# Towards the Automation of E-Negotiation Processes Based on Web Services - A Modeling Approach*

Stefanie Rinderle[1]** and Morad Benyoucef[2]

[1]Dept. DBIS, University of Ulm, Germany, rinderle@informatik.uni-ulm.de
[2] School of Management, University of Ottawa, Canada
benyoucef@management.uottawa.ca

**Abstract.** E-Negotiation is the process of conducting negotiations between business partners using electronic means. The interest in e-negotiation is motivated by its potential to provide business partners with more efficient processes, enabling them to draft better contracts in less time. Most of today's e-marketplaces support some form of e-negotiation. Numerous attempts are being made to design e-marketplaces that support more than one negotiation protocol. The main problem in designing these e-marketplaces is the lack of a systematic approach. In our view, the e-marketplace enforces negotiation protocols and therefore should make them available for consultation by humans and for automation by software agents. Separating the protocols from the e-negotiation media is a step towards a configurable e-marketplace. In this paper we address the requirements for modeling e-negotiation protocols. Then we adopt the Statechart formalism as a modeling language and provide descriptions of five commonly used e-negotiation protocols. Finally, we discuss how we move from these Statechart descriptions of the protocols to modeling the interactions between the e-marketplace participants using a web service orchestration language.

## 1 Introduction

Contracts are the basis for creating business relationships between organizations. A possible sequence of contract operations includes: (1) the establishment phase where the parties negotiate the terms of the contract; and (2) the performance phase where the contract is monitored and enforced [1]. The recent developments of electronic means for communication and collaboration between business partners led to the emergence of electronic contracting (e-contracting) as an alternative to manual contracting. By integrating their IT infrastructures with those of their partners, traditional businesses move a step closer towards becoming real e-businesses. We believe e-contracting to be a cornerstone in that

* This work was conducted as part of a SSHRC funded project on Electronic Negotiations, Media, and Transactions for Socio-Economic Transactions.

** This research work was conducted during a post doctoral stay at the School of Management, University of Ottawa, Canada

integration. Electronic negotiation (e-negotiation) is defined as the process of conducting negotiations between business partners using electronic means. The interest in e-negotiation is motivated by its potential to provide business partners with more efficient processes, enabling them to arrive at better contracts in less time. The research community recognizes three categories of e-negotiation systems [2]: (1) negotiation support systems assist users with communication and decision-making activities; (2) negotiation software agents replace users in their communication and decision-making activities; and (3) e-negotiation media provide a platform that implements a negotiation protocol. There are two categories of e-negotiation media: servers which implement multiple protocols, and applications which implement a single protocol. Traditionally, applications have dominated negotiation design, but lately, the importance of servers has increased, and a need for configurable servers is being felt [3]. Attempts were made to design configurable e-negotiation media to support more than one negotiation protocol. They were partially successful, but they were designed in an ad-hoc manner. Some of these attempts were: the AuctionBot [4] which supports the configuration of various auctions; GNP [5] which separates auction specifications from the logic of the server, and eAuctionHouse [6] which allows for the configuration of auctions with the help of an expert system. Recently, Kersten et al. [7] designed a configurable negotiation server that supports bargaining, based on a process model which organizes negotiation activities into phases; and a set of rules that govern the processing, decision-making, and communication. The main problem in designing e-negotiation media is the lack of a systematic approach. Indeed, to this day, design has been a trial-and-error process. We propose a new model for configurable e-negotiation systems in which "e-negotiation media" is the electronic marketplace (e-marketplace) where human and software participants meet to negotiate deals. We refer to "negotiation software agents" as automated negotiation systems. In our model, automated negotiation systems provide a framework for the existence of software agents. The e-marketplace enforces negotiation protocols, and therefore should make these protocols available for consultation (by humans), and for automation purposes (by automated negotiation systems). Separating the protocols from the e-negotiation media is a first step towards a configurable e-marketplace. Separating negotiation strategies from protocols will also give flexibility to the design of automated negotiation systems. The design of e-marketplaces will have a direct effect on the design of automated negotiation systems. Fig. 1 clarifies this model.

(1) Negotiation protocols are designed, formally specified, and made available to the e-marketplace. (2) Negotiation strategies are designed, formally specified, and made available to the automated negotiation system. (3) The e-marketplace configures the negotiation based on the protocol. (4) The automated negotiation system obtains the protocol from the e-marketplace and uses it (5) along with the negotiation strategies to configure the software agent. (6) Automated negotiation takes place. In this paper we only detail the e-marketplace part of the framework. The other part will be elaborated in future publications.

The first objective of this paper is to investigate and assess various formalisms for specifying negotiation protocols, suggest a set of requirements for a formalism

that enables configurable e-marketplaces, and select and apply a formalism that satisfies the requirements.

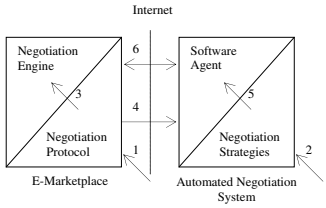


**Fig. 1.** Model for Configurable e-Marketplaces and Automated Negotiation Systems

Businesses are moving towards exposing their services on the web, hoping to interact more efficiently with their partners and to achieve high levels of automation at lower cost. The second objective of this paper is to propose a service oriented architecture (SOA) for our model. According to Kim and Segev [8] web services are the most appropriate way to deploy e-negotiation systems for the following reasons: (1) relationships between negotiating partners are dynamic therefore run-time binding is preferable to design-time binding; (2) negotiation is part of procurement, therefore interoperability with internal and external IT systems is important; and (3) web services provide a standardized and flexible integration technology that no organization can afford to ignore if it wants to interact with its partners. Web services provide the means for software components to communicate with each other on the web using XML. A web service describes itself (using WSDL), can be located (using UDDI), and invoked (using SOAP). A SOA will permit, for instance, an online auction to be deployed on the e-marketplace, and located and invoked through the web by a distant Automated Negotiation System. A web services orchestration language will be used to describe the negotiation process on the e-marketplace. The paper is organized as follows: In Section 2 we propose a set of requirements for a formal specification of e-negotiation protocols and assess different formalisms based on these requirements. Section 3 presents Statechart models for five commonly used e-negotiation protocols. In Section 4 we provide an approach towards the implementation of e-negotiation processes within a SOA. In Section 5 we discuss related work and close with a summary and an outlook in Section 6.

## 2 Requirements for Modeling E-Negotiation Protocols

In this section we state and discuss a set of requirements for describing[1] e-negotiation protocols using common business process modeling formalisms. We proceed in two steps by first summarizing general requirements for business process modeling and then by discussing special requirements in the context of modeling e-negotiation protocols. Based on this, different formalisms are assessed. Finally we select the formalism that best meets our requirements.

[1] Throughout the paper we interchangeably use the terms formal description, modeling, and representation for the same purpose.

### 2.1 General Requirements for Business Process Modeling

There are different formalisms for modeling business processes, e.g., Petri Nets or Statecharts. General requirements for comparing these formalisms are:
**Expressiveness/Completeness:** A first important requirement is the expressive power of the formalism; i.e., which constructs (e.g., sequences, parallel branchings, loops) are supported by the respective formalism. Intensive research has been spent on this question within the workflow patterns project (www.workflowpatterns.com). Important workflow patterns have been identified and commercial workflow systems as well as standard formalisms have been compared to each other regarding their support of these patterns [9]. Like other formalisms (e.g., Petri Nets), Activity Diagrams and Statecharts support the majority of the standard workflow patterns and even some which are typically not supported by commercial Workflow Management Systems (WfMS) [10].
**Formalization/Verification:** It is very important to precisely define the syntax and semantics of a formalism for business process modeling in order to be able to detect modeling errors or inconsistencies (e.g., deadlock causing cycles) at design-time. Petri Nets, for example, provide a sound mathematical foundation such that their dynamic behavior can be examined at design-time. Statecharts as defined in [11] also have a formal specification and a precise operational semantics which enables the use of standard verification methods [12].
**Automation:** In many projects the modeling of business processes comes prior to their automation within, for example, a WfMS. In this case the choice of a business process modeling formalism may also be dependent on whether the processes can be directly executed (e.g., using Petri Nets) or whether or not there exists a mapping to an executable formalism. In the context of process execution within a SOA (which is a major goal within the e-negotiation application domain [8]), the orchestration of web services has become very important. Therefore it is beneficial to use a formalism for which a mapping to a web service orchestration language, for instance the Business Process Execution Language for Web Services (BPEL4WS), can be found. Examples of such model-driven approaches include mappings from Statecharts or Activity charts to BPEL4WS [13, 14].

### 2.2 Specific Requirements for Modeling E-Negotiation Protocols

In addition to these general requirements there are also requirements which are especially important in the context of describing e-negotiation protocols.
**Design for Reactive Systems:** When modeling e-negotiation protocols it is often required to express situations in which the system is waiting for a message (e.g., making a new offer). From this two important requirements can be derived: (1) the modeling formalism should allow to model wait states [10] as well as (2) the sending and receiving of messages. In contrast to many other modeling formalisms Statecharts fulfill both requirements [10]. Statecharts are transition systems where arcs are labeled by Event-Conditions-Action rules. If the Statechart is in a given state then it waits (see (1) above) until a certain event under

a certain condition triggers an action and the system transits to another state. Such events may be sending or receiving messages (see (2) above).
**Understandability/Compactness:** In order to increase user acceptance, business process models should be easy to understand. In particular, models of e-negotiation protocols should be compact and clear. [12] argues that, for instance, Statecharts are perceived by users as being more intuitive and easier to learn than alternative business process modeling formalisms such as Petri Nets. In contrast with activity-oriented formalisms, Statecharts usually lead to much more compact process models as we know from comparative studies. In summary Statecharts meet all the requirements discussed in this section. In particular the compact and understandable representation of the e-negotiation protocols has convinced us to select this formalism for our approach.

## 3 Statechart Models for Five Commonly-Used E-Negotiation Protocols

In this section we provide Statechart models for five commonly-used e-negotiation protocols. We will not discuss the details of the Statechart models since they are self-explanatory. Understandability (see Section 2.2.2) is one of the man requirements of the description formalism. Since the language is complete these models can be modified as needed. Generic templates can also be provided to be used as building blocks for new negotiation protocols.
**Fixed Price:** The Statechart depicted in Fig. 2a) describes the fixed price sale. This protocol (also called the "take-it-or-leave-it" protocol) is a special case of e-negotiations where there is no exchange of offers and counter-offers. A unique offer is created by an "offer to sell" message from the seller which can either be accepted by a buyer or can be withdrawn by the seller to close the negotiation.
**English Auction:** Fig. 2b) shows the Statechart description of the English auction protocol. Each buyer receives an update message containing the bid submitted by a rival buyer and can respond to it with a counter-bid. The auction is closed after a certain time. We modeled this by using the hierarchical state *Auction closed* which contains the final states *Deal* and *No Deal*[2]. There is a deal if there is at least one bid and the last bid exceeds the reserve price [15].
**Dutch Auction:** Dutch auctions are often used to sell perishable goods such as vegetables or airplane seats where the seller starts with a high price and gradually decreases this price [16]. The Statechart depicted in Fig. 2c) reflects a Dutch auction for an arbitrary number of items, i.e., buyers can specify how many items they will purchase at the current price.
**Bargaining:** The bargaining protocol is a two-party negotiation model since both the seller and the buyer can make offers. As can be seen from Fig. 2d) the initial offer is made by the seller. Note that there may be other variations where, for example, initial offers from the buyer are also possible. In order to keep the number of states low we parameterized the offer messages such that we can distinguish between a regular counter-offer and a final offer.

[2] In our models final states are recognized by the absence if outgoing edges.

**Double Auction:** Within a double auction (cf. Fig. 2e) buyers and sellers are bidding at the same time. A match between a seller's and buyer's bid implies a deal. We represented the matchmaking within a clearing phase which is again modeled by using the hierarchical state Auction closed.

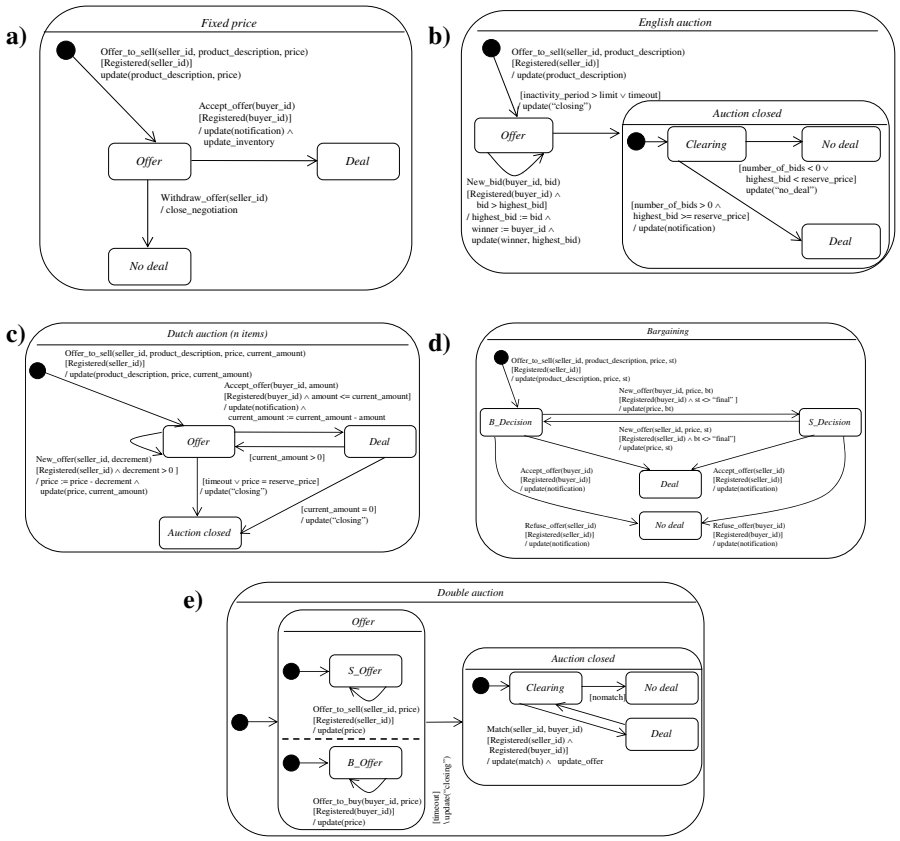


**Fig. 2.** Statechart Models for Commonly-Used E-Negotiation Protocols

## 4 Implementation within a Service-Oriented Architecture

In this section we introduce our approach towards the automation of e-negotiation processes within a SOA.

### 4.1 Why Use a Service-Oriented Architecture?

As pointed out in [8] it is crucial to provide automated e-negotiation systems in order to build flexible inter-organizational supply chain systems. Traditionally,

e-negotiation processes have been carried out by humans registering themselves at certain web pages, placing bids and making offers through fill-in forms, and receiving counter-offers of other participants by updating the respective web pages. One major drawback of this human-centered way of e-negotiation is that the underlying processes are not explicitly modeled but only kept within the human minds. The next step towards automated e-negotiation systems was achieved through software agents which acted as participants in the e-marketplace. In this approach the e-negotiation processes are still not described in an explicit manner. Though they are no longer kept in the human minds they are now hard-coded within the implementation of the software agents. This, in turn, raises important problems regarding the flexibility of the systems in question. Reason is that each process change taking place results in modifying the software agent code. Therefore, on the one hand, an adequate solution for the automation of e-negotiation processes has to be based on the separatio5n of process logic and program code of the invoked applications as realized, for example, in WfMS. On the other hand, a very important requirement for the automation of e-negotiation processes is the adequate support of interoperability between the partner processes in the e-negotiation application domain. In order to meet this requirement, the dynamic invocation of web services within a SOA is the most appropriate approach [8]. Using a SOA also supports the interoperability between internal and external systems of the particular partners. This is crucial since e-negotiation processes are generally part of larger procurement or sales processes [8]. Generally, within a SOA a service provider registers a web service offering a certain functionality (e.g., placing a bid) with a service broker. If a service requester is searching for a certain service it can find the corresponding web service by asking the service broker. The service broker provides a link to the service provider to which the requester can bind. The communication between the parties involved is based on SOAP. Since web services themselves are stateless web service orchestration languages have been developed in order to compose web services into long-running processes. One example for such web service orchestration languages is BPEL4WS. Going back to our Statechart description of e-negotiation protocols and considering a SOA implementation for such protocols it is clear that we need to model the interactions between participants in a given e-negotiation protocol. A web service orchestration language is the obvious solution in this case. The following section provides the BPEL4WS processes for the English and Dutch auction.

### 4.2 Web Service Orchestration for E-Negotiation Processes

Fig. 3a) depicts the BPEL4WS process model for the English auction. In order to illustrate our approach we use an abstract visualization here. The English auction process is initiated if the e-marketplace receives an "offer-to-sell" message from the seller (activity type receive). Using a switch construct a conditional branch is inserted afterwards in order to check whether the seller is registered or not. If so, an update broadcast message is sent using an invoke activity. The bidding phase is modeled using a while construct. The e-marketplace waits until

it receives a "new bid" message from the buyer. After checking the registration, the e-marketplace assigns the new highest bid and sends an update broadcast message including the current highest bid to all participants. If the while loop is terminated by a timeout or by exceeding the inactivity period an update message is sent to all participants indicating that the auction has been closed. Finally, depending on whether there is a deal or not a respective update message is sent to all. Note that in a concrete implementation we precisely distinguish between the different message types. For the sake of understandability we used an abstract update broadcast message type in Fig. 3a). Finding a mapping between the parameterized event messages used within the Statechart models (cf. Fig. 2) and the messages sent within the BPEL4WS processes is one important challenge of our future work.

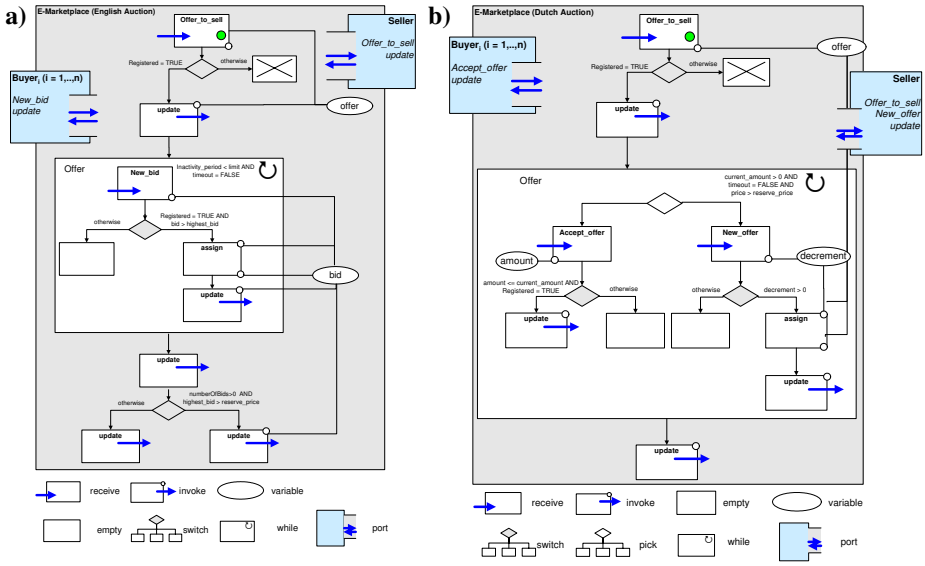


**Fig. 3.** Web Service Orchestrations for English and Dutch Auction

To initiate a Dutch auction the same BPEL4WS pattern (a receive activity waiting for an "offer to sell" from the seller plus a switch construct checking the registration) can be used as in the English auction (cf. Fig. 3b). Finding such patterns is important in order to provide generic e-negotiation protocol templates within the service-oriented e-marketplace. These templates may then be modified and mapped to BPEL4WS processes. The initial phase is followed by a while loop. Here the e-marketplace waits until either a buyer sends an "accept offer" message or the seller sends a "new offer" message. This is modeled using a pick construct. After the auction is closed by terminating the while loop an update broadcast message including all necessary information is sent.

## 5 Discussion

In [15] different price negotiation protocols such as fixed price sale or the Dutch auction are described using finite state machines. The authors aim at discovering common elements, such as basic activities, within auction services in order to provide a design for an auction software. Though finite state machines are a formally founded formalism, Statecharts provide additional constructs (e.g., hierarchical states) which make them better suited for the modeling of e-negotiation protocols. Rolli and Eberhart [17] propose a reference model for describing and running auctions as well as an associated three-layered architecture which is prototypically implemented using a BPEL4WS editor. The processes are then executed within a Java environment. The definition of a reference model and an associated architecture is important. However, it remains unclear how the auction processes are fed into the system. Apparently they are modeled manually using BPEL4WS which might be a complex task for users in general. Therefore our approach of providing generic Statechart models for auction protocols which can then be automatically mapped to web service orchestrations is complementary to the approach proposed by Rolli and Eberhart. Kim and Segev [8] also follow an approach for establishing a web-service enabled e-marketplace. The authors provide a Statechart description for one e-negotiation protocol and the corresponding BPEL4WS process. In this paper we adopt the idea of providing understandable models for e-negotiation protocols and to automate them within a SOA. In [18] e-negotiation protocols are modeled using the Petri Net formalism. Special focus is put on the modeling of attributes which reflect the different strategies the participants in the e-negotiation might adopt. In this paper we focus on the understandability of e-negotiation protocols and therefore we use Statecharts instead of Petri Nets. We also address the question of automating the e-negotiation processes within a SOA afterwards. Chiu et al. [19] present an interesting approach for developing e-negotiation plans within a web services environment. The authors provide meta models for e-contract templates and e-negotiation processes which can be used to set up the concrete e-negotiation processes within a web service environment. Though this approach is generic we believe that providing (generic) e-negotiation templates (i.e., Statechart models) to users which can be individually modified and immediately mapped onto executable web service orchestrations is more intuitive and user-friendly.

## 6 Summary and Outlook

In this paper we introduced Statechart models for five commonly-used e-negotiation protocols. For that we first systematically elaborated the requirements for modeling e-negotiation protocols and select the Statechart formalism as the most appropriate one. We discussed the importance of automating the corresponding e-negotiation processes within a service-oriented environment in order to meet the flexibility and interoperability requirements of the e-negotiation application domain. In order to illustrate our ideas we presented the web service orchestrations for the English and the Dutch auctions which were executed in a web

service environment. In the future we will work on formalizing a mapping between the Statechart models of the e-negotiation protocols and the corresponding web service orchestrations. Based on this mapping it will be possible to provide generic e-negotiation protocol templates to users which are understandable and adaptable. If such a protocol template is chosen and possibly adapted it can be immediately mapped to a BPEL4WS process and then executed in a web-service enabled e-marketplace. This will increase user acceptance and application of e-negotiations and e-marketplaces in practice.

## References


1. Goodchild, A., Herring, C., Milosevich, Z.: Business contracts in B2B. In: Workshop on Infrastructures for Dynamic B2B Service Outsourcing. (2000)
2. Bichler, M., Kersten, G., Strecker, S.: Towards a structured design of electronic negotiations. GDN **12** (2003) 311–335
3. Neumann, D., Benyoucef, M., Bassil, S., Vachon, J.: Applying the MTL taxonomy to state of the art e-negotiation systems. GDN **12** (2003) 287–310
4. Wurman, P., Wellman, M., Walsh, W.: The michigan internet auctionbot. In: Autonomous Agents. (1998) 301–308
5. Benyoucef, M., Keller, R., Lamouroux, S., Robert, J., Trussart, V.: Towards a generic e-negotiation platform. In: Re-Technologies for Inf. Syst. (2000) 95–109
6. University of Washington: The eAuctionHouse (2002)
7. Kersten, G., Law, K., Strecker, S.: A software platform for multi-protocol e-negotiations. Technical report, An InterNeg Research Report 04/04 (2004)
8. J.B. Kim, Segev, A.: A web services-enables marketplace architecture for negotiation process management. Decision Support Systems. **40** (2005) 71–87
9. Aalst, W., Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow patterns. DPD **14** (2003) 5–51
10. Dumas, M., Hofstede, A.: UML activity diagrams as a workflow specification language. In: UML'02. (2001) 76–90
11. Harel, D.: Statecharts: A visual formulation for complex systems. Scientific Computer Programming **8** (1987) 231–274
12. Muth, P., Wodtke, D., Weienfels, J., Kotz-Dittrich, A., Weikum, G.: From centralized workflow specification to distributed workflow execution. JIIS **10** (1998) 159–184
13. Baina, K., Benatallah, B., Casati, F., Tournani, F.: Model-driven web service development. In: CAiSE'04. (2004) 290–306
14. Mantell, K.: From UML to BPEL. Model-driven architecture in a web services world. Technical report, IBM Research (2003)
15. Kumar, M., Feldman, S.: Business negotiations on the internet. Technical report, IBM Research (1998)
16. Kumar, M., Feldman, S.: Internet auctions. Technical report, IBM Research (1998)
17. Rolli, D., Eberhart, A.: An auction reference model for describing and running auctions. In: Wirtschaftsinformatik. (2005)
18. Simon, C., Rebstock, M.: Integration of multi-attributed negotiations within business processes. In: BPM'04. (2004) 148–162
19. Chiu, D., Cheung, S., Hung, P., Chiu, S., Chung, A.: Developing e-negotiation support with a meta-modeling approach in a web services environment. Decision Support Systems **40** (2005) 51–69