A Policy-based Middleware for Web Services SLA Negotiation

Farhana Zulkernine and Patrick Martin
School of Computing,Queen's University
Kingston, ON K7L 3N6, Canada
{farhana, martin}@cs.queensu.ca

Chris Craddock, SVP
OOCTO, CA, Inc.
Houston TX
Chris.Craddock@ca.com

Kirk Wilson, Ph.D
CA, Inc., 273 Corporate Drive, Portsmouth, NH
Kirk.Wilson@ca.com

Abstract

Negotiation of Service Level Agreements (SLAs) is very important for maintaining Quality of Service (QoS) of composite Web services-based business processes. The process of negotiation involves specification of negotiation parameters, exchanging offers to conduct the actual negotiation process, and then finally generating the formal SLA if the negotiating parties come to a consensus. We propose a Negotiation Broker (NB) middleware framework to facilitate automated negotiations of SLAs for Web services in a Service Oriented Architecture (SOA). High level business goals, contexts, preferences, constraints, and values of the negotiation issues are expressed as a policy specification by each of the negotiating parties. The NB maps the policy specifications to low level negotiation strategy models and parameters in order to conduct the negotiation locally as a trusted broker. We present a model and an example of the high level negotiation policy specification. We also present our NB framework including a prototype implementation to illustrate the mapping of the policy to a time-dependent negotiation strategy model.

1. Introduction

In business relationships, interests of the parties involved are protected by legal contracts, which are the Service Level Agreements (SLAs) in the case of Web services. SLAs generally define service attributes, consumer requirements, deliverables, performance levels, rewards and penalties in terms of pricing and usage. Multiple Web services can be orchestrated to create multi-organizational business processes on demand, and for that, efficient negotiation of the SLAs is essential.

Negotiation has been an interesting area of research since 1960s [2]. Researchers have applied combinations of different models and technologies from different problem domains such as economic models (utility functions and cost-benefit models) [10], genetic algorithms [1], game theory [11], learning approaches and decision models [20] from artificial intelligence, and intelligent agents [19] to define various negotiation approaches. Negotiations can be between two (*bilateral*) or more (*multilateral*) parties, and can involve one or more issues. The parties typically communicate using a specific *negotiation protocol* (rule for exchanging messages) until they reach a common consensus regarding the values of the different issues.

Each negotiating party follows a *tactic* to generate the next value to propose to the other parties for a single issue (e.g. price, availability, number of users) based on some criteria (e.g. preference, constraints, desirability) [10]. Weights are used to vary the importance of the different criteria and issues to be modeled, which may change over time. The way the weights of different tactics change over time during the negotiation process is called *strategy*. Based on the information provided by the negotiating parties, a *Decision Support System (DSS)* defines an appropriate *Decision Model* for each party, which in turn, applies various decision making tools and methodologies to define the tactics and strategy.

Negotiation in the case of Web services is usually bilateral between the service provider and the service consumer, and involves one or more issues. The service offering (details of a service provided by a service provider) plays an important role in SLA negotiation. Typically a Web service is offered as different priced packages, for example Gold, Silver, and Bronze as shown in Table 1. Each package may include some service parameters that have fixed values (response time) and some that have negotiable values (availability, price, and

Table 1: Stock Quote service offerings

Option Type	Options in Offer	Package Offers		
		Gold	Silver	Bronze
Fixed	▪ Response Time ▪ Bonus	1 s. 10 free	2 s. 5 free	3 s. 0
Negotia-ble	▪ Price (per month) ▪ Number of users ▪ Availability	30$ - 50$ 500 (max) 98.9-99.9%	20$ - 30$ 350 (max) 97.9-98.9%	10$ - 20$ 200 (max) 96-97.9%

users), and other complimentary service parameters (bonus). While the service consumers may want a certain range of availability for a specific price range, service providers may also offer special discounts to specific category of consumers based on their context information (e.g. large organizations, location) or business potential (e.g. economic value, length of contracts). An automated negotiation system [2] can be very effective in such cases where there are limited numbers of specific negotiable issues. SLA negotiation is an important part of building Web services-based business processes. We propose Negotiation Broker (NB) framework as a part of the Comprehensive Service Management Middleware (CSMM) [25], which is a framework to support the main aspects of client-side management of Web service-based processes. In this paper, we present the detail architecture of the NB framework, which enables fast, secure, and effective policy-based negotiation.

Repeated communication over the Internet during negotiation requires both time and bandwidth. Application of encryption schemes for security and privacy concerns adds additional complexity and overhead of transforming the messages exchanged between the parties. Our NB facilitates local execution of policy-based negotiation using intelligent agents. The paper contributes to the specification of high level negotiation policy by the business administrators, mapping of the policies to appropriate strategy model and its parameters, and application of intelligent agents to conduct impartial local negotiation in the broker middleware. We present a preliminary prototype implementation of the framework to illustrate the validity of our approach using a time-dependent cost-benefit negotiation strategy model for a multi-issue bilateral bargain negotiation.

The rest of the paper is organized as follows. Section 2 presents related research work. The methodology behind our policy-based negotiation middleware is presented in Section 3. The architecture of the NB middleware is described in Section 4 followed by an illustration of the prototype implementation of the framework for the example given in Table 1. Section 5 discusses future work and concludes the paper.

2. Related Work

With the advancement in Web media, automation of the negotiation process has gained much attention. Different models of Negotiation Support Systems (NSSs) [9][16] have been proposed, which assist human negotiators in making decisions in a negotiation process. With the advent of Service Oriented Computing (SOC), researchers have turned their focus to negotiations for e-Services [4][5][15][18][22].

Su *et al.* [21] propose a negotiation server for e-commerce to perform bargaining-type negotiations automatically. Each negotiating party registers with a negotiation server and provide their goals, contexts, requirements and priorities, and the servers then conduct negotiation automatically using constraint satisfaction, rule-based conflict resolution, and event systems. Hung *et al.* [15] propose WS-Negotiation language (contains negotiation message, protocol, and strategy) and an overview of a framework for negotiation by two Web services over the Web. We use standard-based policies with broker middleware to execute negotiation locally.

Li *et al.* [18] propose an automated negotiation framework based on a finite state automata and a set of negotiation protocols. The framework maps negotiation context to negotiation goals using policies and the goals are mapped to negotiation rules and plans using negotiation strategy, to carry out bilateral bargaining. The framework is en extension of the work of Su *et al.* [21]. We use a more general policy specification language and broker-based approach to negotiation.

Comuzzi and Pernici [4][5] propose a policy-based negotiation broker framework similar to the NB to perform partially or fully automated negotiation of QoS parameters for service selection. However, the negotiating parties need to have knowledge about the strategy models supported by the framework to mention their choice of strategy and parameters in the policy specifications. Our framework is different with respect to architecture, policy specification, and features like support for knowledge-based strategy models and external feedback during negotiation.

Yee and Korba [22] propose a scheme for negotiation of e-services under uncertainty using existing records of similar negotiations. In their scheme, a participant who is negotiating in uncertainty obtains assistance in the form of negotiation alternatives and offers made, from other reputable participants who have negotiated the same issue. We build a negotiation knowledgebase in the NB to accommodate this aspect in the future.

Gimpel *et al.* [13] propose Policy-driven Automated Negotiation Decision-making Approach (PANDA), where a policy expresses a party’s private negotiation strategy as a combination of rules and utility functions. In their approach, the decision making problem is decomposed into multiple aspects. Each aspect is handled by a separate Decision Maker (DM) framework, which interact with each other to jointly provide a solution. In NB, each agent pair conducts separate negotiation and multiple pairs of agents can simultaneously conduct different negotiations.

Brzostowski *et al.* [3] describe a multilayered approach to negotiation of SLAs for Web service compositions by three decision making steps. These steps are decomposing a user-defined process level utility function of overall preferences into lower level negotiation preferences for service selection; selecting a set of candidate services from a larger set of potential service providers, and finally negotiating with the set of candidate services to select the service that yields the highest utility value to meet the end-to-end process QoS. The process repeats with an adjusted requirement set if no service can be selected. The approach uses regression analysis [14] for learning opponent's behavior and makes concessions accordingly to maximize its own utility. We use utility functions and a cost-benefit strategy model. The NB works as a part of the CSMM [25] to enable re-negotiation.

3. Methodology

We use a policy-based approach to define the parameters for negotiation. The three most important aspects of a policy-based negotiation system are the *negotiation policy, negotiation protocol*, and the *decision support system (DSS)*. In this section we first describe the policy model, its contents, and the formal specification in detail. Next we define the negotiation protocol and the decision support system used in our framework.

3.1. Negotiation Policy

Policies are basically sets of high level governing rules, which define assertions or actions to be taken when certain conditions are met, and thus guide the decision making process to achieve certain goals. A number of XML-based policy and negotiation languages [1][15][23] have been proposed by different researchers and a standardization effort is ongoing (Policy Language Interest Group [24]). Most of these languages address specific domains and have specific requirements in terms of processing and usability. We use the WS-Policy [23] standard in the NB since it is more general, simple, and an XML-based standard.

Figure 1 shows a UML (Unified Modeling Language) diagram of the policy model used in the NB. Policy contents can be very specific to different domains. We use a Domain Specific Schema (DS-schema) to describe the tags used to define our domain specific policy within the WS-Policy framework. Data types other than the basic ones are also included in the DS-schema. A detailed

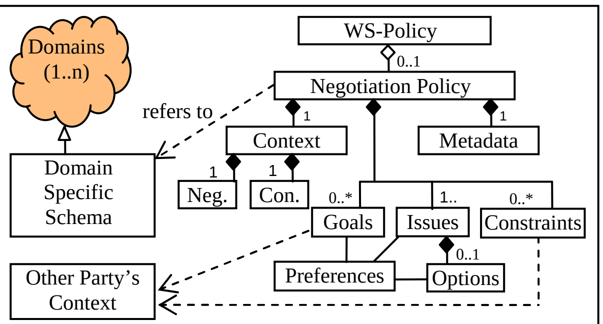

Figure 1: Contents of policy specification

policy specification of a Stock Quote service provider is given in Figure 3. Within the WS-Policy wrapper, there can be 0 or more negotiation policy specifications, which in turn can contain context, goals, issues, constraints and other entities and subclasses as explained below.

For our preliminary prototype implementation, we consider a very limited set of values for each of the negotiation parameters. An ontology can be included to provide a more general implementation.

Context: Three types of context information are referred to in the policy specification: *NegotiationContext* (line [3] in Fig.3[*]) refers to the specific context to which the policy applies; C*onsumerContext* (line [16]) refers to the parties' context information (e.g. location, for personal or business use, size of the company), and *OtherContext* (line [5]) refers to the context of the other negotiating party. For example, the service provider may be more interested (indicated by a higher desirability factor) if the other party is a large company with a good credit record rather than an individual customer with unknown credentials.

Goals: Each party in the negotiation can specify multiple goals (line [4]). High level goals, such as maximizing profit or the number of users, obtaining long or short term contracts, or targeting specific consumer groups or large reputable companies are based on the strategic business plan of the service providers. Service consumers may define short term contracts or lowest price as their goals, or may choose not to specify any goal. Based on the higher level goal, detailed policy specifications can include further details in the form of rules regarding how to achieve the goal. This kind of hierarchical organization of policy specification can be very useful in large organizations where different levels of policy can be specified by a manager at the corresponding level.

Issues and Options: Issues (line [7]) are the negotiable parameters in a service offering and options (line [9]) are the different values that the negotiable parameters can have. The negotiating parties need to specify the *best* (line [10]) and *worst* (line [11]) acceptable values for each issue and option, the *preference* weight (normalized value of all issues to indicate relative importance) (line [8]), and optionally a *threshold* value (if exceeded external resource is contacted for decision making) (line [12]). For example, to make an offer containing 99.7% availability that is beyond the 99.5% threshold value, the service provider should be contacted to verify the feasibility of that offer.

Constraints: Constraints (line [13]) are combinations of conditions (line [14]) that define unacceptable value sets of multiple issues and are declared explicitly in a policy specification to filter out unacceptable solutions.

[*] All line numbers in this section refer to Fig. 3

For example, a combination of price greater than $40 and availability less than 99.4% is unacceptable. The maximum time for negotiation (line [15],[18]) can also be optionally set as a constraint. The NB sets the maximum negotiation time to be either the minimum of the two parties' times, if specified, or the usual average negotiation time.

Preferences: The negotiating parties can define comparative normalized preference values (line [6],[8]) for the different goals when more than one goal applies to facilitate conflict resolution and for issues and options for tradeoff purposes.

Metadata: The metadata (line [17]) contains information about the consumer who invokes the service of the NB, the name and date of the policy specification for easy reference, and a current *DesirabilityFactor (DF)* (line [19]) which indicates the consumer's desirability to reach an agreement.

3.2. Negotiation Protocol

Researchers have proposed several protocols for bilateral bargains over the network [5][12][18]. For the localized negotiation in the NB, we use only a small subset of the FIPA Contract Net Interaction Protocol (CNIP) [12] as shown in Figure 2. *Call-For-Proposal (CFP)* (directed arrow from the consumer to the provider) is used to request an offer from the service provider. *Propose Proposal* (bi-directional arrow) is used by either party for communicating offers and counter-offers. *Refuse Proposal* indicates a party's unwillingness to participate in negotiation. A successful negotiation ends with *Accept Proposal* while failure to reach an agreement is indicated by a *Reject Proposal*. *Failure* indicates inability to receive, transmit, or interpret a message as explained in the message content, which may be replied by the other party by resending the previous message.

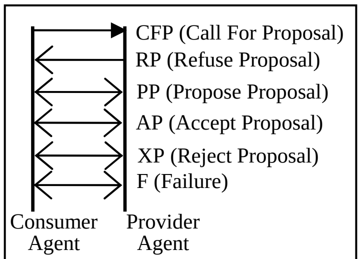

Figure 2: Negotiation Protocol

3.3. Negotiation Decision Support System (DSS)

The DSS determines the possible combination of issues, values of the issues based on the previous offer, considers the policy, and determines the best values for the issues to offer next to achieve the negotiation goal. Since both parties usually benefit from the negotiation, we apply the integrative or collaborative negotiation in our

```
<wsp: Policy xmlns:wsp=                                          [1]
    "http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:nb="http:// CSMM.nb/policy">
 <nb:NegotiationPolicy>                                          [2]
  <nb:NegotiationContext>                                        [3]
   <nb:Role>ServiceProvider</nb:Role>
   <nb:Service>StockQuoteGold</nb:Service>
  </nb:NegotiationContext>
  <nb:Goals>                                                     [4]
   <nb:Goal><nb:Target>Maximize_Profit</nb:Target>
    <nb:OtherContext>personal</nb:OtherContext>                  [5]
    <nb:Preference>1</nb:Preference>                             [6]
   </nb:Goal>
   …
  </nb:Goals>
  <nb:Issues>                                                    [7]
   <nb:Issue>
    <nb:Name>Price</nb:Name>
    <nb:Type>Decimal</nb:Type>
    <nb:Unit>Dollar</nb:Unit>
    <nb:Preference>0.6</nb:Preference >                          [8]
    <nb:Option><nb:Name>Gold</nb:Name>                           [9]
     <nb:Bestval>50</nb:Bestval>                                 [10]
     <nb:Worstval>30</nb:Worstval>                               [11]
    </nb:Option>
   </nb:Issue>
   <nb:Issue>
    <nb:Name>Availability</nb:Name>
    <nb:Type>Decimal</nb:Type>
    <nb:Unit>Percentile</nb:Unit>
    <nb:Preference>0.4</nb:Preference>
    <nb:Option><nb:Name>Gold</nb:Name>
     <nb:Bestval>98.9</nb:Bestval>
     <nb:Worstval>99.9</nb:Worstval>
     <nb:Thresholdval>99.5</nb:Thresholdval>                     [12]
    </nb:Option>
    …
   </nb:Issue>
  </nb:Issues>
  <nb:Constraints><nb:Constraint><wsp:Policy>                    [13]
    <wsp:All>
     <nb:Condition><nb:Issue>Price</nb:Issue>                    [14]
      <nb:Operator>&gt;</nb:Operator>
      <nb:Value>40</nb:Value>
     </nb:Condition>
     <nb:Condition><nb:Issue>Availability</nb:Issue >
      <nb:Operator>&lt;</nb:Operator>
      <nb:Value>99.4</nb:Value>
     </nb:Condition>
     <nb:MaxNegTime>10</nb:MaxNegTime>                           [15]
    </wsp:All></wsp:Policy>
  </nb:Constraint></nb:Constraints>
  <nb:ConsumerContext>                                           [16]
   <nb:Location>Canada</nb:Location>
   <nb:EntityType>Company</nb:EntityType>
   <nb:Size>Medium</nb:Size>
  </nb:ConsumerContext>
  <nb:Metadata>                                                  [17]
   <nb: PolicyName>…</nb:PolicyName>
   <nb:PDate>…</nb:PDate>
   <nb:CustomerInfo>...</nb:CustomerInfo>
   <nb:MaxNegTime>...</nb:MaxNegTime>                            [18]
   <nb:DesirabilityFactor>0.7</nb:DesirabilityFactor>            [19]
  </nb:Metadata>
 </nb:NegotiationPolicy>
</wsp:Policy>
```

Figure 3: Negotiation Policy Specification

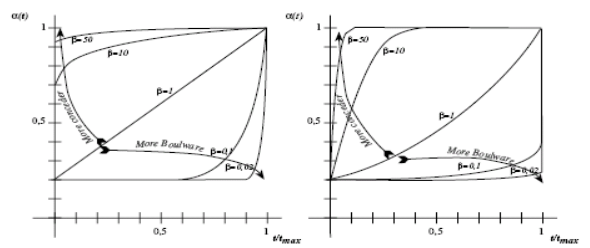

Figure 4: Polynomial (left) and Exponential (right)

NB framework where both parties reach a win-win situation mainly due to the reduced probability of failure.

Researchers have chosen behavioral models, game theory and genetic theory with different learning methodologies to define negotiation strategies. Game theory assumes disclosure of the opponent's information and goal, which is not realistic in business negotiations. Genetic algorithms have shown promising results in several instances but require considerable time for the whole process, which is not feasible in real world business processes. We collect the negotiation history in our framework to apply learning approaches in future research.

We use a time-dependent cost-benefit model for the negotiation strategy for both parties in our preliminary prototype. The model uses a parametric utility function as given below in equation (1) and as shown in the graph on

$$\alpha_j^a(t) = e^{\left(1 - \frac{\min\left(t, t_{\max}^a\right)}{t_{\max}^a}\right)^{\beta} \ln k_j^a} \qquad (1)$$

$$k_j^a < \alpha_j^t(t) \le 1, \textit{ where } 0 \le t \le t_{\max}^a \textit{, and}$$

$$k_j^a = x_j^a[0] \textit{ (const at } t = 0\textit{)}$$

the right in Figure 4. Faratin *et al.* [10] suggested two families of functions as shown in Figure 4, which have different conceding patterns based on the β value. We chose the exponential function because of its conceding pattern. Equation (2) maps the preference weights of the

$$\beta = \frac{\delta \;*\; DF}{pref} \qquad (2)$$

issues and the desirability factor (DF) from the policy specification into a β value for each issue where δ is a constant (0< δ <1). The function was first chosen based on the influence of each component on the next offer and then refined through trial and error.

We list below the other decision functions used in the NB. Detailed explanations of these functions can be found in [10]. The three major aspects of a decision model are: convergence of offers towards either timed termination of the process, or acceptance or rejection; decision regarding the next action (accept, reject, or make counter-offer), and finally definition of a mathematical model for generating counter-offers. We show that our model supports all three aspects as we define the various equations. An offer from agent a to agent b at time t consists of the values of n negotiable issues $x = <x_1,.., x_n>$ and can be expressed as:

$x_{a\to b}^t$ where the acceptable range of x_i for agent a is:

$$x_i \in [\min_i^a, \max_i^a]$$

In a service oriented architecture, clients and servers would commonly have conflicting interests. For example, a client desires a lower price for a service whereas a service provider likes to get a higher price. Issues for which both parties have mutual interests are generally excluded from the negotiation process by selecting maximum possible values for those issues. The goodness of an offer is judged by its utility value. The total utility value of an offer at time t containing i issues is computed from the weighted sum of the utility values of all the issues as given below. The weights represent the relative importance of the issues.

$$V^a(x_{a\to b}^t) = \sum_{1\le i\le n} w_i^a V_i^a\left(x_i\right) \qquad (3)$$

Here,

$V^a(x_{a\to b}^t)$ is the total utility value of the offer $(x_{a\to b}^t)$

w_i^a is the weight associated with i^{th} issue where $\sum_{1\le i\le n} w_i^a = 1$

$V_i^a(x_i) \in [0,1]$ is the utility value for the i^{th} issue

The convergence aspect of negotiation is satisfied by a timeout constraint to ensure termination of the process in a finite period. The second aspect, i.e., acceptability criterion for an offer is defined using its utility value. An agent a inferences at time t' about its next action based on the offer received at time t $(t < t')$ using the following logic equations:

$$I^a(t', x_{b\to a}^t) = \begin{cases} reject \textit{ If } t' > t_{\max}^a \\ accept \textit{ If } V^a(x_{b\to a}^t) \ge V^a(x_{a\to b}^{t'}) \\ x_{a\to b}^{t'} \textit{ otherwise} \end{cases} \qquad (4)$$

where,

$x_{b\to a}^t$ is the offer a received from b at time t

$x_{a\to b}^{t'}$ is the offer a should make to b at time t'

$t_{\max}^a$ is the const. max. time for negotiation for a

The utility value for an issue is computed using equations (5) or (6). The third negotiation aspect is addressed by using equations (7) or (8) to calculate the values of the different issues to generate counter-offers.

$$V_j^a = \begin{cases} \dfrac{\max_j^a - x_{a\to b}^t[j]}{\max_j^a - \min_j^a} & V_j^a \text{ increases as x decreases} \quad (5) \\ \dfrac{x_{a\to b}^t[j] - \min_j^a}{\max_j^a - \min_j^a} & V_j^a \text{ increases as x increases} \quad (6) \end{cases}$$

$$x_{a\to b}^t[j] = \begin{cases} \min_j^a + \alpha_j^a(t)(\max_j^a - \min_j^a) & \text{If } V_j^a \text{ is decreasing} \quad (7) \\ \min_j^a + (1-\alpha_j^a(t))(\max_j^a - \min_j^a) & \text{If } V_j^a \text{ is increasing} \quad (8) \end{cases}$$

4. Negotiation Broker (NB) Framework

We present a middleware framework as shown in Figure 5, which provides a trusted broker service for SLA negotiation through a *Negotiation Broker Web Service (NBWS)* endpoint. The negotiating parties invoke the NB with the policy specifications. Policies can also be collected by the NB from a service provider when negotiation request is obtained from the corresponding service consumer.

The process of negotiation is divided into three phases, *pre-negotiation*, *negotiation*, and *post-negotiation*. In the *pre-negotiation* phase, the negotiation policies received by the NBWS are processed by the *Policy and Context Pre-processor (PCP)* which stores the policies into a local policy database (PolicyDB) for easy retrieval and use. Afterwards simple policy updates and retrievals can be performed by the NBWS without involving the PCP. The negotiating parties can use policy references (containing policy name, date, URL and name of the party) to update or re-use an existing policy.

At this point, the *Decision Support System (DSS)* uses the pre-processed policy information from the PolicyDB and if necessary, the negotiation history from the *Negotiation Knowledgebase (NegKB)* to choose an appropriate negotiation strategy from the *Negotiation Strategy Model (NSM)* for each negotiating party. Then the DSS computes the necessary parameters of the strategy models from the policy information. For the cost-benefit strategy, the equations described in the previous section are used.

Next a *Decision Model (DM)* for each party is initialized by the DSS, which includes the corresponding negotiation strategy model; protocol; maximum time for negotiation; DF; best, worst, preference, and threshold values of the issues; constraints, and other rules necessary to guide the decision process. These rules may include goal-based decisions, rules for tradeoff or modifying tactic during negotiation, and modified constraints for the issues added when threshold values are exceeded.

Once the decision models are defined, the *Agent Factory (AF)* is notified, which creates an autonomous agent for each negotiating party. The agents execute the negotiation phase and conduct the negotiation independently based on their own decision models. This approach reserves the privacy of policies provided by each party and enables impartial negotiation. A *Negotiation Process Manager (NPM)* receives notifications from the agents about the status of the negotiation process and the offers exchanged, which are stored in a *Negotiation Knowledgebase (NegKB)*. The *External Resource Monitor (ERM)* waits for notifications from the agents to communicate with the corresponding parties when the threshold values specified in the policies are exceeded during the negotiation process. The parties at this point have the opportunity to redefine the values of the negotiation issues, update the constraints, or guide the decision for the next step, which is a powerful feature of the NB framework.

The context information of the parties obtained with the policy can also be stored in a separate context database within the NegKB to use later as a reference to resolve ambiguity in case of uncertainty or to apply

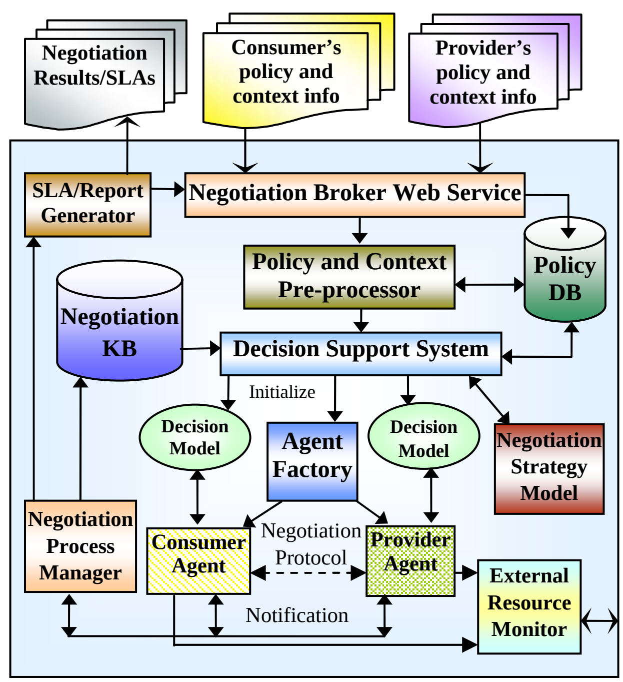

Figure 5: Framework of the Negotiation Broker

learning strategies. We assume that the context information provided by the parties is authentic since it is also used to define their personal negotiation strategies.

Upon completion of a successful negotiation, the NB enters the post-negotiation phase. The NPM sends the necessary information to a *SLA/Report Generator*, which generates a set of formal SLAs for the NBWS to send out to both parties. Several SLA specification languages have been proposed by researchers [8] but none has been standardized so far. We use an XML expression of the SLAs but the module can be designed to generate any preferred form of SLA specification.

4.1. Prototype Implementation

We implemented a prototype of the system to validate the framework and the decision model. We use the example given in Table 1 and the policy specification shown in Figure 3 with all the issues and options. We use IBM DB2 for the databases in the NB framework, IBM Intel Pentium 2.66 GHz desktops with 512 MB of RAM and Windows XP 2002 SP2 to develop and execute the framework. Java with JRE 1.4 is used to develop the software and program threads are used to simulate the agents. The different modules in the NB are all implemented as Java classes.

A higher value of $x[0]$ in equation (1) results in larger changes in the values of the issues ($e^{\ln x}$ increases slowly when $\ln x<0$ or $0<x<1$). We use equation (9) where λ is a constant, to calculate the initial value of α in equation (5).

$$k_j^a = x_j^a[0] = \lambda * \frac{x_{\max}^a - x_{\min}^a}{x_{\max}^a * t_{\max}^a} \qquad (9)$$

We ran experiments for different values of preferences for issues, DF, and maximum time for negotiation with the service offering shown in Table 1. All the experiments completed within the maximum time, which satisfies the convergence aspect of negotiation. The β values reflect the choice of the negotiating party in determining the conceding pattern. The values of more important issues exhibit a slow increase or decrease in values and vice versa. However, due to the nature of the curve, each party behaves more conceding towards the end and accepts an offer if equation (4) is satisfied and the values of all the issues are within the acceptable limits.

Table 2: Negotiation Process

	Consumer's offers			Provider's offers		
Time	**Price**	**Users**	**Avail.**	**Price**	**Users**	**Avail.**
2/3	10.085	149	0.978	19.736	102	0.96
50/51	10.315	139	0.978	19.150	125	0.96
100/101	11.03	115	0.977	17.528	172	0.964
136/137	12.13	103	0.973	15.268	193	0.971
138	12.218	103	0.973	OA		

* OA : Offer Accepted

Table 2 shows the negotiation results for an instance with Bronze customers (Table 1) with preference values of 0.5, 0.2, and 0.3, best-worst value pairs of (10-15), (150-100), and (0.979-0.97) for price, users and availability respectively. For both parties, we use the same preference weights, DF of 0.7, and MaxNegTime of 200. At time 138, provider agent accepts the offer. With consumer price limit (8-9.5) the process terminates without any agreement at the maximum time. Since the offers are calculated based on the utility function, the convergence is only time driven. In the example run, the value of the 2nd issue (users) crosses each other around time 61/62 and continues towards lower utility value. Figure 6 shows how the total utility values of the offers merge towards an agreement for the provider agent.

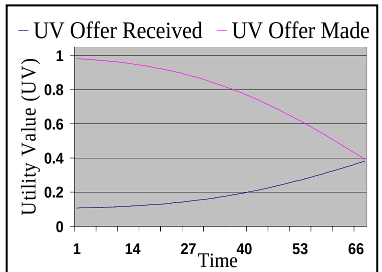

Figure 6: Utility Value

Better negotiation results may be obtained if the other party's offer is taken into consideration while generating a counter-offer and tradeoff is made to achieve better values for preferred issues at the cost of the less important issues. In this preliminary implementation, we have not considered tradeoffs of different issues and mapping of goals to strategy as these require more enhanced negotiation strategy model, which we intend to address in our future research.

5. Conclusion and Future Work

Automated negotiation of SLAs can play an important role in leveraging Web services-based composition of business processes. We present a high level business policy-based Negotiation Broker middleware framework that provides a Web service endpoint for easy invocation of the NB broker service for the negotiation of SLAs for Web services. The main contributions of the paper are the modeling of a high level policy specification, the corresponding WS-Policy specification for negotiation, and the presentation of a middleware broker framework for conducting an automated policy-based negotiation.

We illustrated the progress of the automated bargain between the service provider and the service consumer agents using a prototype implementation of the NB. The negotiation fails if no acceptable offer exists; otherwise both parties show conceding behavior as per the strategy model used in the prototype to reach an agreement. The NB provides a flexible framework for incorporating multiple strategy models and based on the policy specification an appropriate model can be chosen for the current negotiation. Using the negotiation knowledgebase within the NB, learning mechanisms can be implemented to dynamically modify the negotiation tactics to develop efficient negotiation strategy models.

Future research includes implementation of other strategy models, tradeoff mechanisms, and use of the NegKB to apply learning strategies, and illustration of the use of the external resource monitoring feature in the NB framework. The NB can be effectively used for re-negotiation with modified policy specifications. When used within the CSMM [25], it can facilitate service selection based on the best values of the negotiated SLAs. Different trust models have been proposed by researchers that can be used to establish a trust relationship with the NB prior to sending negotiation policy specifications.

The NB provides a flexible framework for policy-based negotiation in a service oriented architecture. The collection of information at a later stage during the negotiation process through the external resource monitor allows informed decisions to be made by the negotiating parties and thus reach a better decision.

6. References

[1] Anderson A.H. (*Sun Microsystems Laboratories*), 2004. An Introduction to the Web Services Policy Language (WSPL). In *Proc. of IEEE Int. Workshop on Policies for Distributed Systems and Networks*, Yorktown Heights, NY.

[2] Beam, C., and Segev, A., 1997. Automated Negotiations: A Survey of the State of the Art. *Wirtschaftsinformatik*, vol. 39(3), pp. 263-268.

[3] Brzostowski, J. and Kowalczyk, R., 2006. Predicting partner's behaviour in agent negotiation. In *Proc. of Int. Joint Conf. on Autonomous Agents and Multiagent Systems, (AAMAS '06),* Hakodate, Japan. ACM, NY, pp. 355-361.

[4] Cappiello, C., Comuzzi, M., and Plebani, P., 2007. On Automated Generation of Web Service Level Agreements. *Proc. of IEEE Int. Conf. on Advanced Information Systems Engineering (CAiSE)*, Trondheim, Norway, pp. 264-278.

[5] Chen, J., Anane, R., Chao, K., Godwin, N., 2002. Architecture of an agent-based negotiation mechanism. In *Proc. of Int. Conf. on Distributed Computing Systems Workshops*, Vienna, Austria, pp. 379-384.

[6] Chhetri, M., Lin, J., Goh, S., Zhang, J., Kowalczyk, R., and Yan, J., 2006. A Coordinated Architecture for the Agent-based Service Level Agreement Negotiation of Web Service Composition. In *Proc. of Australian Software Engineering Conference. (ASWEC'06),* IEEE Washington, DC, pp. 90-99.

[7] Comuzzi, M., and Pernici, B., 2005. An Architecture for Flexible Web Service QoS Negotiation. *Proc. of IEEE Int. EDOC Conf.*, pp. 70-82, Enschede, The Netherlands.

[8] Dan, A., Davis, D., Kearney, R. Keller, A., King, R., Kuebler, D., Ludwig, H., Polan, M., Spreitzer, M., and Youssef, A., 2004. Web services on demand: WSLA-driven automated management, *IBM Systems Journal*, vol. 43(1).

[9] Delaney, M., Foroughi, A., and Perkins, W., 1997. An empirical study of the efficacy of a computerized negotiation support system (NSS). *DSS*, vol. 20(3), pp. 185-197, Elsevier Science.

[10] Faratin, P., Sierra, C., and Jennings, N., 1998. Negotiation Decision Functions for Autonomous Agents. *Int. Journal of Robotics and Autonomous Systems,* vol. 24(3-4), pp.159-182.

[11] Fatima, S. S., Wooldridge, M., and Jennings, N. R. 2005. A Comparative Study of Game Theoretic and Evolutionary Models of Bargaining for Software Agents. *Artificial Intelligence Review.* vol. 23(2), 187-205.

[12] FIPA Contract Net Interaction Protocol Specification. At: http://www.fipa.org/specs/fipa00029/SC00029H.pdf.

[13] Gimpel, H., Ludwig, H., Dan, A., and Kearney, B., 2003. PANDA: Specifying Policies for Automated Negotiations of Service Contracts. In *Proc. of Int. Conf. on Service Oriented Computing (ICSOC'03)*, Trento, Italy. Orlowska, LNCS 2910, pp. 287–302, Springer.

[14] Hou, C., (2004). Predicting agents' tactics in automated negotiation. In *Proc. of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04),* pp. 127- 133, Beijing, China.

[15] Hung, P., Li, H., and Jeng, J., 2004. WS-Negotiation: An Overview of Research Issues. *Proc. of Hawaii Int. Conf. on System Sciences (HICSS).* Washington, DC, vol. 1, IEEE.

[16] Jelassi, T., and Foroughi, A., 1989. Negotiation Support Systems: an Overview of Design Issues and Existing Software. *Decision Support Systems,* vol. 5(2), 167-181.

[17] Kersten, G. E., and Cray, D., 1997. Perspectives on Representation and Analysis of Negotiation: Towards Cognitive Support Systems. Centre for Computer Assisted Management, Carleton University, Ottawa, Canada.

[18] Li, H., Su, S., and Lam, H., 2006. On Automated e-Busines Negotiations: Goal, Policy, Strategy, and Plans of Decision and Action. *Journal of Organizational Computing and Electronic Commerce*, vol. 13(1), pp. 1-29, 2006.

[19] Ludwig, S. A., Kersten, G. E., and Huang, X., 2006. Towards a Behavioral Agent-based Assistant for e-Negotiations. *In Proc. of Montreal Conf. on E-Technologies (MCETECH)*, Montreal.

[20] Narayanan, V. & Jennings, N., 2006. Learning to negotiate optimally in non-stationary environments. *Int. Workshop on Cooperative Information Agents,* ACM, Edinburgh UK.

[21] Su, S., Huang, C., Hammer, J., Huang, Y., Li, H., LiuWang, Liu, Y., Pluempitiwiriyawej, C., Lee, M., and Lam, H., 2001. An Internet-based Negotiation Server for E-Commerce, *VLDB Journal*, Vol. 10(1), pp. 72-90.

[22] Yee, G., and Korba, L., 2003. Bilateral E-Services Negotiation under Uncertainty. In *Proc. of the Intl. Symposium on Applications and the Internet (SAINT '03)*, Orlando, Florida, and National Research Council Canada.

[23] W3C WS-Policy, 2006. Web Services Policy Framework. v1.2. At: http://www.w3.org/Submission/WS-Policy/.

[24] W3C PLING, 2007. Policy Language Interest Group. At: http://www.w3.org/Policy/pling/.

[25] Zulkernine, F., and Martin, P., 2007. Conceptual Framework for a Comprehensive Service Management Middleware. *IEEE Int. Workshop on Service Oriented Architectures in Converging Networked Environments (SOCNE, AINA'07*), pp. 995-1000, Niagara Falls, Canada.