Interactive Multi-Attribute Electronic Negotiations in the Supply Chain: Design Issues and an Application Prototype

Michael Rebstock and Philipp Thun
University of Koblenz-Landau, Institute of Business Information Systems
Universitätsstr. 1, 56070 Koblenz, Germany
rebstock@t-online.de, philipp.thun@web.de

Abstract

We report on the development of a generic application for interactive electronic negotiations in the supply chain. We introduce the concept of interactive bilateral multi-attribute electronic negotiations and explain its significance. We specify an application architecture for this domain, showing three main components and four interfaces. We apply the ebXML framework to model the negotiation process and negotiation documents. As ebXML does not supply suitable process descriptions for generic, interactive electronic negotiations, we suggest a novel ebXML negotiation process model. Additionally, we develop ebXML document structures for this domain.

1. Introduction

Negotiations and contracts form the economic and legal links between organizations in their role as elements of the supply chain. Supply chain decisions within an organization are closely coupled to negotiations (about quantities, times, prices, product specifications etc.) with other organizations of the supply chain. Supporting these negotiations electronically, and thus supplying a frictionless electronic integration of the supply chain, allows major progress in supply chain management and decision support.

Unsurprisingly, electronic negotiations are popular. Research [e.g., 1, 2, 3, 4, 5, 6] and application development [e.g., 7, 8, 9] has been performed in this area, often focusing on auctions and tenders. But whereas in the 1990s electronic negotiations have largely been a domain of stand-alone applications [e.g., 10, 11], today, with focus on supply chain transactions, their inter-organizational aspects become predominant.

In this paper, we focus on interactive bilateral multi-attribute electronic negotiations. Some work has already been done on bilateral and on multi-attribute negotiations, either conceptually [e.g. 12, 13, 14], or in application design and development [e.g. 15, 16, 17]. Contributions also come from the research field of electronic contracting [18].

Because of the inter-organizational character of electronic negotiations in the business domain, the use of standards is crucial. Electronic Business XML (ebXML) is a standard for global electronic business processes. Its development is lead and sponsored by UN/CEFACT and OASIS [19]. The standardization process is aimed at the exchange of electronic business documents using the Extensible Markup Language (XML) [20].

In our paper, we discuss the application architecture for interactive bilateral multi-attribute electronic negotiations. We use the ebXML Business Process Specification Schema (BPSS) to model the inter-organizational negotiation process and the corresponding documents. UML diagrams are used for graphical representations. As ebXML does not supply suitable process descriptions for generic, interactive electronic negotiations, we suggest a novel ebXML negotiation process model. We also develop new ebXML document structures for this domain.

2. Electronic negotiations

An electronic negotiation is a joint decision-making process of two or more parties within an electronic market [21]. The objective of this process usually is to establish a contract between the parties involved. In order to reach an agreement, the parties issue offers to their market partners. The numbers of parties involved in this process, its temporal and its logical conditions depend on the negotiation protocol chosen [22].

Surprisingly, electronic negotiations are often conceived as being either auctions, tenders or exchanges [e.g., 1, 10, 11]. Looking at real-world phenomena, we find that negotiations in the business-to-business domain to the largest amount are interactive bilateral multi-attribute negotiations. Often, these negotiations show a complex structure and a large variance in negotiation semantics. Examples are negotiations on:

Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)
0-7695-1874-5/03 $17.00 © 2002 IEEE

- Complex, configurable products with individually defined attributes, like in the computer or the automotive industry.
- Semi-structured, but not standardized products like chemicals with specific qualities or grades in the process industry.
- Framework contracts, e.g. in the automotive, the process or the retail industry.
- Service contracts, construction or engineering contracts in the respective industries.
- Contracts on industrial services bundles, i.e. on goods combined with accompanying services (financial, warranty, deployment, installation, service, maintenance, etc.). Examples range from the deployment of a computer lab (hardware, software, services) to the complete process of planning, building and deploying a power station.

As our goal is the electronic support of the main type of negotiations in the business-to-business domain, we focus on interactive bilateral multi-attribute electronic negotiations. Auctions, tenders and exchanges are not part of our analysis. We apply a process-oriented view, where we analyze negotiations as a sequence of tasks performed in pursuit of achieving a contract.

When performed within an electronic market, an electronic negotiation is an optional activity within the agreement phase of an electronic market transaction [22, 23]. This agreement phase may include other steps, like matching or scoring [24]. In this paper, we focus on the electronic support of the negotiation itself and will not deal with matching and scoring functions.

Also, we do not deal with the automation of negotiations. During the 1990s, the major objective of many projects was full automation of negotiation processes. Even though it has been expected since many years that full negotiation automation can be achieved [21], this still is the subject of controversial discussions [e.g., 25, 26, 27]. Some projects have not been developed further or have been terminated without successful realizations [e.g., 10, 11, 28]. Even more important, empirical results substantiate doubts that full automation is a desirable goal for negotiation applications at all. It was shown that users want to understand and influence the negotiation process and not leave the negotiation process to agents that only produce a final result [29, 30]. Consequently, our focus is on electronic negotiation process support.

3. Object framework

In a business object framework (BOF), business objects (classes), their relations and functions are defined for a given business application domain. Like an ontology in general [31], the BOF is a set of definitions of content-specific knowledge representation primitives. Work has already been done and is being done on ontologies in the domain of electronic markets (for an overview cf. [32, 33]). Standardization approaches (like EDIFACT, BME-cat, Biztalk or others; for an overview cf. [34, 35]) define syntactic and semantic structures of documents exchanged in electronic market transactions.

All these approaches share the problem that a business object framework, once elaborated, is static and thus restricts the flexibility of electronic negotiation applications. As our objective is a generic and flexible application, in our approach we have to consider heterogeneous negotiation situations and the potential use of different, not necessarily standardized, frameworks.

Electronic negotiations in the domains described above may even make it necessary to flexibly adjust the business object framework prior to or during a transaction. As generic negotiation applications may be a functional part of an electronic marketplace with a wide range of products and services, framework re-definition may become necessary if the application's business object framework does not meet the requirements of specific companies, industries or goods. In traditional face-to-face negotiations, the strategy of bringing in an additional negotiable attribute during the process of the negotiation in order to expand the potential agreement space is also straightforward and often practiced; markets partners generate win-win situations by doing so.

These strategies can be transferred to electronic negotiations. Consequently, we suggest an evolutionary approach to the BOF problem. We add a negotiation meta-layer (for the "negotiation of negotiation semantics") to our model and thus link the process of managing negotiation semantics to the negotiation process itself. Not only the negotiation of the values of the negotiation attributes, but also the definition or change of its framework may thus be performed interactively.

In our model, negotiations start from an initial set of attributes. These can be defined by common standards or may be company specific. (The latter often is the case for a sell-side or buy-side system dominated by one large company.) If necessary, a market partner can then request the addition, re-definition or deletion of an attribute of the initial set. Thus, the negotiation can be flexibly adjusted to the needs of the parties involved. (Note that we do not deny the necessity of standardization work. Only where necessary, the evolutionary development as described will be used.)

4. Application architecture

The work we report on in this paper is part of an ongoing research project on the development of generic application components supporting interactive bilateral multi-

attribute electronic negotiations in an inter-organizational context [36]. Main objectives are an architectural design suitable for different industries, company sizes and products as well as a communication interface design that allows the integration of inter-organizational with intra-organizational applications. Besides conceptual aims, the project also serves exploratory purposes, as different architectural options are designed, implemented and evaluated.

Derived from our analysis above, in our model we distinguish three application layers, the (1) negotiation meta-layer, (2) negotiation layer and (3) communication layer. As can be seen in Figure 1, each layer is represented by a major component of our application architecture [37, 38]:

- BOF Engine: This component supports the management of the application's business object framework as described above (Layer 1).
- Negotiation Engine: This component supports interactive, bilateral multi-attribute electronic negotiations as described in this paper (Layer 2).
- Communication Engine: This component supplies communication functions. It handles incoming and outgoing messages and provides authentication and encryption functionality. It also supplies workflow functionality to manage the negotiation process flow with internal or external, human or electronic agents (Layer 3).

The three engines are conceptualized as large-grained electronic commerce components that expose standard interfaces based on XML messages, so as to facilitate communication with other components or services locally as well as on the Web. General functionalities like service directories, catalog services or settlement services are conceptualized as external components and are therefore not part of our architectural model.

All components are programmed in Java and thus can be implemented on client-side as well as on server-side. Client-side instances (using trusted Java Applets) are necessary if local security functions are to be implemented [38]. Still, the component as a whole or some of its modules can also be implemented server-side (using Java Servlets). Indeed, both alternatives have been conceptualized and partly realized in the project.

We have identified and defined four interfaces of the application. These interfaces support the exchange of messages with different types of collaborating applications on the inter-organizational as well as the intra-organizational level. Depending on the type of interface, messages hold operational data (negotiation content or input), workflow data, BOF structures or front-end input/output data. All messages are XML-based.

Interface 1, the BOF application interface, is used for the communication with external business object framework applications (proprietary or official standards). These are usually developed by a third party or consortium and can be accessed via the Web. To be able to cope with diverse transaction semantics, the e-negotiation application uses a flexible, generic object framework (which can be instantiated with public as well as proprietary object frameworks). We will discuss the corresponding class model further down.

The e-market application interface (interface 2) is the connector to inter-organizational systems, such as electronic marketplaces or inter-organizational supply chain planning and optimization applications.

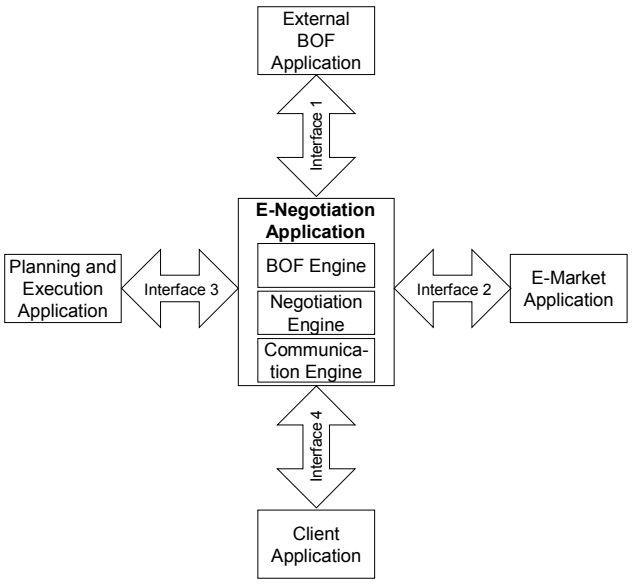

Figure 1. Application and interface architecture

The planning and execution application interface (interface 3) is used for message exchange with in-house applications (usually ERP systems). The importance of this interface results from the fact that, within supply chain relations, contract decisions cannot be made without reference to internal objectives and restrictions of the respective organizations. Although the use of this interface leads to additional requirements concerning security and confidentiality, it is necessary, if feasible negotiation decisions and a frictionless information flow along the supply chain are to be achieved. Interface 3 also serves as a connector between the e-negotiation application and workflow management applications (often a functional part of ERP applications). With the negotiation process being part of an overall supply chain process, parts of this overall process may be controlled by the respective companies' internal applications.

With interfaces 2 and 3, the e-negotiation application as a whole serves as an integration point of intra-

organizational planning and allocation processes with inter-organizational market processes.

Finally, interface 4, the client interface, allows the exchange of structured (XML-based) negotiation documents with different types of client applications, especially browser-based portals, or proprietary front-end applications.

During a negotiation, all communications are mediated by the e-negotiation application via its four interfaces. If a negotiation is ended successfully, i.e., if both parties express their approval of the attributes of a transaction, a contract is closed. In a final step, the e-negotiation application supplies the negotiation results to in-house (ERP) systems and, if existent, supply chain planning tools for further processing. Follow-up processing of the contract may include direct information exchange between these systems for, e.g., dispatch advices or financial settlement information.

5. Process modeling

In the next chapters, we report on two specific tasks of our application development: the negotiation process definition and the negotiation document definition.

With the prerequisites described above, inter-operable, standards-based processes and messages are key features of an e-negotiation application supporting the supply chain. We design the communication and process flow of our application completely based on an emerging, but already widely discussed standard, ebXML. The openness of ebXML allows to define processes and documents in a standardized way and still maintain the flexibility and generic applicability introduced as objectives above. [19]

In ebXML, a business process specification defines the information flow of business transactions, i.e. the process structure for the exchange of business documents between business partners. Several atomistic transactions can be combined to complex collaborations. The Business Process Specification Schema (BPSS) [39], which is used in this context, is available as XML DTD as well as W3C XML Schema document [40]. The complete process model is represented by one XML document, elements of which we describe in the following. To show the correspondence between the ebXML framework and UML diagrams (and to make the XML extracts easier understandable), we use UML diagrams [41] additionally to the XML document extracts - even if some of the diagrams may seem trivial.

Innovative aspects of our concept concerning the process model are the novel process patterns we suggest. These especially deal with the aspect of interactivity, which is reflected in the way offers and counter-offers are being dealt with.

The option of submitting counter-offers, in our view, is one of the fundamental aspects of a negotiation. In the understanding of existing ebXML pattern suggestions [42], an offer can only be accepted or rejected. A counter-offer is equal to a rejection. (In the rejection case, referring to a counter-offer is optional even if the negotiation is to be continued.) Thus the different stages of a negotiation - represented by consecutive counter-offers - cannot be identified as being part of the same collaboration without additional effort; an effort, that usually results in higher application complexity. Our approach puts the necessary reference information into the documents exchanged, so that discovery of referring documents, and thus accurate interpretation of the business partner's intention and the reconstruction of the negotiation process as a whole, is as easy as possible.

5.1 Process overview

Within a negotiation (i.e., the agreement phase of an electronic transaction), we identify two processes to be modeled. These are related to the two document types used, "offer" and "contract": the (1) offer negotiation process ("RequesterNegotiation") and (2) the closing of the contract ("RequesterContract"). Figure 2 shows the respective UML use case diagram, Figure 3 the corresponding part of the Business Process Specification (to enhance readability of the XML extracts, we do not state the complete number of attributes in this and the following XML documents).

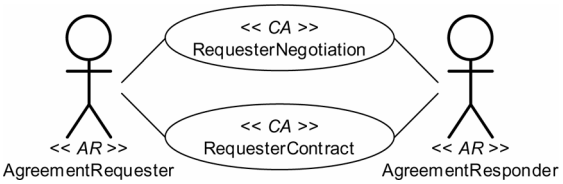

Figure 2: Use case diagram of "ReachAgreement"

Within this part of the BPSS, a collaboration (<*BinaryCollaboration*>) named "ReachAgreement" is defined, to which an <*InitiatingRole*> and a <*RespondingRole*> are assigned. Within our generic application scenario, an initial offer can be an offer to sell as well as an offer to buy. For this reason, the roles defined are not "Buyer" and "Seller", but "AgreementRequester" and "AgreementResponder".

The two processes are modeled as independent complex collaborations. They are specified within the XML document at another place. Here, they are instantiated. Two roles are assigned to each of the two collaborations. These roles are linked to the level of the "ReachAgreement" collaboration by the attributes *fromAuthorizedRole* and *toAuthorizedRole*.

```
<BinaryCollaboration
  name="ReachAgreement">
    <InitiatingRole
      name="AgreementRequester"
      ... />
    <RespondingRole
      name="AgreementResponder"
      ... />
    <CollaborationActivity
      name="RequesterNegotiation"
      binaryCollaboration="ConductNegotiation"
      fromAuthorizedRole="AgreementRequester"
      toAuthorizedRole="AgreementResponder"
      ... />
    <CollaborationActivity
      name="RequesterContract"
      binaryCollaboration="CloseContract"
      fromAuthorizedRole="AgreementRequester"
      toAuthorizedRole="AgreementResponder"
      ... />
```

Figure 3: BPSS definition of "ReachAgreement" (Part 1)

To describe the states of the process flow we use a UML statechart diagram (Figure 4). States are depicted as described in the corresponding ebXML document (Figure 5).

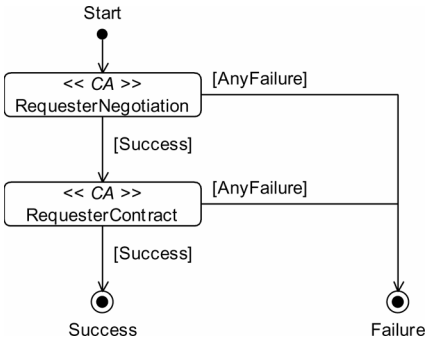

Figure 4: Statechart diagram of "ReachAgreement"

As can be seen from both representations, each of the collaboration activities defined above ("RequesterNegotiation" as an instance of "ConductNegotiation" and "RequesterContract" as an instance of "CloseContract") is a possible state within the process. Additionally, one start and two end states (<*Success*>, <*Failure*>) of the process have been modeled.

```
    <Start
      toBusinessState="RequesterNegotiation"
      ... />
    <Transition
      fromBusinessState="RequesterNegotiation"
      toBusinessState="RequesterContract"
      conditionGuard="Success"
      ... />
    <Failure
      fromBusinessState="RequesterNegotiation"
      conditionGuard="AnyFailure"
      ... />
    <Success
      fromBusinessState="RequesterContract"
      conditionGuard="Success"
      ... />
    <Failure
      fromBusinessState="RequesterContract"
      conditionGuard="AnyFailure"
      ... />
</BinaryCollaboration>
```

Figure 5: BPSS definition of "ReachAgreement" (Part 2)

5.2 Negotiation

In Figure 6, we take a closer look at the first of the two processes, "ConductNegotiation". (As the structure of the XML definition of this collaboration is very similar to the definition above, we do not state the XML representation of the process.)

Because roles on the different specification levels of the process are directly tied to each other, the roles of "AgreementRequester" and "NegotiationRequester", as well as "AgreementResponder" and "NegotiationResponder", are identical. Within this collaboration we find two Business Transaction Activities ("BTA"). Both are instances of the same transaction ("SubmitProposal"), with only the role assignments changed. Figure 7 shows the definition of this transaction, as part of the Business Process Specification.

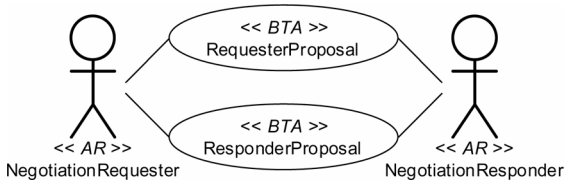

Figure 6: Use case diagram of "ConductNegotiation"

```
<BusinessTransaction
  name="SubmitProposal"
  ... >
    <RequestingBusinessActivity
      ... >
       <DocumentEnvelope
         businessDocument="NegotiableProposal"
         ... />
    </RequestingBusinessActivity>
    <RespondingBusinessActivity
      ... >
       <DocumentEnvelope
         businessDocument="NPAcceptance"
         isPositiveResponse="true"
         ... />
       <DocumentEnvelope
         businessDocument=
           "NPCounterproposalAdvice"
         isPositiveResponse="true"
         ... />
       <DocumentEnvelope
         businessDocument="NPRejection"
         isPositiveResponse="false"
         ... />
    </RespondingBusinessActivity>
</BusinessTransaction>
```

Figure 7: BPSS definition of "SubmitProposal"

<*BusinessTransaction*> is described by a requesting and a responding activity. These activities refer to business documents that are sent and received by the process.

The document structure is defined in separate specifications. Figure 8 shows an example of the specification of the proposal document. As part of the specification, the location of the structure definition is stated. (Note that the location statement here is not referring to an existing location.)

```
<BusinessDocument
  name="NegotiableProposal"
  specificationLocation=
    "http://www.ebxml.org/bd/NP.xsd"
  ... />
```

Figure 8: BPSS definition of "NegotiableProposal"

The other, responding documents of this transaction do not contain any payload. They serve only as an expression of the negotiation partner's intention to accept or reject the proposal or to submit a counter proposal. The relevant information therefore is already contained in the name of the document transmitted, "NPAcceptance", "NPRejection" or "NPCounterproposalAdvice". The content of the documents can remain unfilled. Figure 9 gives an overview of the document flow.

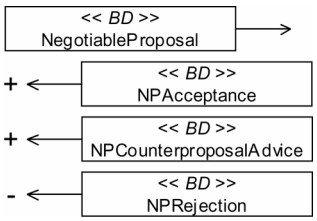

Figure 9: Document flow of "SubmitProposal"

(Plus and minus attributed to the responding flows signify the value of the boolean attribute *isPositiveResponse*. This attribute allows a quick and doubtless interpretation of the document's meaning by the receiving party.)

Figure 10 shows the UML statechart diagram for this process. If "NPAcceptance" is transmitted, the negotiation has been terminated successfully. In this case, the following process, "CloseContract", starts. If the offer proposal is rejected ("NPRejection"), the negotiation is terminated with a final negative state. This state leads to the end of the whole agreement process (cf. Figure 4). If the responding business partner wants to submit a counter proposal instead, the responding document "NPCounterproposalAdvice" informs the business partner that a counter-offer will be transmitted.

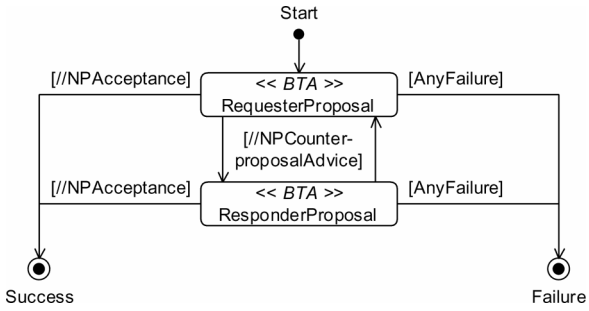

Figure 10: Statechart diagram of "ConductNegotiation"

The submission of the counter-offer itself is performed within a new transaction. The reason for this additional transaction lies in the ebXML structure definition of Business Transactions [39]. As ebXML requires exactly one requesting and only one or none responding document for each transaction, subsequent counter-offers have to be transmitted within new transactions.

5.3 Closing of contract

Figure 11 depicts the use case of the second process within the agreement phase, "CloseContract". This collaboration only contains one transaction activity, "RequesterOffer", which is an instance of the transaction "SubmitOffer". Figure 12 shows the corresponding XML description.

Figure 11: Use case diagram of "CloseContract"

```
<BusinessTransaction
  name="SubmitOffer"
  ... >
    <RequestingBusinessActivity
      ... >
        <DocumentEnvelope
          businessDocument="BindingOffer"
          ... />
    </RequestingBusinessActivity>
    <RespondingBusinessActivity
      ... >
        <DocumentEnvelope
          businessDocument="BOAcceptance"
          isPositiveResponse="true"
          ... />
        <DocumentEnvelope
          businessDocument="BORejection"
          isPositiveResponse="false"
          ... />
    </RespondingBusinessActivity>
</BusinessTransaction>
```

Figure 12: BPSS definition of "SubmitOffer"

The structure of the transaction shows similarities to the transaction previously modeled. The requesting document ("BindingOffer") of this process contains the contract parameters that have been negotiated upon in the previous process (Figure 13). The contents of this document thus are identical to the finally accepted offer proposal. Besides its contents, the document also carries the digital signature of the requester for legal and security reasons. As a response, only two documents are valid: acceptance ("BOAcceptance") or rejection ("BORejection").

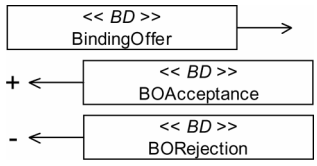

Figure 13: Document flow of "SubmitOffer"

The reason for this is that the only purpose of this process is the legally binding closing of the contract. Both partners have already agreed upon the contents of the contract in the previous process. Still, a final rejection is possible at this point. Similar to the previous process, "BORejection" expresses the business partner's intention already by the fact of its creation. Like above, the document's envelope is the message, its content is blank. "BOAcceptance", on the other hand, has to hold the full content of "BindingOffer" (including the signature of "ContractRequester"). Additionally, it bears the digital signature of "ContractResponder".

Figure 14 shows the corresponding statechart diagram. The process is initiated by the successful termination of the previous process ("ConductNegotiation"). The process's sole transaction is "RequesterOffer". Only if this transaction is terminated successfully, the higher-level collaboration ("ReachAgreement") reaches a positive end state (cf. Figure 4).

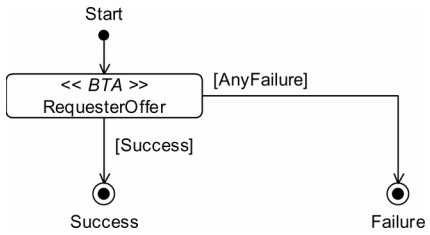

Figure 14: Statechart diagram of "ConductContract"

6. Data modeling

Existing ebXML patterns, concerning negotiations, only provide process descriptions. In this chapter, we develop a class model for electronic negotiation document structures. The innovative aspect of our approach is that it supports the evolutionary approach to business object framework definition as described above.

Usually, the electronic negotiation's document structures follow the business object framework applied. But by existing e-negotiation approaches, the question of business object frameworks is not necessarily addressed: Some projects use ad-hoc frameworks; others do not use an explicit framework at all (like most agent-based approaches, as they only negotiate one attribute) [e.g., 10, 11]. Still others elaborate business object frameworks for specific applications that typically cannot be generalized [e.g., 1, 43]. Only a few papers investigate into the general aspects of specifying frameworks for electronic negotiations [13].

In the following, we develop definitions for the documents (1) "NegotiableProposal" and (2) "BindingOffer", that allow for the flexibility described above and complement the ebXML framework. (These two documents contain the payload of the collaboration discussed above. The other documents of the collaboration need not be discussed further: the content of "BOAcceptance" is, as described above, identical to "BindingOffer", additionally it only bears digital signatures. The rest of the collaboration documents contain, as discussed above, no payload, and thus do not have an internal structure.)

6.1 Contract document

Following our evolutionary approach to framework and document management, we use a contract document ("BindingOffer") that can be composed of any number of contract attributes ("Attribute") and any number of contract items ("Item") (Figure 15). Contract attributes contain general contract related information, such as terms of delivery, terms of payment, currency, etc. Items contain information about products and services. They, as the offer header, are composed of any number of attributes. Item attributes can be product name, description, amount, price, etc.

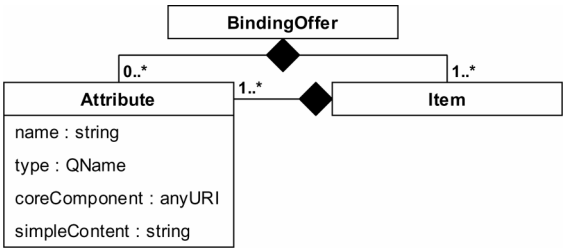

Figure 15: Class diagram of "BindingOffer"

Besides its name, data type and content, every attribute bears an optional reference to an ebXML Core Component [44]. By this reference, standardized semantics can be ascribed to attributes. (ebXML Core Components are managed by the ebXML consortium and can be retrieved from the ebXML Core Library.)

The data model is expressed by a W3C XML Schema document, where we also use W3C XML Schema data types. These data types can be used to describe the data format of attributes in an instance document as well. Figure 16 shows an example of an instance of the "BindingOffer" document. This example includes, among other elements, a global currency attribute ("currency"). This currency attribute refers to Core Component No. 133 [45]. Using this reference, an application is able to perform actions appropriate to the semantic value of the attribute - in this case, for instance, perform an automated currency conversion.

6.2 Negotiation document

The structure of the negotiation document ("NegotiableProposal") might be considered as being similar to the structure of the contract document. At a closer look though, as it serves additional purposes, the negotiation document structure has to be more complex (Figure 17).

Its root element contains a "name"-attribute that allows referencing. It also contains a sub-element "Version", which includes a "timestamp". The reason for this is that, in our model, "NegotiableProposal" records the complete history of the negotiation. This allows referring to previous stages of the negotiation during the collaboration. As all changes are contained in each version of the document, no additional application functionality is needed to collect negotiation history data.

```
<BindingOffer
  ... >
   <Attribute
    name="currency"
    type="xsd:string"
    coreComponent=
      "http://www.ebxml.org/cc/000133">
      EUR
   </Attribute>
   <Item>
      <Attribute
       name="productNumber"
       type="xsd:string">
         1234-5678
      </Attribute>
      <Attribute
       name="productName"
       type="xsd:string">
         Notebook Computer
      </Attribute>
      <Attribute
       name="productDescription"
       type="xsd:string">
         Mobile ABC Processor 2 GHz
         512 MB RAM
         ...
      </Attribute>
      <Attribute
       name="orderQuantity"
       type="xsd:integer">
         12
      </Attribute>
      <Attribute
       name="orderPrice"
       type="xsd:decimal">
         3456.78
      </Attribute>
   </Item>
</BindingOffer>
```

Figure 16: Document instance example

As described above, we allow that items can be changed, added to or deleted from a contract during a negotiation, and also that single attributes of an item or the contract in general can be changed, added or deleted. The sub-structure of the "Version"-element therefore is, compared to the structure of "BindingOffer", extended. Additional elements allow marking attributes or items as deleted or changed. Comments can be attributed to these changes.

7. Discussion

In this paper, we reported on the development of an application for the support of interactive bilateral multi-attribute electronic negotiations in the supply chain. We gave an insight into the architecture, the process and the communication structure of the application. We found that a major task in building the application was the modeling

and specification of the negotiation process flow and the negotiation object structure.

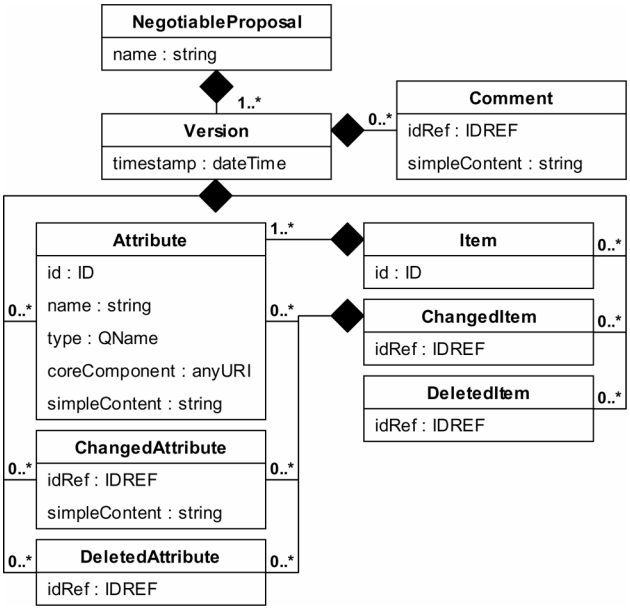

Figure 17: Class diagram of "NegotiableProposal"

Although the ebXML framework was used and generally found very suitable for the specification of process and document structures, existing ebXML patterns were found to be insufficient in our domain. Hence we developed a new, generic model of an interactive bilateral negotiation process using the ebXML Business Process Specification Schema. We also developed document specifications, which were found completely missing in our domain.

Based on the process and document specifications developed, we have built an application prototype for the support of interactive bilateral multi-attribute electronic negotiations. ebXML documents have proven well-suited in this work. First experiences with our prototype imply that the specifications suggested in this paper supply sufficient flexibility for the use in diverse application situations.

As the project is still ongoing, and the focus of the project at this stage is on design and prototyping, the application's impact on the performance of real-world negotiations has not be tested yet. After the completion of the development work, we will research the application's usability and usefulness as well as its performance impact on negotiations.

8. References

[1] Addis, M.; Allen, P.; Surridge, M. (2000). Negotiating for Software Services. Proceedings of the 11th International Workshop on Database and Expert Systems Applications (Tjoa, A.M. et al., Eds.), 1039-1043. IEEE, Los Alamitos, CA.

[2] Andreoli, J. M.; Castellani, S.; Munier, M. (2000). AllianceNet: Information Sharing, Negotiation and Decision-Making for Distributed Organizations. Electronic Commerce and Web Technologies (Proceedings of the EC-Web 2000. Bauknecht, K. et al., Eds.), 122-132. Springer, Berlin.

[3] Benyoucef, M.; Keller, R.K. (2000). A Conceptual Architecture for a Combined Negotiation Support System. Proceedings of the 11th International Workshop on Database and Expert Systems Applications (Tjoa, A.M. et al., Eds.), 1015-1019. IEEE, Los Alamitos, CA.

[4] Koetsier, M.; Grefen, P.; Vonk, J. (2000). Contracts for Cross-Organizational Workflow Management. Electronic Commerce and Web Technologies (Proceedings of the EC-Web 2000. Bauknecht, K. et al., Eds.), 110-121. Springer, Berlin.

[5] Limthanmaphon, B.; Zhang, Y.; Zhang, Z. (2000). An Agent-Based Negotiation Model Supporting Transactions in Electronic Commerce. Proceedings of the 11th International Workshop on Database and Expert Systems Applications (Tjoa, A.M. et al., Eds.), 440-444. IEEE, Los Alamitos, CA.

[6] Lo, G.; Kersten, G.E. (1999). Negotiation in Electronic Commerce: Integrating Negotiation Support and Software Agent Technologies. Interneg Report INR03/99, http://interneg.org/interneg/research/papers/1999/03.pdf, 1999-12-20.

[7] DiCarta (2001). http://www.dicarta.com, 2001-01-24.

[8] IBM (2000). Virtual Market Place (ViMP), http://www.zurich.ibm.com/activities/csc/csc_e-business-vimp.html, 2000-11-02.

[9] Menerva (2000). http://www.menerva.com, 2000-12-11.

[10] Chavez, A.; Maes, P. (1996). Kasbah: An Agent Marketplace for Buying and Selling Goods. Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (Crabtree, B. and Jennings, N., Eds.), 75-90. The Practical Application Company Ltd, Blackpool.

[11] Wurman, P.R.; Wellman, M.P.; Walsh, W.E. (1998). The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents. Proceedings of the 2nd International Conference on Autonomous Agents (Sycara, K.P. and Wooldridge, M., Eds.), 301-308. ACM Press, New York.

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

[13] Wedekind, H. (2000). On Specifying Contract Negotiations. Proceedings of the 8th European Conference on Information Systems, Volume 1 (Hansen, H.-R. et al., Eds.), 23-30. Vienna University of Economics and Business Administration, Vienna.

[14] De Paula, G.E.; Ramos, F.P.; Ramalho, G.L. (2001). Bilateral Negotiation Model for Agent-Mediated Electronic Commerce. In: Agent-Mediated Electronic Commerce III. Current Issues in Agent-Based Electronic Commerce Systems (Dignum, F. et al., Eds.), 1-14. Springer, Berlin.

[15] Interneg (2001). http://interneg.org, 2001-05-23.

[16] IBM Research (2001). SilkRoad. http://www.zurich.ibm.com/csc/ebizz/silkroad.html, 2001-05-23.

[17] Menerva (2001). http://www.menerva.com, 2001-05-23.

[18] Runge, A. (1998): The Need for Supporting Electronic Commerce Transactions with Electronic Contracting Systems. EM - Electronic Markets 8 (1).
[19] Electronic Business XML Joint Coordination Committee (2002): ebXML - Enabling A Global Electronic Market. http://www.ebxml.org, 2002-04-17.
[20] World Wide Web Consortium (2002): Extensible Markup Language (XML). http://www.w3c.org/XML, 2002-04-17.
[21] Davis, R. and Smith, R.G. (1983): Negotiation as a Metaphor for Distributed Problem Solving. In: Artificial Intelligence 20 (1), 63-109.
[22] Rebstock, M. (2001). Electronic Negotiations in Business-to-Business Marketplaces - A Framework and Four Propositions. JFB Journal für Betriebswirtschaft 51 (5-6), 208-214.
[23] Schmid, B. (1993): Elektronische Märkte. Wirtschaftsinformatik, 35 (5), 465-480.
[24] Ströbel, M. (2000). On Auctions as the Negotiation Paradigm of Electronic Markets. EM - Electronic Markets 10 (1), 39-44.
[25] Beam, C.; Segev, A.; Bichler, M. et al. (1999). On Negotiations and Deal Making in Electronic Markets. Information Systems Frontier 1 (3), 241- 258.
[26] Kersten, G.E.; Noronha, S.J. (2001). Negotiations in Electronic Commerce: Methodological Misconceptions and a Resolution. INR02/99. http://interneg.org/interneg/research/ papers/1999/02.pdf, 2001-05-23.
[27] Pradella, M.; Colombetti, M. (2001). A Formal Description of a Practical Agent for E-Commerce. In: Dignum, Frank et al. (eds.): Agent-Mediated Electronic Commerce III. Current Issues in Agent-Based Electronic Commerce Systems (LNCS Vol. 2003), 84-95. Springer, Berlin.
[28] Guttman, R.H., Viegas, F., Kleiner, A.F.III (2001). Tête-à-Tête. http://guttman.www.media.mit.edu/people/guttman/research/tete/tete.html, 2001-05-23.
[29] Guttman, R.H.; Maes, P.; Chavez, A. et al. (1997). Results from a Multi-Agent Electronic Marketplace Experiment. http://guttman.www.media.mit.edu/people/guttman /research/pubs/maamaw97.pdf, 2001-05-23.
[30] Liang, T.-P.; Dong, H.-S. (2000). Effect of Bargaining in Electronic Commerce. International Journal of Electronic Commerce 4 (3), 23-43.
[31] Gruber, T.R. (1992). Ontolingua: A Mechanism to Support Portable Ontologies, Technical Report KSL-91-66 (Final revision June 92), Knowledge Systems Laboratory, Stanford University. ftp://ksl.stanford.edu/pub/KSL_Reports/./KSL-91-66.ps.gz, 2001-05-23.
[32] Fensel, D. (2001). Ontologies. A Silver Bullet for Knowledge Management and Electronic Commerce. Springer, Berlin.
[33] Omelayenko, B. (2001). Integration of Product Ontologies for B2B Marketplaces: A Preview. SIGecom Exchanges Newsletter 2 (1), 19-25. http://www.acm.org/sigecom/exchanges/issue-2.1/Omelayenko.pdf, 2001-06-25.
[34] Willaert, F. (2001). XML-based Frameworks and Standards for B2B eCommerce. Dissertation, Katholieke Universiteit Leuven, 2001. http://www.ebxml.org/documents/ebxml-thesis.pdf, 2002-04-17.
[35] Frank, U. (2001). Standardisierungsvorhaben zur Unterstützung des elektronischen Handels: Überblick über anwendungsnahe Ansätze. Wirtschaftsinformatik 43 (3), 283-293.
[36] MultiNeg (2002). http://www.fbw.fh-darmstadt.de/multineg/, 2002-31-05.
[37] Rebstock, M. (2001). An Application Architecture for Supporting Interactive Bilateral Electronic Negotiations, In: Electronic Commerce and Web Technologies. Proceedings of the EC-Web 2001 (Bauknecht, K., Madria, S.K. and Pernul, G., Eds.), 196-205. Springer, Berlin.
[38] Rebstock, M.; Amirhamzeh Tafreschi, O. (2002). Secure Interactive Electronic Negotiations in Business-to-Business Marketplaces. In: Proceedings of the 10th European Conference on Information Systems. University of Gdansk, Gdansk.
[39] ebXML Business Process Team (2001). Business Process Specification Schema, Version 1.01. http://www.ebxml.org/specs/ebBPSS.pdf, 2002-02-12.
[40] World Wide Web Consortium (2002). W3C XML Schema. http://www.w3c.org/XML/Schema, 2002-04-17.
[41] Object Management Group (2002). UML Home Page. http://www.uml.org, 2002-02-12.
[42] ebXML Business Process Team (2001). E-Commerce Patterns, Version 1.0. http://www.ebxml.org/specs/bpPATT.pdf, 2002-02-12.
[43] Bichler, Martin; Klimesch, R. (2000). Simulation multivariater Auktionen - Eine Analyse des OTC-Handels mit Finanzderivaten. Wirtschaftsinformatik 42 (3), 244-252.
[44] ebXML Core Components Team (2001). Core Component Overview, Version 1.05. http://www.ebxml.org/specs/ccOver.pdf, 2002-02-12.
[45] ebXML Core Components Team (2001). Core Component Dictionary, Version 1.04. http://www.ebxml.org/specs/ccDICT.pdf, 2002-02-12.