Multiple Issue Negotiation Based on Standardized Documents

A Communication Framework

Christoph Pflügler

Inter-organizational Systems Group, University of Augsburg

Abstract. An evolving approach to conduct business transactions over B2B electronic marketplaces is to temporarily connect the ERP systems of the involved business partners. For this type of B2B integration, standardized processes, interfaces and business documents as provided by, e.g., RosettaNet or the NES Consortium constitute an indispensable requirement. However, the respective standards have significant deficits regarding the negotiation of contracts such as purchase orders. The work at hand closes that gap by providing a novel XML communication framework for argumentation-based negotiation which incorporates standardized business documents and allows for interaction between human as well as software agents. Concrete application scenarios are outlined.

1 Electronic Negotiation and Standardization Efforts

There has been a lot of work done in developing middleware for integrating and automating business processes across company boundaries. Recently, XML based technology approaches gained increasing importance in this area. Various consortia and organizations (e.g. RosettaNet[1], NES Consortium[2]) offer XML based business document definitions and process specifications covering many aspects of business within companies of all sizes. However, these standards have significant deficits regarding the negotiation of agreements and are often little more than an electronic catalogue from which a customer can select and order goods. This may be appropriate to many scenarios, but other ones certainly require more sophisticated mechanisms to computationally negotiate, e.g., price, quality and delivery terms of a purchase order (Turowski 2002). In contrary, multi-agent negotiation systems based on communication languages such as FIPA's ACL (FIPA 2003) or KQML

[1] http://www.rosettanet.org/

[2] http://www.nesubl.eu/

(Finin et al. 1994) are fairly complex and, as Beer et al. (1998) state, too academic for many applications. The work at hand strives to close that gap by providing a communication framework for bilateral negotiation which is based on the latest findings in electronic negotiation and argumentation theory, and allows for a seamless integration into existing standards such as RosettaNet's PIPs or the NES profiles through its ability to negotiate on the basis of pre-existing, standardized XML document definitions.

The paper is structured as follows: Section 2 gives a brief overview of the field of electronic negotiation and thereby lays the theoretical foundation for the framework presented. Related work is summarized in section 3. Section 4 elaborates on the requirements for and describes the core components of the proposed communication framework for argumentation-based negotiation: The structure of a so-called negotiation extension for pre-existing business document definitions and a bilateral negotiation process defining required tasks, the control flow of each participant, as well as the resulting message exchanges. An exemplary incorporation of the negotiation extension into an XML document definition is depicted. Further, the framework is evaluated against requirements on negotiation mechanisms. The contribution of the communication framework presented by the work at hand is outlined in section 5, which also summarizes ongoing and future work.

2 Theoretical Background

Negotiation theory distinguishes two types of negotiation (Hung et al. 2004; Walton and McKersie 1965): *competitive negotiation* (also referred to as zero sum or distributive negotiation) and *collaborative negotiation* (integrative negotiation). The first one can be characterized by the fact that every negotiating party exclusively tries to maximize its individual utility. In contrary, collaborative negotiation aims at finding and adopting the solution that provides the greatest joint utility taken the parties collectively. These two types represent two extremes of the spectrum of possible negotiation types; most real-world negotiation scenarios contain competitive as well as collaborative elements (Walton and McKersie 1965). Further, *electronic negotiation* is defined as a joint decision-making process of two or more parties within an electronic market (Smith and Davis 1983). It can be seen as an optional activity within the agreement phase of an electronic market transaction (Rebstock et al. 2003). Electronic negotiation essentially deals with three topics: *negotiation protocols*, *negotiation objects* and the *agent's decision making model*. According to Jennings et al. (2001), a *negotiation protocol* is a set of rules that govern the interaction. This includes the permissible type of participants such as the negotiators and relevant third parties, negotiation states, events that trigger a change of the negotiation state, permissible actions of each participant in a particular negotiation state, as well as resulting message exchanges. Regarding the design of a negotiation protocol, there exist two fundamental alternatives: *offer-counter-offer* protocols (also called send/resend protocols) on the one hand and *mediated* protocols on the other. During a negotiation performed in the offer-counter-offer fashion, one party offers a proposal to the other party, which in turn accepts the proposal or sends a counter-proposal. This procedure is repeated until an agreement is reached or a party aborts the negotiation. In the context of the work at hand, the term mediated protocol does not refer to mediators used to ensure quality of service properties such as non-repudiation or mediators that serve as message queues to store and forward negotiation messages. Mediated protocols as the one proposed by Klein et al. (2003) can be leveraged for fully automated negotiation in a multi-agent environment. They employ a central mediator that initially generates a random contract. Each negotiating party then votes to accept or reject the proposal. If the proposal is accepted by both parties, the mediator mutates that proposal (by randomly flipping a value of an issue) and repeats the procedure. In case one or both parties vote to reject, a mutation of the last mutually accepted proposal is suggested instead. This procedure is repeated for a fixed number of steps.

Negotiations can further be categorized by *negotiation objects* which constitute the range of issues an agreement must be reached for (Jennings et al. 2001). *Single issue* negotiations, such as negotiating a price only and *multiple issue* negotiations, e.g., negotiating price, quality and delivery terms are distinguished.

If automated agents are involved in the negotiation process, their *decision making model* is utilized for further classification (Jennings et al. 2001). On the one hand, *game theoretic models* aim at finding the best solution for a negotiating party. However, they are subject to fundamental limitations, most notably partial knowledge, inconsistent beliefs and bounded rationality (Först et al. 2009). On the other hand, bargaining agents incorporating *heuristic approaches* try to find a good instead of the optimal solution and overcome the limitations of game theoretic approaches by searching the negotiation space in a non-exhaustive manner (Jennings et al. 2001). Moreover, *argumentation-based approaches* have been proposed as an alternative to classical mechanism design (Först et al. 2009). They extend simple proposal-based heuristic approaches by the ability of an agent to justify and persuade the counter-party of its negotiation stance by the use of *arguments*.

3 Related Work

Various approaches exist in the field of agent communication, most notably DARPA's KQML and FIPA's ACL, which cover roughly the same aspects (Beer et al. 1999). The contribution of these approaches is the provision of standard languages and protocols for agent interaction (Finin et al. 1994). Although these standards cover a wide range of applications, practical use of agent systems based on them remained limited in e-commerce. This fact is mainly induced by their complexity which inhibited a widespread adoption in small and medium enterprises (SMEs), or companies in less developed countries. Recently, the Argument

Interchange Format (AIF) was proposed for the representation and communication of arguments between software agents (Chesnevar et al. 2006; Rahwan and Reed 2009). The AIF provides all means to describe *complete argument networks* (i.e. multiple issues); additionally a corresponding extension to handle dialogue has been proposed (Modgil and McGinnis 2008). A language for argument representation particularly useful for supporting a *single issue* is the Argument Markup Language (AML) proposed by Reed and Rowe (2004) in the framework of the Araucaria system.

To communicate XML documents between agents, Kurbel et al. (2000) proposed their integration as payload of KQML messages and built a prototype of a corresponding system architecture. An offer-counter-offer protocol for electronic negotiation is frequently proposed in literature (Ströbel 2002; Rebstock et al. 2003; Kim et al. 2003). A general design and application framework for electronic negotiation based on XML is provided by Ströbel (2002). Regarding the messages exchanged, he proposes to extend a pre-defined negotiation base schema with domain specific ontologies. Rebstock et al. (2003) depict that an offer-counter-offer protocol can be specified in ebXML BP (OASIS Open 2006). It is shown that making an XML document negotiable increases the document complexity. However, they did not provide a generic approach to extend pre-existing document definitions as I do. Further, semi-structured electronic negotiation is proposed for B2B environments (Rebstock 2001). Thereby, structured business documents are accompanied by unstructured notes which clearly limit the degree of automation that can be achieved through software agents. Kim et al. (2003) as well as Rinderle and Benyoucef (2005) concluded that a Web Service / BPEL based implementation is most promising for electronic negotiation in e-commerce.

To the best of my knowledge, I am the first one that incorporates the argumentation-based negotiation approach into XML-based multi-issue negotiation.

4 Argumentation-Based Negotiation of XML Documents

This section starts with the elaboration of the requirements for a negotiation system by literature review. Based on these requirements, I design the core components of my negotiation framework, incorporating latest findings in the field of electronic negotiation. First of all, the structure of a so-called negotiation extension enabling for electronic negotiation based on arbitrary XML document definitions is elaborated. Then, a corresponding bilateral negotiation process which takes advantage of the features of the negotiation extension is developed. The integration of the negotiation extension into pre-existing XML document definitions is exemplified. Lastly, the proposed communication framework is evaluated.

4.1 Requirements for a Negotiation Mechanism

Rosenschein and Zlotkin (1994, pp. 20-22), Sandholm (1999, pp. 202-204), Jennings et al. (2001), Lomuscio et al. (2003) and others define a *negotiation mechanism* as a combination of the negotiation protocol and the agents' decision making model. Thereby, desired properties for the design of such systems are described. The commonly accepted ones are:

- *Pareto Efficiency:* This criterion takes a global perspective and evaluates alternative outcomes a mechanism can lead to. An agreement should fulfill this criterion as utility should not be wasted. An agreement is pareto-efficient, if there is no other possible agreement that improves the utility of one agent without leaving the other agent worse off.
- *Communication and computational efficiency:* A mechanism should not require high computational effort, and should keep communication effort low. All other things being equal, a mechanism that is computationally efficient and handles communication between agents in an efficient way is preferable.
- *Distribution of computation:* A negotiation process should not require a central decision maker as this would create a single point of failure. Besides, truly distributed mechanisms offer advantages in terms of scalability.
- *Symmetry:* A mechanism should not be biased in favor or against some agent. For instance, one agent should not be obliged to carry out many more tasks than another one.

Furthermore, I identified the requirement that a negotiation mechanism should allow for interaction between human agents and software agents. A scenario I explicitly want to support is a large enterprise employing a software negotiation agent, which is supposed to negotiate about, e.g., orders with a supplier that, in turn, lacks the financial resources and/or expertise to leverage agent technology and is thus forced to draw decisions manually during the course of a negotiation process. A similar requirement was found by Modgil and McGinnis (2008) in the context of a medical multi-agent system.

4.2 Communication Framework for Electronic Negotiation

This section describes the core components of the negotiation framework presented by the work at hand. Thereby, corresponding design decisions are discussed on the basis of latest developments in electronic negotiation.

The Negotiation Extension

Först et al. (2009) have shown in an empirical case study that argumentation-based negotiation produces significantly better results regarding social welfare as well as number of communication units and steps than negotiation based on the exchange of simple proposals. Thus, considering the requirements on a negotiation mechanism listed in section 4.1, the argumentation-based approach is favourable. As opposed to multilateral negotiation, bilateral negotiation is important for both theoretical and practical reasons (Lomuscio et al. 2003). Therefore, I set the scope of the work at hand to one-on-one negotiation.

To enable an arbitrary XML document to serve as basis for electronic negotiation, I extend every relevant XML component (i.e. *elements* and *attributes*) with a so-called *component negotiation extension*. Additionally, a corresponding *document negotiation header* which needs to be integrated into the document's root element is required. The structure of this negotiation header is depicted in figure 1. It contains a *Version* attribute which needs to be incremented whenever a proposal is modified and sent to the corresponding counter-party. Thus, the course of a negotiation process is traceable. To guarantee the termination of a negotiation process, the attribute *isLastOffer* is introduced. A proposal with this attribute set to *true* indicates the last negotiation iteration as described in section "The Negotiation Process". The attribute *GlobalState* shows the overall state of the document being negotiated and can be derived by aggregating the negotiation states of the single components. For the framework proposed by the work at hand, the states *Rejected*, *Pending*,

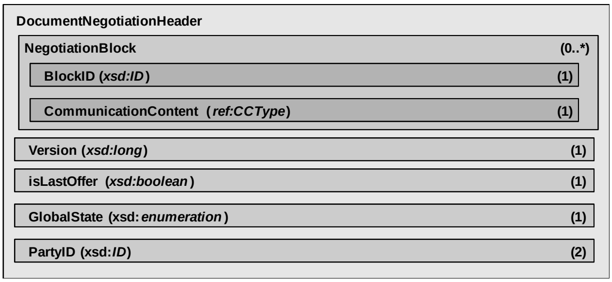

Figure 1: The structure of the document negotiation header

Initiator-Accepted, *Responder-Accepted* and *Accepted* (by both sides) are required. Every party participating in a negotiation needs to be declared using the *PartyID* fields. As in real-world negotiation scenarios complete bundles of issues are negotiated as a whole, the negotiation header also allows for the specification of so-called *negotiation blocks*, which enable the aggregation of several issues (i.e. document components such as price, quality and quantity of an order item) to a single negotiation issue.

Every component of an XML document is extended with a c*omponent negotiation extension*. A component negotiation extension contains either a reference to a negotiation block defined in the negotiation header (if it should be negotiated jointly with other components) or a *CommunicationContent* element of type c*ommunication content type (CCType)*. The structure of the component negotiation extension is shown in figure 2.

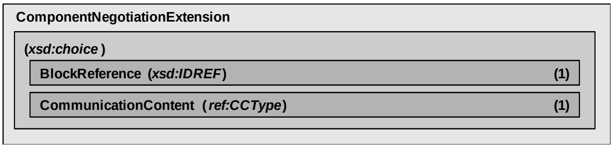

Figure 2: The structure of the component negotiation extension

The *CCType* (see figure 3) referenced in both negotiation block and component negotiation extension encapsulates the actual content of an agent conversation. Analogously to GlobalState, the *IssueState* attribute indicates the actual state of the negotiation of a single issue. Possible values are *Pending*, *Accepted* and *NotNegotiable*. The latter one must be incorporated to tag issues not part of the negotiation (such as a supplier's address). The *communication block* allows a negotiating party to communicate *speech acts* and corresponding *arguments*. Wooldrigde and Parsons (2000) introduced a multi-party negotiation language for e-commerce, which includes the speech acts *request*, *offer*, *reject*, and *accept*. Van Veenen and Prakken (2006) slightly adapted it to bilateral negotiations, my framework adopts that app-roach. The *Arguments* element contains the arguments belonging to a speech act. As in my approach arguments always refer to a single issue (i.e. to the value of a single component or to a negotiation block), I found the AML to be most appropriate for

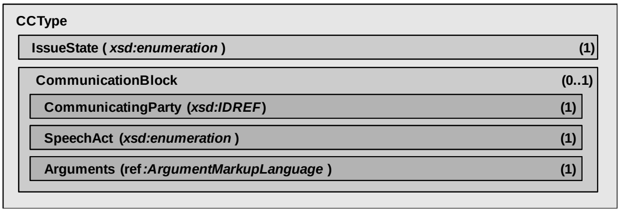

Figure 3: The structure of the communication content type

argument representation. Note that the AML is just a format to represent given arguments supporting a claim, thus a domain-specific set of arguments with fixed semantics needs to be defined if automated agents are involved in a negotiation process.

The Negotiation Process

As described in section 2, there exist two basic types of negotiation protocols: mediated and offer-counter-offer protocols. I found that only an offer-counter-offer protocol fulfils the requirements listed in section 3.1. On the one hand, the mediated protocol has only been tested for boolean issues (Klein et al. 2003) which is not adequate to negotiation in electronic market transactions. On the other

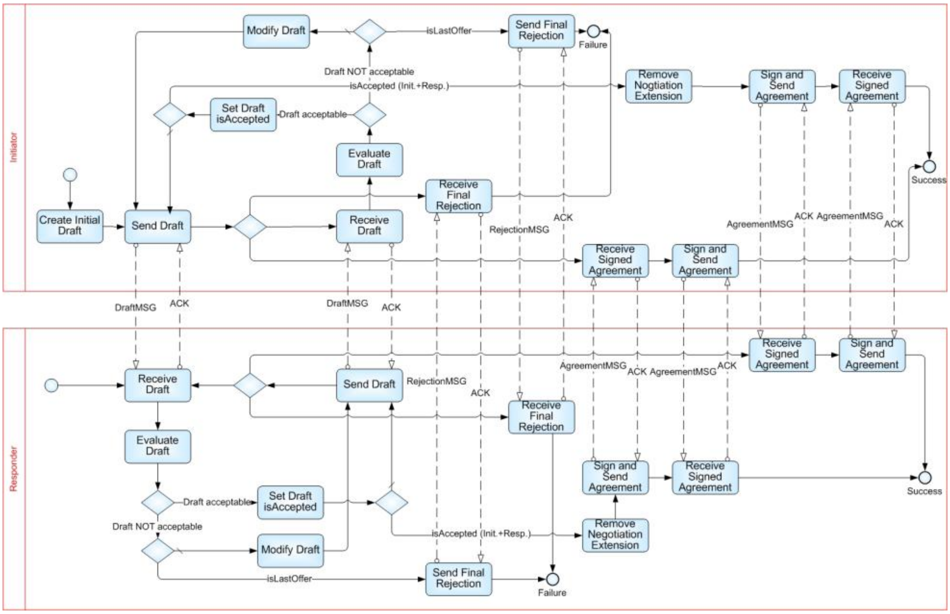

Figure 4: Proposal-counter-proposal negotiation process modelled in BPMN

hand, the multitude of random proposals generated throughout a mediated negotiation process inhibits an adequate incorporation of human agents (hundreds or even thousands of random proposals would have to be rated to reach an agreement). Furthermore, a central mediator would create a single point of failure in the negotiation system. An offer-counter-offer negotiation process defining the control flow, required tasks, as well as corresponding message exchanges is shown in figure 4. The *Initiator* starts the process by creating a first draft of the document that should be negotiated, including the previously described negotiation extension. Then, the draft is sent to the *Responder* of the negotiation process. Note that initially defined negotiation blocks can be altered throughout the course of a negotiation process. The Responder, in turn, receives and evaluates the draft. If the attribute *isLastOffer* defined in the document negotiation header is set to *true* and the overall draft is not acceptable, a final rejection message (the latest draft with *GlobalState Rejected*) is sent to the Initiator and the process terminates with both parties moving to state *Failure*. Otherwise, if the actual draft is not the last offer, but still not acceptable, the Responder creates a counter-proposal including corresponding speech acts and arguments for each issue, and sends this counter-proposal back to the Initiator, thus triggering the next iteration of the negotiation process. If the Responder receives an acceptable draft (whereas a draft is acceptable, if an agreement is reached for all issues), it sets the *GlobalState* attribute to *Responder-Accepted* and transmits the document to the Initiator. After evaluating the draft already accepted by the Responder, the Initiator decides whether the draft is acceptable for him or not. If it is acceptable, it changes the *GlobalState* attribute to *Accepted* and removes the negotiation extension. To ensure *non-repudiation* of the agreement, the negotiating parties then sign the agreement using a predefined signature algorithm. Finally, the signed agreements are exchanged. Note that the negotiation process is symmetric for the most part, and therefore a complete description is omitted.

Further, a micro-protocol concerning the usage of speech acts in the course of the negotiation of a single issue has to be considered (McBurney and Parsons 2003).

Table 1: Speech Acts and Possible Replies

Speech Act	*Possible Replies*
request(φ)	*offer(φ')*
offer(φ)	*offer(φ')* or *accept(φ)* or *reject(φ)* or *withdraw*
reject(φ)	*offer(φ')* or *withdraw*
accept(φ)	
withdraw	
φ ≠ φ'	

The negotiation language incorporated restricts the usage of speech acts as shown in table 1 (van Veenen and Prakken 2006). A proposal for an issue including corresponding arguments is denoted with φ. If a proposal for an issue is accepted, the *IssueState* attribute has to be set to *Accepted*. Note that in case of modifications of a negotiation block (i.e. adding or removing issues) the micro-protocol of the affected issues has to be reset.

Extending XML Documents

My approach is to negotiate on the basis of standardized XML documents as provided by, e.g., OASIS Open's UBL or RosettaNet's PIPs. For this purpose, the previously introduced negotiation extension needs to be integrated into the respective XML document definitions. This section briefly demonstrates by the example of an excerpt of a simple XML schema definition (presented in listing 1), how the proposed negotiation extension can be integrated.

The schema consists of two elements, PurchaseOrder, which is assumed to be the root element of a document instance, and LineItem, assumed to be a child element of PurchaseOrder. To enable the negotiation of a document instance based on this definition, first of all the XML schema defining the negotiation extension must be

```
<element name="PurchaseOrder">
  <complexType>
      <sequence>
          <element ref="tns:LineItem" />
      </sequence>
  </complexType>
</element>

<element name="LineItem">
  <complexType>
      <sequence>
          <element name="price" type="double" />
      </sequence>
      <attribute name="currency" type="string" />
  </complexType>
</element>
```

Listing 1: Excerpt of a very simple XML Schema document definition

imported. Then, the document negotiation header needs to be added as an additional element to the document root element (not depicted for space reasons). Further, every XML element or attribute is extended with the component negotiation extension as depicted for element *LineItem* in listing 2. The structure of the extension also allows for an algorithmic removal which is useful for the automation of the task “Remove Negotiation Extension” at the end of the negotiation process.

```
<element name="LineItem">
  <complexType>
    <sequence>
      <element name="price">
        <complexType>
          <sequence>
            <element name="content" type="double" />
            <element
                ref="neg:ComponentNegotiationExtension" />
          </sequence>
        </complexType>
      </element>
      <element name="attributeWrapper">
        <complexType>
          <sequence>
            <element
                ref="neg:ComponentNegotiationExtension" />
            </sequence>
            <attribute name="currency" type="string" />
        </complexType>
      </element>
    </sequence>
  </complexType>
</element>
```

Listing 2: XML Schema elements and attributes extended with the negotiation extension

4.3 Evaluating the Approach

In this section, the presented XML communication framework for argumentation-based multi-issue negotiation based on standardized business documents is evaluated against the requirements introduced in section 4.1.

As Först et al. (2009) have shown, argumentation-based negotiation produces better results regarding social welfare and communication effort than the exchange of simple proposals. Thus, my approach provides the technical prerequisites for reaching a pareto-efficient outcome while reducing the number of communication units and steps. However, for software agents the incorporation of arguments is likely to increase computational complexity as they need to process and generate complex arguments. My communication framework does not require a central decision maker and thus satisfies the requirement of distributed computation. The requirement of symmetry is also fulfilled as neither one of the participants of the negotiation process has to carry out many more tasks than the other one. Lastly, the offer-counter-offer protocol allows for the incorporation of human as well as software agents.

5 Conclusion and Future Work

The XML communication framework presented by the work at hand is a novel approach that incorporates argumentation-based negotiation and pre-existing document definitions. Its structure facilitates its integration into existing, standardized process environments as provided by RosettaNet's PIPs or the NES profiles. For example, a *PIP Negotiate Purchase Order* (based on my framework) could follow *PIP 3A1 Request Quote*. A potential customer could create an initial draft of a negotiable order based on the respective standardized business document. After an arbitrary number of negotiation iterations, either a purchase order is reached or the transaction is aborted. Further, the framework could be applied to the negotiation of collaboration partner agreements, e.g., ebXML CPAs (OASIS Open 2002), or to collaborative process modeling in a P2P environment.

In the future extension of this work, I plan to develop well-defined sets of arguments for key application domains in order to facilitate the development of software agents that incorporate the argumentation-based approach. The design of respective agents that can cope with the complexity of real-world e-commerce scenarios constitutes another crucial aspect that needs further attention.

References

Beer M, D'inverno M, Luck M, Jennings N, Preist C, Schroeder M (1999) Negotiation in multi-agent systems. The knowledge engineering review 14(3):285–289.

Chesnevar C, Modgil S, Rahwan I, Reed C, Simari G, South M, Vreeswijk G, Willmott S (2006) Towards an argument interchange format. The knowledge engineering review 21(4):293–316.

Finin T, Fritzson R, McKay D, McEntire R (1994) KQML as an agent communication language. In: Proceedings of the third international conference on information and knowledge management. Gaithersburg.

FIPA (2003) FIPA ACL message structure specification. http://www.fipa.org/specs/fipa00061/SC00061G.html (Accessed July 31, 2009).

Först A, Rettinger A, Nickles M (2009) Argumentation- vs. proposal-based negotiation: an empirical case study on the basis of game-theoretic solution concepts. In: Argumentation in multi-agent systems: fifth international workshop. Revised selected and invited papers. Estoril.

Hung PCK, Li H, Jeng JJ (2004) WS-Negotiation: an overview of research issues. In: Proceedings of the 37th Hawaii international conference on system sciences. Koloa.

Jennings NR, Faratin P, Lomuscio AR, Parsons S, Wooldridge MJ, Sierra C (2001) Automated negotiation: prospects, methods and challenges. Group decision and negotiation 10(2):199–215.

Kim JB, Segev A, Patankar A, Cho MG (2003) Web services and BPEL4WS for dynamic eBusiness negotiation processes. In: Proceedings of the first international conference on web services. Las Vegas.

Klein M, Faratin P, Sayama H, Bar-Yam Y (2003) Negotiating complex contracts. Group decision and negotiation 12(2):111–125.

Kurbel K, Teuteberg F, Szulim D (2000) XML-based agent communication for electronic marketplaces. In: Proceedings of ISA 2000 - intelligent systems and applications. Wollongong.

Lomuscio AR, Wooldridge M, Jennings NR (2003) A classification scheme for negotiation in electronic commerce. Group decision and negotiation 12(1):31–56.

McBurney P, Parsons S (2003) Dialogue game protocols. Lecture notes in computer science 2650:269–283.

Modgil S, McGinnis J (2008) Towards characterising argumentation based dialogue in the argument interchange format. Lecture notes in computer science 4946:80-93.

OASIS Open (2006) ebXML Business Process Specification Schema Technical Specification v2.0.4. http://docs.oasis-open.org/ebxml-bp/2.0.4/OS/spec/ebxmlbp-v2.0.4-Spec-os-en-html/ebxmlbp-v2.0.4-Spec-os-en.htm.

OASIS Open (2002) ebXML Collaboration-Protocol Profile and Agreement Specification Version 2.0. http://www.oasis-open.org/committees/ebxml-cppa/documents/ebcpp-2.0.pdf.

Rahwan I, Reed C (2009) The argument interchange format. In: Argumentation in artificial intelligence. Springer, Heidelberg.

Rebstock M (2001) An application architecture for supporting interactive bilateral electronic negotiations. Lecture notes in computer science 2115:196–205.

Rebstock M, Thun P, Tafreschi AO (2003) Supporting interactive multi-attribute electronic negotiations with ebXML. Group decision and negotiation 12(4):269-286.

Rinderle S, Benyoucef M (2005) Towards the automation of e-negotiation processes based on web services-a modeling approach. Lecture notes in computer science 3806:443-453.

Rosenschein JS, Zlotkin G (1994) Rules of encounter: designing conventions for automated negotiation among computers. The MIT Press, Cambridge.

Sandholm TW (1999) Distributed rational decision making. In: Multiagent systems: a modern approach to distributed artificial intelligence. The MIT Press, Cambridge.

Smith RG, Davis R (1983) Negotiation as a metaphor for distributed problem solving. Artificial Intelligence 20:63–109.

Ströbel M (2002) An XML schema representation for the communication design of electronic negotiations. Computer Networks 39(5):661–680.

Turowski K (2002) Agent-based e-commerce in case of mass customization. International Journal of Production Economics 75(1-2):69–81.

van Veenen J, Prakken H (2006) A protocol for arguing about rejections in negotiation. Lecture notes in computer science 4049:138-153.

Walton RE, McKersie RB (1965) A behavioral theory of labor relations. McGraw-Hill, New York.

Wooldridge M, Parsons S (2000) Languages for negotiation. In: Proceedings of the fourteenth European conference on artificial intelligence. Berlin.