Web-service semantic enabled implementation of Machine vs. Machine business negotiation

Laurentiu Vasiliu, Michal Zaremba, Matthew Moran, Christoph Bussler

DERI Technical Report 2004 -02 -04

February 2004

DERI Ireland
University Road
Galway
IRELAND
www.deri.ie

DERI Austria
Technikerstrasse 13
A-6020 Innsbruck
AUSTRIA
www.deri.at

Abstract

The ultimate business to business (B2B) integration and deployment target is complete business process automation within and between enterprises with no human intervention. From a business point of view, negotiation is the main mechanism that modern enterprises use for achieving their profit-maximising targets. This paper introduces an automated B2B negotiation solution: the implementation of a semantic-enabled machine versus machine business negotiation as a web service. It is argued in this article that the shift from human vs. human business negotiation to machine vs. machine business negotiation is facilitated by using semantic web technology and implemented as a web service. The negotiation process in the present work is designed for multiple machines (minimum of 3 computers), using a negotiation algorithm that emulates human business negotiation behaviour. Conclusions and possible development directions are brought forward.

Keywords: Web services, Semantic web, B2B, negotiation, Bayesian updating beliefs rule, Automated Negotiation of Collaboration Protocol Agreements (ANCPA).

1. Introduction

In our view, the ultimate target of B2B development is complete business processes automation within and between enterprises with no human intervention in business processes. Therefore we consider that the shift from human negotiation to machine negotiation at an industrial scale is the next B2B development step.

Business to business (B2B) concept, functionality and implementation grew and developed due to Internet world wide development and success. Nowadays, Internet technology relies on the initial WWW standards that emerged during late 1980' and early 1990's. Internet predecessors were Electronic data interchange standard (EDI) [7] and SWIFT standard [18] (used mainly of electronic inter-banking payment) used at the very early development stages of information technology.

Internet industrial and non-industrial applications are becoming more and more complex. Enterprise application integration as well as internet commercial applications are reaching new levels of complexity while the current internet technology and capabilities have almost fulfilled their potential.

Currently New IT technologies are emerging as semantic web and web services ones. The actual WWW is mainly a collection of information that does not provide the support in processing this information in using the computer as a computational device. Software programs can be accessed and executed via the web based on idea of web services. The ultimate vision is that a program designed to achieve a result can use web services as support for its computation or processing. Furthermore, this program might be able to mediate any difference between its specific needs and a web service that almost fits.[8]

The motivation of the present work is driven by some factors: first, the IT technology has reached a development point where complex business application can be designed and implemented, secondly the stage of business automation is in an incipient stage, thirdly the new emerging IT technologies (semantic web and web services) show a real potential for new effective applications and fourthly the negotiation process itself has never been automated for industrial applications.

Considering the above mentioned situation, the focus of this paper is the implementation of a semantic enabled web service for machine versus machine negotiation, based on a negotiation algorithm. Here is presented a negotiation algorithm that is based on Bayesian equilibrium negotiation situation. However, several other negotiation algorithms can be adapted in accordance with the negotiation type that is intended to be performed. The negotiation process is designed as a composed web service. The computer cluster (made of 3 machines or more) can be invoked to perform negotiation for other companies that meet similar industrial and business requirements The negotiation algorithm is implemented as a web service. Semantic web technology offers the benefits of syntactic and semantic mediation between different negotiation concepts by using business ontologies repositories while the web service technology offers the advantage of implementing the negotiation algorithm as a web service that can be invoked and use by more clusters of enterprises for negotiating on behalf of their interests. Next we present the main features of our approach:

1. From the software standards implementation there are three types of negotiation: simple matchmaking, negotiation on static subject and negotiation on dynamic subject. Our solution provides the answer to the first type, by using a static set of agreed parameters such as price, quality and time. None of these type of negotiations have been industrially implemented yet by using ANCPA (see section 5).
2. The algorithm is implemented as a web service.
3. The negotiation algorithm structure is independent of the web service functionality, allowing the use of different negotiation algorithms within the same web service.
4. The algorithm uses semantic web technology to mediate business meanings that in many cases are different from one trading partner to another.
5. A negotiation algorithm is implemented using Bayesian updating beliefs rule and a decision algorithm that emulates human negotiation behaviour, for 3 machines (computers) or adaptable to more.

In the next section (2) the basic concepts which have been chosen as the algorithm implementation for modelling negotiation in the perfect Bayesian equilibrium scenario are introduced.

2. Bayesian negotiation: perfect Bayesian equilibrium

In this section we describe briefly the concepts of the perfect Bayesian equilibrium game in dynamic games of incomplete information. This is an example we have chosen to implement for the negotiation web service application.

Games can be of several kinds: static games of complete information, dynamic games of complete information, static games of incomplete information and dynamic games of incomplete information. Therefore there are several negotiation situations that might be encountered in real industrial situations like: Nash equilibrium in static games of complete information, subgame-perfect Nash equilibrium in dynamic games of complete information, and Bayesian Nash equilibrium in static games of incomplete information

Definition: A perfect Bayesian equilibrium consists of strategies and beliefs satisfying Requirements 1 to 4.

Requirement 1: at each information set, the player with the move must have a belief about which node in the information set has been reached by the play of the game. For a non-singleton information set, a belief is a probability distribution over the nodes in the information set; for a singleton information set, the player’s belief puts probability on the single decision node.

Requirement 2: Given their beliefs, the players' strategies must be sequentially rational. That is, at each information set, the action taken by the player with the move (and player subsequent strategy) must be optimal given the player's belief at that information set and the other players' subsequent strategies (where a 'subsequent strategy' is a complete plan of action covering every contingency that might arise after the given information set has been reached).

Requirement 3: At information sets on the equilibrium path, beliefs are determined by Bayes rule and the players' equilibrium strategies.

Requirement 4: At information sets off the equilibrium path, beliefs are determined by Bayes rule and players' equilibrium strategies where possible.

Perfect information: each information is a singleton – each player knows exactly where he is in the game tree. [9]

The Bayes rule that is used for the belief updates is :

$$P(A_i|A) = \frac{P(A_i)P(A|A_i)}{\sum_{j=1}^{N} P(A_j)P(A|A_j)}$$ [15]

Section 3 presents the industrial case scenario where the perfect Bayesian equilibrium models an example of a possible negotiation situation. For a real industrial implementation, first the negotiation type has to be identified and the appropriate model used to design the negotiation algorithm.

3. Industrial use case scenario

For the web service negotiation application, we propose the following industrial scenario: one company (the customer) needs manufacturing components several times a year (e.g. every 2 months) and asks for RFQ (request for quotation). It has two reliable suppliers from which it buys the needed components. The suppliers are competing on price, quantity and delivery time. Negotiation occurs every two months and by automating it, the negotiation time is intended to be highly reduced obtaining similar or better negotiation payoffs for the customer. By automating the negotiation process, three machines (computers), based in each company will perform the negotiation process on behalf of the company they are representing. In the general case, n companies can be involved in the negotiation process.

The application of the proposed web service is a negotiation algorithm that uses Bayesian updating beliefs rule and a decision algorithm that emulates human negotiation behaviour. The negotiation algorithm is deployed to run as a process on each of the machines involved. Before the negotiation starts, the probabilistic weights within the Bayesian model are introduced first by the owners of the negotiation machines. As the negotiation occurs, the probabilistic weights are updated after every negotiation closed, accordingly with the negotiation payoffs.

4. Negotiation algorithm

In this section we are introducing an example of a negotiation algorithm that emulates human negotiation behaviour and uses the Bayes rule to update the players' beliefs, in a proposed perfect Bayesian equilibrium situation. The algorithm is the following:

Phase 1: Each machine (computer) upload from the ERP (warehouse, accounting modules), basic data to start negotiation as needed quantity, price, delivery time and allowable negotiation margins.

Phase 2: Machines learn behaviours about each others.

In Figure 1 is presented the general architecture of the negotiation process. The actors are 3 machines: Machine 1, Machine 2 and Machine 3. Machine 3 is acting on behalf of the customer while Machine 2 and Machine 1 act on behalf of suppliers that are in competition. First, each machine is updated by the human owners with negotiation beliefs in regard with the other machine(s) by their user. By updating the beliefs we mean setting the probabilities weights within the decision tree each machine have coded in about the other machine(s) they negotiate with. By updating them and using Bayes rule,[19], [20], [21], each machine rule out best negotiation strategy to choose. Based on it, they choose and upload stored negotiation pattern (set of variables) to apply on the negotiation game in regard of price, quantity and delivery time. In figure 1, the 'tree' pair structures that are positioned near Machine 1, Machine 2 and Machine 3, are schematic representations of decision trees based on which Bayes rule is applied for each Machine, Such tree representations are the usual representations for game theory modelling. [15]

Phase 3: Negotiation phase

Once they have decided the negotiation strategy to be applied, it follows the negotiation stage. The machine that represents the customer (Machine 3) asks for a RFQ from the machine 2 and machine 1 (the supplier). Machine 1 and machine 2 send back the first offers, based on their belief about machine 3 and also about each other. Machine 3 compare the prices, quantities and times and ask a new RFQ for a better offer to the machine that has a less attractive offer. Based on its own negotiation algorithm this machine may send a better offer or not. If yes, Machine 3 compares again the offer of Machine 2 vs. Machine 1. Next, machine 3 sends again a request for a better RFQ from the machine with the less competitive offer. If Machine 1 or Machine 2 ask Machine 3 about price, quantity or delivery time details from 'competition', Machine 3 may send them or not, according with the particular negotiation strategy is using. This game continues until Machine 1 and Machine 2 send three times consecutively denials to modify their last offers. Then, Machine 3 pick the best[1] offer, closes the negotiation and issue a firm purchase order to the machine that have issued the best final offer. The negotiation parameters follow a convergent pattern (in case of negotiation agreement and closure) as described in figure 4 or a divergent one in case of negotiation failure.

Phase 4.Learning behaviour: updating the beliefs following the negotiation closure. Once the negotiation is closed, all machines update their probabilistic weights in the Bayesian decision tree algorithm, based on the prices exchanged between them, on the information received during the negotiation and on the negotiation result. In this way they learn and adapt for the next negotiation round.

5. Web service implementation

WSDL negotiation definition

In this section are introduced the WSDL files that model message exchanging between the involved machines. There are two such files, one for the machines that are selling and another one for the machine that is buying. The next file define the buyer (Machine 3) message exchange.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions
```

[1] Here the best offer is understood by the triple (price, quantity, delivery time) that is the most advantageous for M3.

```
--/comment/-- here are defined the namespaces for: soap, http, xs, soapenc,
mime, Y --/comment/--
targetNamespace="http://www.negotiations.com/definitions">
	<types>
		<xs:schema/>
	</types>
	<message name="sendQuoteIn">
		<part name="uniqueToken" type="xs:string"/>
		<part name="quantity" element="" type="xs:double"/>
		<part name="deliveryDate" element="" type="xs:date"/>
		<part name="price" element="" type="xs:double"/>
	</message>
	<message name="lastQuoteWasSentIn">
		<part name="uniqueToken" element="" type="xs:string"/>
	</message>
	<message name="poAcknowledgementIn">
		<part name="uniqueToken" element="" type="xs:string"/>
	</message>
	<portType name="NegotiationServiceSOAP">
		<operation name="poAcknowledgement">
			<input message="y:poAcknowledgementIn"/>
		</operation>
		<operation name="lastQuoteWasSent">
			<input message="y:lastQuoteWasSentIn"/>
		</operation>
		<operation name="sendQuote">
			<input message="y:sendQuoteIn"/>
		</operation>
	</portType>
	<binding name="NegotiationServiceSOAP"
type="y:NegotiationServiceSOAP">
		<operation name="poAcknowledgement">
			<input/>
		</operation>
		<operation name="lastQuoteWasSent">
			<input/>
		</operation>
		<operation name="sendQuote">
			<input/>
		</operation>
	</binding>
	<service name="NegotiationService">
		<portname="NegotiationServiceSOAP"
binding="y:NegotiationServiceSOAP">
			<address location="http://www.buyer.com"/>
		</port>
	</service>
</definitions>
```

The WSDL file for the seller (Machine 1 and Machine 2) is:

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions
--/comment/-- here are defined the namespaces for: soap, http, xs, soapenc,
mime, Y --/comment/--
targetNamespace="http://www.negotiations.ie/definitions">
	<types>
		<xs:schema/>
	</types>
	<message name="requestForQuoteIn">
		<part name="productCode" type="xs:string"/>
		<part name="expectedQuantity" type="xs:double"/>
		<part name="latestExpectedDeliveryDate" type="xs:date"/>
	</message>
	<message name="requestForQuoteOut">
		<part name="uniqueToken" type="xs:string"/>
	</message>
	<message name="requestForBetterQuoteIn">
```

```
            <part name="uniqueToken" type="xs:string"/>
            <part name="quantityTooLow" element="" type="xs:boolean"/>
            <part name="maxQuantityExpected" element="" type="xs:double"/>
            <part name="deliveryDateTooDistant" element=""
type="xs:boolean"/>
            <part name="maxExpectedDeliveryDate" element=""
type="xs:date"/>
            <part name="priceTooHigh" element="" type="xs:double"/>
      </message>
      <message name="sendPOIn">
            <part name="uniqueToken" type="xs:string"/>
      </message>
      <message name="negotiationLostIn">
            <part name="uniqueToken" type="xs:string"/>
      </message>
      <portType name="NegotiationServiceSOAP">
            <operation name="negotiationLost">
                  <input message="y:negotiationLostIn"/>
            </operation>
            <operation name="requestForBetterQuote">
                  <input message="y:requestForBetterQuoteIn"/>
            </operation>
            <operation name="getRequestForQuote">
                  <input message="y:requestForQuoteIn"/>
                  <output message="y:requestForQuoteOut"/>
            </operation>
            <operation name="sentPO">
                  <input message="y:sendPOIn"/>
            </operation>
      </portType>
      <binding name="NegotiationServiceSOAP"
type="y:NegotiationServiceSOAP">
            <operation name="negotiationLost">
                  <input/>
            </operation>
            <operation name="requestForBetterQuote">
                  <input/>
            </operation>
            <operation name="getRequestForQuote">
                  <input/>
                  <output/>
            </operation>
            <operation name="sentPO">
                  <input/>
            </operation>
      </binding>
      <service name="NegotiationService">
            <portname="NegotiationServiceSOAP"
binding="y:NegotiationServiceSOAP">
                  <address location="http://www.supplier.com"/>
            </port>
      </service>
</definitions>
```

The negotiation protocol

The design and implementation approach of proposed algorithm has focused around ebXML and its Collaboration Protocol Profile (CPP) [reference] and Automated Negotiation of Collaboration Protocol Agreements (ANCPA) [1],[6] specifications.

ANCPA (see figure 2) allows for negotiation of both long term and short term (even single business exchange) contracts based on CPP. It can minimise the amount of human interactions to maintain and establish new business relationships. Although the whole architecture with clearly separated component has been proposed, one crucial element is missing – the negotiation algorithm. The creators of the specification decided not to include in version 1.0 of the ANCPA specification, the negotiation algorithm that would be running as a private process at each business partner. The intention of authors of this paper has been to incorporate the missing module into the existing architecture.

The central point of an every ebXML-based system is the registry. When such a system is already in operation, the company needs to publish it profile known as Collaboration Protocol Profile (CPP). It must connect to the registry and uploads its CPP there, which can be updated at any time. The CPP describes, in ebXML specification, the profile of the company e.g. the messages exchanged and transport mechanism for them, business processes supported in the company, security issues etc. The CPP becomes a widely accessible source of information about the business the company performs and allows other organisations to search and access this data. Together with CPP the company can publish the Negotiation Description Document (NDD), which describes what is negotiable, the allowed ranges for numeric values e.g. in case of our algorithm the range for price, quantity or delivery time.

Organisation can search through the repository and find the most suitable partner with whom to perform business. The company retrieves CPP and in this way finds the information of the prospectus partner. The profile includes information on, which business processes are supported in the company, the messages to exchange and transport mechanisms or security – all the valuable information to conduct business. Collaborative Program Agreement (CPA), which is the reflection of the CPP of both companies. The CPA template is created by the second party based on information provided in NDD. On the other hand the CPA template may be published by the first party together with CPP. The Negotiation CPA (NCPA) controls the negotiation protocol. The two business partners exchange negotiation messages based on the negotiation protocol (see figure 3).

The agreed CPA must meet the businesses' needs of both organisations to conduct business over the Internet.

ANCPA presents three types of negotiation: simple matchmaking, negotiation on static subject and negotiation on dynamic subject. Our solution provides the answer to the first type, by using a static set of agreed parameters such as price, quality and time. In this situation two parties share the same understanding of negotiable terms (we can say they use the same ontology) and must accept some values in a given ranges. Two other types of negotiations are going to be a focus of our further research. We have to mention that none of these type of negotiations have been industrially implemented yet by using ANCPA.

Next, Figure 4 shows a distributed negotiation architecture for the particular case presented in this paper. In this distributed architecture, the machines that negotiate (M1, M2 and M3) are physically located and maintained within the companies that they are representing. The negotiation algorithms are modified and improved by the owners of the machines, accordingly with their business policies. Here also is introduced a semantic mediator between negotiation concepts that might be different from customer. The mediation is made by the help of a negotiation ontology repository.

The following figure 5 presents a hosted negotiation also for the particular case presented in this paper. M1, M2 and M3 are hosted in a different organisation that offers the negotiation as a web service for other organisations.

Figure 6 is introduced the generic architecture of the web service defined in this work. Mc1, Mc2,…, Mcy are the customers (ERP servers) that invoke M3 to perform negotiations on their behalf. On the other side we have Ms1, Ms2,…,Msz that are suppliers (ERP servers) that invoke M1, M2,…Mn to negotiate on their behalf. Our implementation presented in figure 1 is a particular case of this generic web service architecture.

6. Related work

There is a substantial body of work devoted to the development of negotiation systems. In this section we compare the contents of this paper with related work in terms of functionality and architecture. Beam and Segev [2] provide a state of the art survey identifying two key stumbling blocks to automated negotiation – absence of ontologies and which strategy to employ. Our system employs a well defined Bayesian model for sequential learning similar to that proposed by Zeng and Sycara in [21] and uses ontologies to provide a semantically enriched implementation. Our system is predicated on the application of Game Theory to negotiations. [3] provides experiential feedback on the benefits of such an approach.

[10], [21], [14] and [16] discuss the application of intelligent agents for e-Commerce negotiation. In [9] different types of agent-based negotiation systems are compared. They range from NSS requiring significant human intervention to more automated systems such as the e-Marketplace system , Kasbah[5], where automated negotiations can take place. However the implementation details of these systems are not provided and no systems employing web services are included.

Bazaar proposed in [21] is a sequential decision making model using Bayesian learning. Bayesian learning is the procedure by which the Bayesian equilibrium is reached. Bayesian equilibrium is based on Nash equilibrium [15]. Nash equilibrium is a set of strategies (or actions) in which each firm is doing the best, given the action of its rivals [18]. Here, we are using also the Bayesian equilibrium situation as [21], as an example used for the negotiation model implementation. There are also other negotiation models different to the Bayesian one that can be also implemented according to the reality they intend to model.

In our approach with this model we go further than [21] by providing a real implementation with the help of web services and semantic web technology. [14] describes a model for negotiation using co-ordination among autonomous agents in co-operating domains where agents can always negotiate to increase their mutual benefits. It presents a formal model of negotiation called the PEA model but doesn't describe an implementation. [16] describes a system for automated negotiation where rounds of negotiations take place between an agent for the buyer (and its subagents) and the agents of the sellers. The next step is determined by the seller at the end of each round. This is similar to our approach but there are also several differences. We specify our negotiation strategy and use a combination of web services and ebXML rather than agents as mentioned in [21], [16] and the JSolver library mentioned in [16].

Kim and Segev in [11] describe five components necessary for an effective dynamic eBusiness negotiation framework. They focus on the problem of modelling an open environment of dynamic eBusiness processes and argue that negotiation is at the core of such processes. Different types of negotiation structures are identified but the paper stops short of implementation details. [17] focuses on how Semantic Web technology can provide support systems for enterprises. Ontology-based search systems are identified as a useful negotiation support tools.

In [12] and [13] Kim, Segev et al. propose using web services and the process definition language BPEL4WS as appropriate technologies for implementing automated negotiations. BPEL4WS is identified by [12] as a solution for managing long running processes involving multiple web services. The paper suggests that Web Services as a standards-based technology supporting dynamic binding is suitable to the ad-hoc nature of inter company negotiations. [12] does not focus on the actual negotiation strategy to employ or the details of the implementation. [13] is a substantial paper that describes using web services in a marketplace architecture for automated negotiations. BPEL4WS is proposed as the standard to follow for managing the process itself while a shared ontology would provide common semantic understanding of concepts. Our approach uses ebXML and CPP to provide the business process framework. We use a Bayesian learning model in our negotiation strategy and implement the negotiation module using semantically enriched Web Services. Combining these technologies, the business and conceptual shift from human vs. human negotiation to machine vs. machine negotiation is facilitated. The right technology is in place to allow such shift.

7. Conclusions

Machine versus machine negotiation automation implementation has been presented. The technology based upon such a B2B application has been detailed. Related research has been introduced and analysed. The Bayesian equilibrium situation is only a negotiation situation that might be encountered. For other negotiation situations, negotiation has to be modelled from case to case. Our proposed negotiation implementation as web service covers a singular area of industry from the negotiation point of view, namely the one where negotiations are usually reaching Bayesian equilibrium. Various negotiation models are already available due to the current and past economic research performed. The technology to implement them in one web service is available.

A first industrial implementation could concern implementation of several different negotiation web services that are addressing different negotiation situations and can be invoked based on their specialisation. Future developments may concern a design of a generalised negotiation web service. A generalised negotiation web service would concern detection, upload and use of one of several negotiation models on behalf of their customers, based on their inputs.

The industrial and business implications of using negotiation web services are several. Considering the Extended Enterprises situations with many interacting supply chains, negotiation automation would mean increased efficiency by eliminating many subjective human influences. Another advantage would be time reduction from months or weeks to minutes. In an extended enterprise, such time reduction would generate a time reduction ‘chain reaction’ as companies are waiting one after another in supply chains.

Figures:

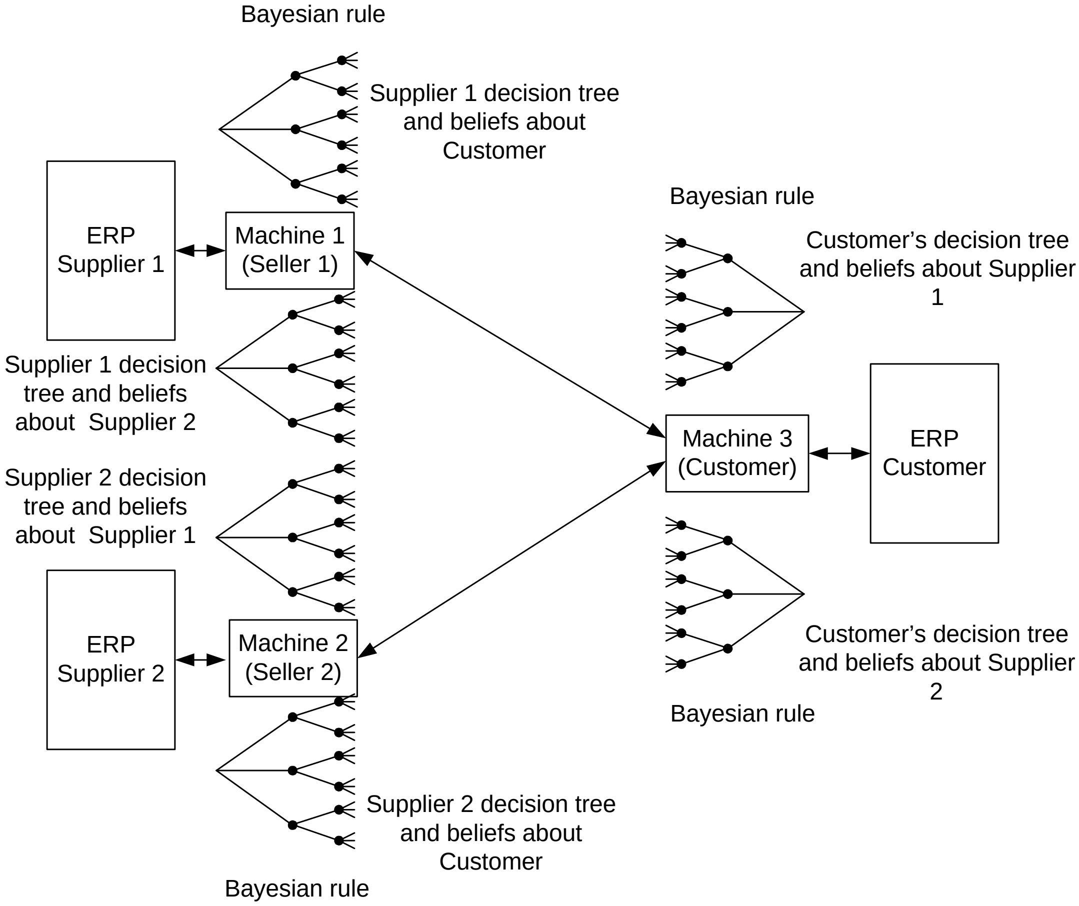

Figure 1 – General machine vs. machine negotiation architecture

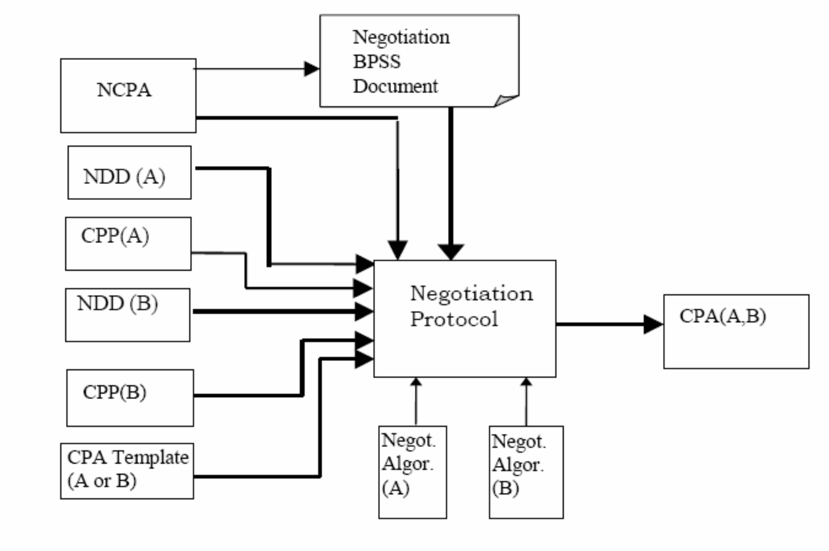

Figure 2: Components of ANCPA negotiation [18]

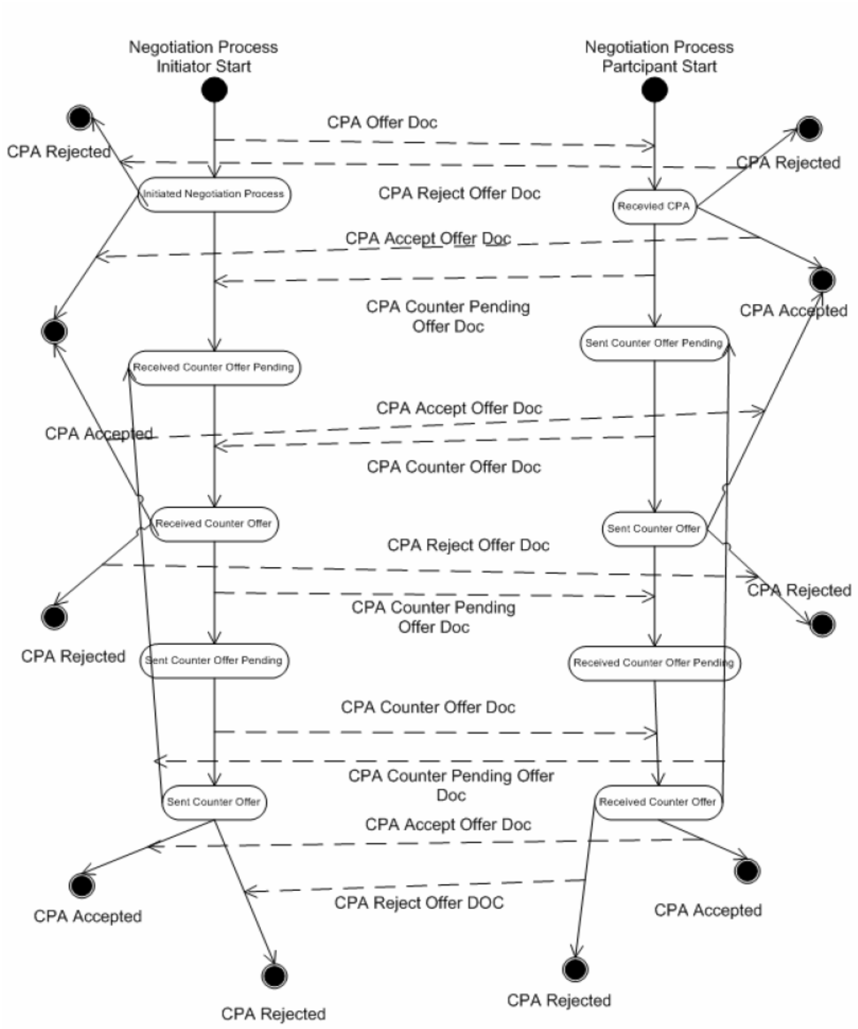

Figure 3: Negotiation protocol – initial offer and counter offers [6]

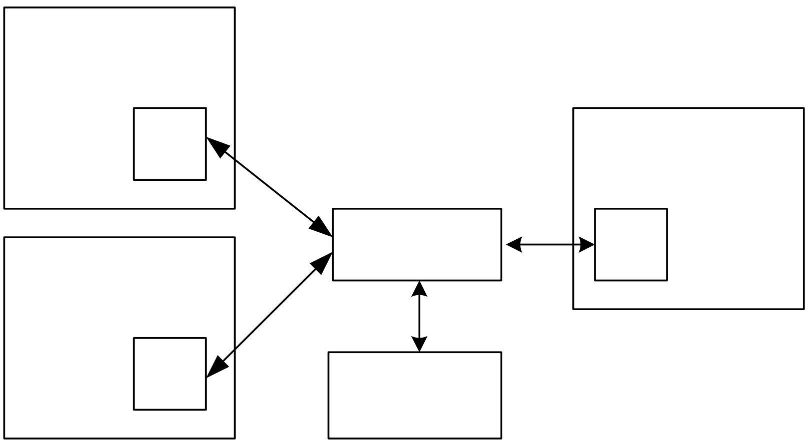

Figure 4 – Distributed web service negotiation architecture

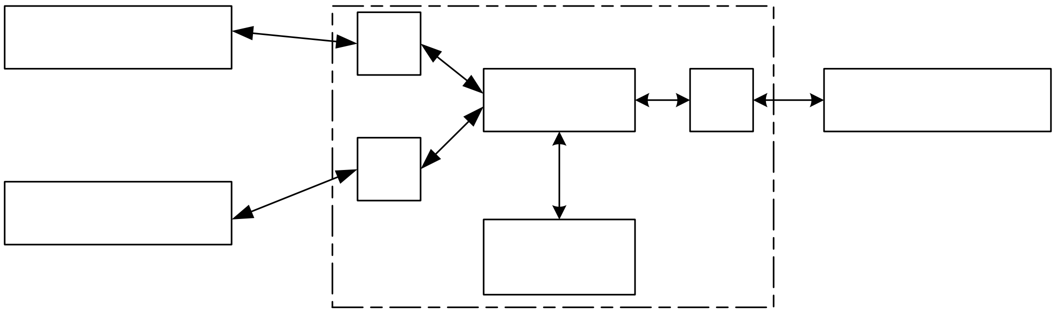

Figure 5 – Hosted web service negotiation architecture

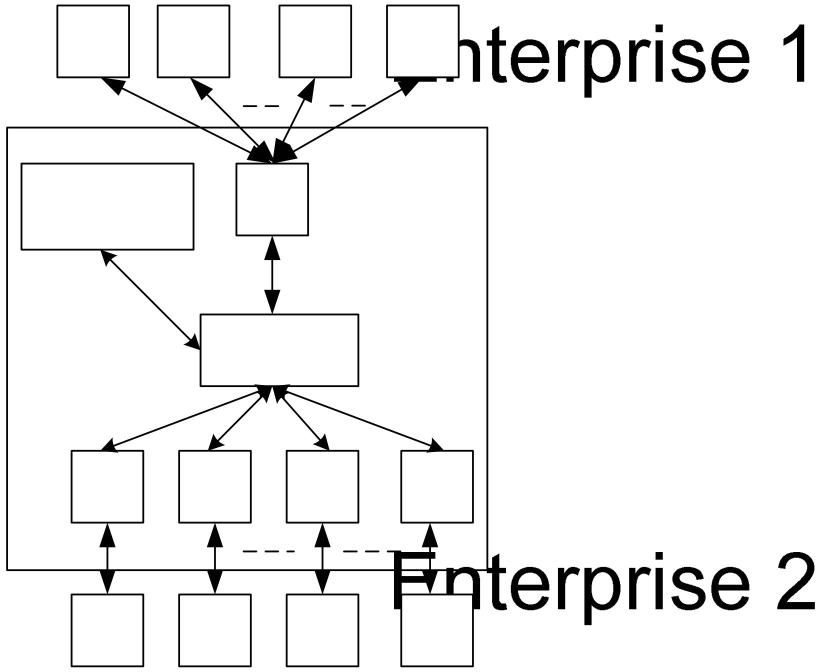

Figure 6. Web-service hosted general architecture

Reference:

[1] Automated Negotiation of Collaboration Protocol Agreements Specification Version 0.10 6; OASIS ebXML Collaboration Protocol Profile 8 and Agreement Technical Committee

[2] Beam C., Segev A., Automated Negotiations: A Survey of the State of the Art, Wirtschaftsinformatik, Vol. 39, No. 3, (1997)

[3] Binmore K., Vulakn N., Applying Game Theory to Automated Negotiation, ESRC Centre on Economics Learning and Social Evolution (1997)

[4] Brickley J.A., Smith C.W. Jr.,Zimmermann J.L. – Managerial Economics and Organisational Architecture, Second Edition – McGraw_Hill (2001)

[5] Chavez A., Maes P., Kasbah: An Agent Marketplace for Buying and Selling Goods, First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (1996)

[6]Collaboration-Protocol Profile and Agreement Specification Version 2.0; OASIS ebXML Collaboration Protocol Profile and Agreement Technical Committee

[7] EDI standard - http://www.unece.org/trade/untdid/welcome.htm ; http://edi.jeita.or.jp/english/about /unei.htm - accessed 30th January, 2004

[8] Fensel, D., Bussler, C. The Web Service Modelling Framework http://informatik.uibk.ac.at/users/c70385/wese/wsmf.paper.pdf
[9] Gibbons, R, A primer in game theory – Prentice-Hall, Pearson Education, 1992
[10] Kersten, G.E., The Science and Engineering of E-Negotiation: An Introduction. IEEE Proceedings of the Hawai'i International Conference on System Sciences 2003
[11] Kim J. B., Segev A., A Framework for Dynamic eBusiness Negotiation Processes; Proceedings of IEEE Conference on eCommerce 2003, p84
[12] Kim J. B. & Segev A., Patankar A., Cho M. G., Web Services and BPEL4WS for Dynamic eBusiness Negotiation Processes, ICWS 2003: Las Vegas, Nevada, USA
[13] Kim J.B., Segev A., A Web Services-Enabled Marketplace Architecture for Negotiation Process Management; Accepted, Decision Supporting Systems Special Issue on Web Services and Process Management
[14] Lyndon C. Lee, Progressive Multi-Agent Negotiation; Proceedings of the First International Conference on Multi--Agent Systems, MIT Press (1995)
[15] Rasmusen, E., Games and Information -An introduction to Game Theory, Blackwell, 1990-1991
[16] Rahwan I., Kowalczyk R. & Pham H.H., Intelligent Agents for Automated One-to-Many eCommerce Negotiation; ACSC2002 Conferences in Research and Practice in Information Technology, Vol. 4
[17] Schoop M., Becks A., Quix C., Burwick T., Engels C., Jarke M., Enhancing Decision and Negotiation Support in Enterprise Networks Through Semantic Web Technologies; SEWASIE Workshop XML Technologies for the Semantic Web, XSW 2002
[18] SWIFT standard – www.swift.org – accessed 30th January, 2004
[19] Vasiliu, L., Browne J., Modelling negotiation in Extended Enterprises - a fractal and game theory approach, International Conference on Industrial Engineering and Production Management, Porto, Portugal, June 2003
[20] Vasiliu, L., Browne J., An integrated modelling approach of the Extended Enterprise using fractals, game theory and neural network approach, 9th International Conference on Concurrent Enterprising on "Enterprise Engineering in the Networked Economy", Espoo-Helsinki, Finland, July 2003
[21] Zeng, D., Sycara, K., Bayesian Learning in Negotiation