thus is able to provide pricing strategy suggestions to the seller. In order to decouple our engine from the underlying technology choices and make it easier to communicate with other third-party resources, we constructed our IAR engine as a Web service and use SOAP as its communication protocol. Because all queries are transmitted via XML, a universal format for structured documents and data definitions on the Web, our engine can be deployed to various platforms.



**Figure 2. Intelligent Auction Registry**

After the auction is performed, the statistical manager gathers information from the runtime environment and stores it in the history repository for future usages. The types of statistical data that needed to be built and maintained in order to provide the probability and timing data for the registry's decision process are every service requestor's SLA document, bidding price, predicted risk preference, and actual risk preference. These data can be in turn utilized to further verify the pricing selection algorithm.

The system has been under development. The agent, including the data-collecting capability described in this paper, is in place. The principle elements of the Intelligent Auction Registry



agent are complete, including a highly adaptable search engine designed to support bid evaluation in this environment.

## RELATED WORK

A comprehensive set of theories and techniques have been established in the traditional auction field. In recent years, enormous amount of research has been conducted in the field of electronic auction. Bajari and Hortacsu (2003) conduct a comprehensive survey on e-auction. A rich body of literature focuses on establishing efficient negotiation protocols in order to automate electronic auctions by customers and merchants (Byde 2003). Kikuchi (Kikuchi 2001) propose a new protocol for multiple distributed auctioneers to find the highest price and the set of winners; Hirakiuchi (Hirakiuchi and Sakurai 2001) propose a group of signatures and the identity escrow-based, anonymous sealed-bid auction protocol. Subramanian (Subramanian 1998) proposed a secure electronic auction protocol to favour security, privacy, anonymity, atomicity, and low overhead. Some researchers constructed electronic auction systems to explore various online auction frameworks. Among them, AuctionBot (Wurman, Wellman et al.) is an experimental Internet auction server established at the University of Michigan. AuctionBot provides a list of predefined auction types for sellers to choose from. Sellers can also customize related parameters. After an auction is set up, the system will enforce the multilateral distributive negotiation protocols. Panzieri and Shrivastava (Panzieri and Shrivastava 1999) replicate auction servers to achieve data integrity, responsiveness, and scalability. These specific protocols significantly extend the traditional physical auction model into online electronic auctions that provide more flexibility, dynamicity, friendly auction environments. These protocols enrich the traditional ones and can be used as starting points of Web services-oriented auctions.

There exist popular Web sites to support electronic auction activities, such as eBay (EBay 2005), onSale (OnSale), AuctionNet (AuctionNet), etc. In recent years, many known e-auction Web sites have started to adopt the Web services technology to extend and broaden their services. The most typical example is eBay. Its site offers a rich body of eBay APIs to allow user applications to directly communicate the eBay databases in XML format. In other words, the eBay APIs permit an easy construction of custom applications with eBay as the backend. Since 1999, more than 100 eBay APIs have been published to expose eBay Web services, such as pricing information, buy-it-now features, and payment options through its PayPal subsidiary.

Esmailsabzali and Larson (Esmailsabzali and Larson 2005) use the game theory to model Web services allocation. In their work, they map service providers to auction sellers, and service requestors to auction buyers but do not consider any Web services-specific features as auction constraints. In other words, they directly apply the existing auction models to Web services negotiation without any adjustments. Defining Web services auction as single-round auctions, they do not consider the re-enterable feature of Web services. Unlike traditional bidding objects that will transfer their ownership from sellers to winners after auctions end, bidding services will never change ownerships away from service providers. Therefore, we argue that Web services auctions should not be simply considered as the same as traditional auctions.

Zeng et al. (Zeng, Benatallah et al. 2004) propose a linear programming approach to maximize a service requestor's quality requirements based upon a vector of quality concerns. They present a global planning selection approach that not only takes into account multiple criteria (e.g., price, duration, reliability), but also global constraints and preferences set by service requestors (e.g., budget constraints). A pre-determined quality and price are assumed for a service requestor to

select among a group of service providers. Their work is completely from the computer science's perspective, without taking into consideration of essential economic dynamics of the services negotiation and selection.

Li et al. (Li, Chao et al. 2005) constructed a testing auction system using multi-agents-based Web services. Templates are defined as generic processes to enable seamless integration and collaboration among services within and across auction places. Zhu and Shan (Zhu and Shan 2005) use agent-oriented modeling language and abstract specification to guide an online auction system construction using Web services.

Researchers have been exploring applying the auction mechanism in Web service resource allocation. Vohra (Vohra 2001) proves that if users' preference are incorporated in the centralized resource allocation, searching optimal allocation in terms of economic efficiency is an NP complete problem. Huang et al. (Huang, Chen et al. 2005) propose a progressive auction mechanism for resource allocation that allows user differentiation of the service value with least resource acquisition latency. Their market-based service bidding model contains four types of components: auctioneer, user agent, market update server, and secured bank. The four roles map to service providers, service requestors, and service registry. Comparing with their work focusing on building a market-oriented resource allocation and pricing rules using progressive auction technique, our research focus on establishing a fundamental auction model in the context of Web services and explore how to obtain machine understandable information to predict auctioneers' risk preferences through their risky behaviors.

In summary, significant amount of research work has been conducted on affiliating the Web services technology and auction model. Some reported works adopt the basic format of auction as a negotiation method between Web service providers and requestors. However, since Web services-specific auction is not the focus of their research, they intuitively utilize the basic concept of auctions to realize price bargaining. They do not consider the specific features of Web services-based auctions; nor do they intend to build a formal model specific for Web services-based auctions. Some other researchers adopt the Web services technology to implement electronic auctions, such as eBay. Essentially they focus on utilizing SOAP messages as communication channels among auction participants. However, the electronic auction models are unchanged from typical electronic auction models. Contrast with either type of their work, our research aims at establishing a fundamental model for adopting auction model to facilitate Web services negotiation and interactions.

Our earlier research (Zhang, Zhang et al. 2004) explored the seller side pricing strategy on e-auctions. We examined the possible pricing strategies based upon how they can differentiate seller's profits. The work was conducted in the context of online electronic human-oriented auction. Compared to our previous work, this paper reports our continuous research on applying the auction model to interoperable machine-oriented Web services field. We formally define and build a Web services-based auction model. Possible pricing strategies are examined in the context of Web services subject to specific constraints and requirements in the field. In addition, instead of using human background to predict buyers' risk preferences, we propose to analyze WS-Agreement-based SLA documents to automatically elicit service requestors' risk preferences, and propose a dynamical model to adjust predictions based upon historical data.

## **CONCLUSIONS**

Conducting auctions in the context of Web services introduces unprecedented requirements into the traditional auction model, such as interoperable machine-to-machine interactions, re-enterable bargaining, independent bidding, etc. This deeply explored field thus needs re-examination; and new auction models tailored in the field of Web services is on demand. In this paper we proposed a formal model of Web services-based auctions. We examined one-sided sealed auction type, a popular auction model that could be applied to Web services-based auctions. We proved mathematically that service providers need to adopt different pricing strategies to pursue higher profit based upon service requestors' risk preferences. We argue that Service Level Agreement (SLA) documents can be used to analyze, quantify, and predict service requestors' preferences. On top of WS-Agreement, we proposed a basic service requestor risk preference elicitation algorithm and a historical data-based service requestor risk preference prediction model. We reported our construction of a prototype of an intelligent engine that helps service providers to select optimal auction strategy for higher profit, which provides a proof-of-concept of our services-oriented auction model.

In order to simplify the discussions of our auction model, we focus on one-sided sealed auction type. Further work will widen the understanding of Web services-oriented auction model. Furthermore, at current stage, we do not consider the overhead and efficiency of the risk preference prediction model and algorithms. This topic will be pursued in the future work. However, the work reported in this paper is an important starting point. To our best knowledge, our work is one pioneer work in the topic of applying auction model into the Web services field. A formal services-oriented model is established and its usages and applicability are shown. A history data-based risk preference prediction model is presented. In addition, we extend and digitize the traditional abstract concept of risk preference into an interoperable, machine understandable and processable, quantifiable format based on industry standard-based SLA specifications. The methods and techniques presented in this paper can be reused to investigate and examine more facades of services-oriented auctions, towards establishing a new research realm on comprehensive services-oriented auctions.

We plan to continue our research along the following directions. We will examine how a service provider's risk preference type influences its auction pricing set up. We will also investigate what kinds of marketing promotions a service provider should take to attract a larger range of potential service requestors. We will enhance the GUI of the intelligent auction registry, and explore how to increase its intelligence on auction strategy prediction and decision. Finally we will closely examine WS-Agreement and formally define extensions for better facilitating service level requirements specifications oriented to Web services auctions.

## REFERENCES

- Arrow, K. J. (1963). "Comment." *Review of Economics and Statistics*, XLV: 24-27.
- Arrow, K. J. (1965). *Aspects of the Theory of Risk-Bearing*, Helsinki.
- AuctionNet <http://www.auction.net>.
- AuctionWeb <http://www.auctionweb.com/online>.
- Byde, A. (2003). "Applying Evolutionary Game Theory to Auction Mechanism Design." *Proceedings of IEEE International Conference on E-Commerce*, Jun., 347-354.
- EBay (2005). <http://www.ebay.com>.
- Esmailsabzali, S. and Larson, K. (2005). "Service allocation for composite Web services based on quality attributes." *Proceedings of the 2005 Seventh IEEE International Conference on E-Commerce Technology Workshops (CECW)*, Munich, Germany, Jul. 19, 71-79.

- ETrade (2006). <https://us.etrade.com>.
- Ferris, C. and Farrell, J. (2003). "What Are Web Services?" *Communications of the ACM*, Jun., 46(6): 31.
- Fudenberg, D. and Tirole, J. (1991). *Game Theory*, MIT Press, MA.
- Gartner (2003). "Composite Applications Head Toward the Mainstream." Oct. 16. <http://www.gartner.com>.
- GGF "Global Grid Forum." <http://www.ggf.org>.
- Hirakiuchi, D. and Sakurai, K. (2001). "English vs. Sealed Bid in Anonymous Electronic Auction Protocols." *Proceedings of Tenth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Jun., 171-176.
- Huang, S., Chen, H. and Zhang, L.-J. (2005). "Progressive Auction Based Resource Allocation in Service-Oriented Architecture." *Proceedings of the 2005 IEEE International Conference on Services Computing (SCC)*, Orlando, FL, USA, Jul. 11-15, 2, 85-92.
- IBM (2003). "Web Service Level Agreement (WSLA) Language Specification." Jan. 28, 2003. Version 1.0. <http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf>.
- Kikuchi, H. (2001). "Power Auction Protocol Without Revealing Winning Bids." *Proceedings of The 15th International Conference on Information Networking (ICOIN'01)*, Jan., 61-64.
- Klemperer., P. (1999). "Auction Theory: A Guide to the Literature." *Journal of Economic Surveys*, 13(3): 227-286.
- Krishna, V. (2002). *Auction Theory*, Academic Press Inc.(London) Ltd., London, UK.
- Li, Y., Chao, K.-M., Younas, M., Huang, Y. and Lu, X. (2005). "Modeling E-marketplaces with Multi-agents Web Services." *Proceedings of the 2005 11th International Conference on Parallel and Distributed Systems (ICPADS)*, Fukuoka, Japan, Jul. 20-22, 175-181.
- Mas-Colell, A., Whinston, M. and Green, J. (1995). *Microeconomic Theory*, Oxford University Press.
- McAfee, R. P. and McMillan, J. (1987). "Auctions and Bidding." *Journal of Economic Literature*, 25: 699-738.
- OnSale <http://www.onsale.com>.
- Panzieri, F. and Shrivastava, S. K. (1999). "On the Provision of Replicated Internet Auction Services." *Proceedings of 18th IEEE Symposium on Reliable Distributed Systems*, Oct., 390-395.
- Porter, M. (1980). "An Algorithm for Suffix Stripping Program." *Automated Library and Information Systems*, 14(3): 130-137.
- Roy, J. and Ramanujan, A. (2001). "Understanding Web Services." *IEEE IT Professional*, Nov.: 69-73.
- Subramanian, S. (1998). "Design and Verification of a Secure Electronic Auction Protocol." *Proceedings of The 17th IEEE Symposium on Reliable Distributed Systems*, Oct., 204-210.
- Vohra, R. V. (2001). "Research Problems in Combinatorial Auctions." *Proceedings of DIMACS Workshop on Computational Issues in Game Theory and Mechanism Design*, Rutgers University, Piscataway, Nov.
- WS-Agreement (2005). "Web Services Agreement Specification (WS-Agreement)." Sep. 20. [http://www.ggf.org/Public\\_Comment\\_Docs/Documents/Oct-2005/WS-AgreementSpecificationDraft050920.pdf](http://www.ggf.org/Public_Comment_Docs/Documents/Oct-2005/WS-AgreementSpecificationDraft050920.pdf).
- Wurman, P. R., Wellman, M. P. and Walsh, W. E. "The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents." *Proceedings of the Second International Conference on Autonomous Agents (Agent-98)*, Minneapolis, MN, USA, May., 301-308.
- Wurman, P. R., Wellman, M. P. and Walsh, W. E. (1998). "The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents." *Proceedings of the Second International Conference on Autonomous Agents (Agent-98)*, Minneapolis, MN, USA, May, 301-308.

- Zeng, L., Benatallah, B., Ngu, A. H. H., Dumas, M., Kalagnanam, J. and Chang, H. (2004). "QoS-Aware Middleware for Web Services Composition." *IEEE Transactions on Software Engineering*, May, 30(5): 311-327.
- Zhang, J. and Chung, J.-Y. (2003). "Mockup-driven Fast-prototyping Methodology for Web Application Development." *Software Practice & Experience Journal*, 33(13): 1251-1272.
- Zhang, J., Zhang, N. and Chung, J.-Y. (2004). "Assisting Seller Pricing Strategy Selection for Electronic Auction." Proceedings of IEEE Conference on Electronic Commerce (CEC 2004), San Diego, CA, USA, Jul. 7-9, 27-33.
- Zhu, H. and Shan, L. (2005). "Agent-Oriented Modelling and Specification of Web Services." Proceedings of the 10th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS), Sedona, AZ, USA, Feb. 2-4, 152-159.

## AUTHOR INFO

**Jia Zhang, Ph.D.**, is an Assistant Professor of Department of Computer Science at Northern Illinois University. She is now with BEA Systems Inc.; and also a Guest Scientist of National Institute of Standards and Technology (NIST). Her current research interests center around software trustworthiness in the domain of Web services, with a focus on reliability, integrity, security, and interoperability. Zhang has published over 60 technical papers in journals, book chapters, and conference proceedings. She also has seven years of industrial experience as software technical lead in Web application development. Zhang is an Associate Editor of the International Journal of Web Services Research (JWSR). She is serving as the Program Vice Chair of IEEE International Conference on Web Services (ICWS 2006). Zhang received a Ph.D. in computer science from University of Illinois at Chicago in 2000, and a M.S. and a B.S. in computer science from Nanjing University in 1994 and 1991, respectively. She is a member of the IEEE and ACM. Contact her at [jjazhang@cs.niu.edu](mailto:jjazhang@cs.niu.edu).

**Ning Zhang** is currently a Ph.D. candidate in the Department of Policy Analysis and Management of Cornell University. Her current research interests lie in health economics, labor economics, and Internet economics. She earned her Master degree from the Cornell University and Bachelor degree from Nanjing University, China. Contact her at [nz33@cornell.edu](mailto:nz33@cornell.edu).

**Liang-Jie Zhang, Ph.D.**, is a Chief Architect of Industrial Standards of IBM Software, the founding chair of the Services Computing Professional Interest Community (PIC), and a research staff member at the IBM T.J. Watson Research Center. He is part of the Business Informatics research team with a focus on SOA and Web services for industry solutions and business performance management services. He has filed more than 30 patent applications in the areas of e-commerce, Web services, rich media, data management, and information appliances, and he has published more than 80 technical papers in journals, book chapters and conference proceedings. Dr. Zhang is the chair of the IEEE Computer Society's Technical Committee on Services Computing and the Editor-in-Chief of the International Journal of Web Services Research (JWSR). He serves as the General Co-Chair of the IEEE International Conference on Web Services (ICWS 2004-2006) and the General Co-Chair of the IEEE Conference on Services Computing (SCC 2004-2006). Liang-Jie received a B.S. in Electrical Engineering at Xidian University in 1990, an M.S. in Electrical Engineering at Xi'an Jiaotong University in 1992, and a Ph.D. in Computer Engineering at Tsinghua University in 1996. Contact him at [zhanglj@us.ibm.com](mailto:zhanglj@us.ibm.com).