Knowledge Extraction in Mass e-Negotiations

Willy Picard, Wojciech Cellary
Department of Information Technology
The Poznan University of Economics
Mansfelda 4, 60-854 Poznan, Poland
{picard, cellary}@kti.ae.poznan.pl

Abstract

In the context of economy globalization, the need for globally distributed negotiations involving a high number of negotiators communicating through the Internet becomes an important business issue. In such negotiations, the amount of information describing the negotiation process is too high to be easily understood by humans. In this paper, a negotiation support model adapted to mass e-negotiations is presented. The proposed model consists of a multi-facet analysis mechanism which provides synthesized views of the negotiation process, allowing to extract knowledge concerning various aspects of the negotiation process.

1. Introduction

Negotiation is a fundamental act in business. Every business transaction is based on a contract that has been previously negotiated. In the context of economy globalization, companies doing business with other companies all around the world need to negotiate at a global scale. Such negotiations are needed not only for multinational enterprises spread in many countries, but also for small and medium size enterprises, which are working more and more in an international environment.

Classical ways of conducting negotiations are not well adapted to negotiations at the global scale. People involved in a negotiation process are used to personally meet to exchange information and to confront their interests and goals. Personal meetings are, however, costly in terms of time and money, as well as difficult to organize, in particular if negotiators work in different countries. In classical negotiations only a small number of participants are involved.

With the rise of Internet, negotiators may be arbitrarily geographically distributed. Internet allows a potentially unlimited number of negotiators from the whole world to remotely negotiate on a given contract. Now, the problem arises to organize and manage remote negotiations conducted by a high number of negotiators of a range of a few tens or more, denoted here *mass e-negotiations*.

An attempt to achieve this goal is to delegate the responsibility of the negotiation from a human negotiator to a computer. In such case we talk about *automated negotiations*. The negotiation is said to be *fully automated* if negotiations are conducted by software agents without human intervention in the negotiation process. Research topics involved in automated negotiation are the following [Jennings 2001]:

- negotiation protocols defining types of participants, valid actions, negotiation states, and events that cause negotiation states to change [Andreoli 2001];
- establishment of ontologies defined as agreements among the negotiators about how the negotiation objects are defined and what is the meaning of these definitions. XML Schemas [Ströbel 2001][Fallside 2001] and UML [Cranefield 1999][Rumbaugh 1998] have been proposed as candidates to the design of ontologies;
- decision-making models that are used by software agents to achieve their goals.

In the case of multi-attribute contracts with both aggregable attributes (e.g. price, quantity, etc.) and non-aggregable attributes (e.g., legal clauses, appendices, quality clauses, etc.), automated, humanless negotiations are not a viable solution. Software agents cannot operate on non-aggregable attributes, because of the lack of semantics concerning these attributes. We conclude that negotiations on multi-attribute contracts have to be conducted by humans. However, humans without any support are unable to deal with negotiations involving a high number of negotiators. The amount of data generated during such a negotiation process is too high to be understood by humans. Therefore, negotiation support systems are required that may facilitate mass e-negotiation processes conducted via the net.

A negotiation support system is particularly important in a case of mass e-negotiations conducted via the net, because it is almost impossible to remember all the

Witold Abramowicz, Gary Klein (eds.), Business Information Systems, Proceedings of BIS 2003, Colorado Springs, USA

propositions made by a great number of negotiators. In mass e-negotiations, in which the number of negotiators is high, and the number of proposals is very high, a negotiation process is possible only if negotiators are provided with synthetic views of the negotiation process. A fundamental element of every negotiation strategy is the planning process ([Lewicki 2001], pp. 40-51) based mainly on various analyses of the current status of negotiations. In mass e-negotiations, negotiators cannot conduct these analyses manually, because the amount of data to be analyzed is too high. Therefore, in this paper an analysis mechanism is proposed to be integrated into the negotiation support system. Moreover, a mass e-negotiation support system has to provide negotiators with a possibility of various analyses of contract versions authored by different negotiators to well understand different aspects of a conducted negotiation process. For instance, a negotiator may want to analyze the involvement of different negotiators in the negotiation process, or analyze the correlation between a contract clause defining a delivery date and other contract clauses. The proposed *multi-facet analysis* mechanism permits extracting knowledge concerning various aspects of the negotiation process.

The analysis mechanism should be generic enough to allow dealing with contracts of different types. Two categories of analysis techniques are distinguished: semantic and syntactic. Semantic analysis techniques are based on ontologies. An ontology is an agreement among the negotiators (humans or software agents) concerning the semantics associated with terms in a given area of knowledge. Building ontologies for multi-attribute contracts, with both aggregable attributes (e.g. price, quantity, etc.) and non-aggregable attributes (e.g., legal clauses, appendices, quality clauses, etc.), is usually a very difficult task. The set of terms used in such a contract is large, while the relationships between them, are usually complex. Moreover, an ontology depends on the domain related to the transaction defined in the contract. Therefore, each new contract may potentially require a new ontology. Moreover, the completeness of an ontology for a given contract can never be assumed, as a new proposal may use new terms which were not defined in the ontology attached to the contract being negotiated. An ontology-based solution is then not a promising approach to e-negotiations of complex contracts.

On the contrary to the ontology approach, which is a semantic one, in this paper a syntactic analysis is proposed based on versioning techniques [Cellary 1990]. The proposed syntactic analysis is based on relationships between negotiators' proposals. These relationships may be captured be a proper contract model. In this paper, a negotiation process is modeled as *a multiversion contract*. One may notice that a contract, which is under negotiation, is usually modified many times until the final agreement. Various versions of the contract reflect various propositions made by negotiators. Information concerning relationships between negotiators' proposals may be retrieved from the multiversion contract model and be the object of various analyses.

In this paper, we propose a negotiation support model adapted to mass e-negotiations. The proposed solution consists of a multiversion contract model and a multi-facet hierarchical analysis mechanism based on ultrametrics. The multi-facet hierarchical analysis mechanism provides synthetic views of the negotiation process, extracting knowledge related to various aspects of the negotiation process.

The paper is organized as follows. In Section 2, a multiversion contract model that addresses the problem of capturing relationships between negotiators' proposals is presented. In Section 3, a multi-facet negotiation analysis mechanism allowing for extraction of knowledge concerning various aspects of the negotiation process is discussed. Section 4 concludes the paper.

2. Capturing the Relationships between Proposals

In the proposed contract model, a contract consists of a number of versions. Various versions of a contract are organized hierarchically. The tree root is the initial contract version. When a new negotiator joins the negotiation, she/he must derive a version of an existing contract version. Each contract version is identified by a unique identifier.

A right choice of a contract version identifier can capture the tree structure and the "is-owner" relationship with related negotiators. As those two aspects are orthogonal, the contract version identifier consists of two independent parts:

- the identifier of the owner;
- a subidentifier which identifies the position of the contract version in the version tree.

Subidentifiers are responsible for capturing the structure of the contract version tree. Having a given contract version, we have to be able to find its parent and children versions. The concept of subidentifiers is the following:

Rule 1. If a contract version is the n-th child of a contract version whose subidentifier is p, the subidentifier of the child contract version is p.n. The root contract version subidentifier is 0.

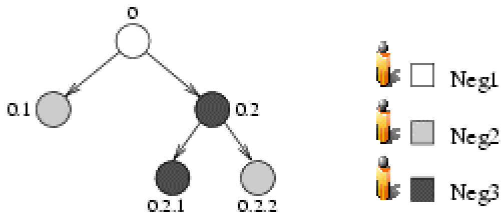

Figure 1. An example of a contract version tree

A simple contract version tree is presented in Figure 1. Three negotiators are involved in the negotiation process. Negotiator NEG1 starts the negotiation process with the publication of the root version whose subidentifier is 0. Subidentifiers of other versions are built according to Rule 1. For example, the first derived version is the version derived by negotiator NEG2 from the contract version whose subidentifier is 0. According to Rule 1, its subidentifier equals to 0.1.

A contract consists of *members*. A member may be, for instance, a paragraph, multimedia data, e.g. a picture or a digital signature, or a representation of the structure of the contract. In the contract model it is assumed that a *multiversion* contract consists of *multiversion* members, while a given contract version consists of given versions of these members. It is assumed that all the versions of a contract are composed of the same set of members. Differences between contract versions are reduced to differences between member versions. If a member is missing in a given contract version, the version of this member in this contract version is null.

Adding a new member m to a given contract version *v* causes the addition of a multiversion member M to the whole multiversion contract. The value of a newly added multiversion member is m for contract version v, and it is null for all the other contract versions.

Some multiversion members may point to other members to capture the structure of the contract. Such members are called *composite multiversion members*. A simple multiversion contract may consist of several multiversion members, each one modeling a contract paragraph, and a composite multiversion member modeling the structure of the contract as a paragraph list. Another contract with additional semantics may consist of multiversion members modeling various, semantically different contract parts, a multiversion member modeling the price, another multiversion member modeling the warranty, etc. A more complex contract may consist of composite multiversion members to capture a tree structure of contract parts. The paragraph concerning the price may then be part of a section concerning offers, which is a part of the contract. At a higher level of abstraction, the structure of the contract may be complex, modeling semantics of various parts of a contract, e.g. addenda.

In the proposed contract model, a fixed structure of contracts is not assumed. The proposed contract model does not limit contracts with regard to their structure and allows new contract structures to be built on the top of the proposed multiversion contract model. Therefore, advanced contract structures (e.g. tree structured contracts) may be built using the concepts of multiversion members and multiversion composite members proposed in the multiversion contract model.

Information concerning an agreement among two or more negotiators on a given contract member may be captured by the multiversion contract model. When two or more negotiators agree on a given contract member, the same member version occurs in various contract versions. The concept of *member instance* is proposed to capture member version sharing among contract versions. The value of a member instance is a member version. A member instance is associated with one or more contract versions. One-to-many relationships between member instances and contract versions are implemented as *association tables*.

An association table associates each member instance with at least one contract version. An association table consists of rows, one row per member instance. Each row is a pair (memberInstanceID , set of contract versions associated with the given member instance).

Figures 2 and 3 illustrate the representation of two multiversion members of a multiversion contracts. The contract version tree is assumed to be the one presented in Figure 1. The contract consists of a price and a warranty. The structure of the association table for the price is presented in Figure 2, while the structure of the association table for the warranty is presented in Figure 3. Analyzing simultaneously association tables for price and warranty, one may notice that price and warranty are independent clauses, each new warranty corresponding to a new price and vice-versa. This result is obtained only from syntactic information, i.e. the relationships between negotiators' proposals. No semantics concerning price or warranty is known. Therefore, association tables may be used as the basis for analysis of the negotiation process.

3. Multi-facet Analysis

The goal of multi-facet analysis is to provide a synthetic view of an aspect of the negotiation process. To analyze the multi-thread history and the current status of a negotiation, both the abstract objects to be analyzed and the analysis criteria must be defined. Therefore, the multi-facet analysis consists of two parts: definition and

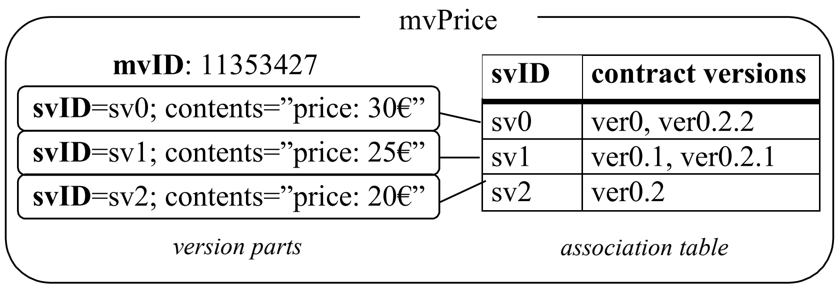

Figure 3. An example of a multiversion member of price

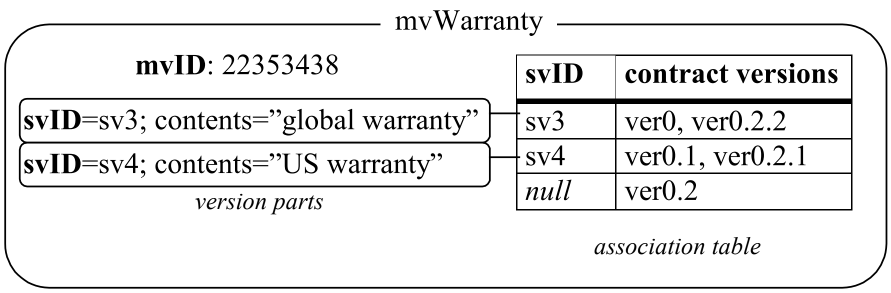

Figure 2. An example of a multiversion member of warranty

retrieval of data concerning a given facet of a negotiation process, and classification of the retrieved data.

Domain objects are used to model various facets of the negotiation processes. Domain objects may for instance represent the activity of negotiators, the importance of paragraphs, etc. As a consequence, domain objects must be flexible enough to represent various data types. An analysis domain is a set of domain objects modeling a facet of a negotiation process.

Formally, let D_{facet} denote the analysis domain modeling a facet of a negotiation process, denoted facet. The analysis domain D_{facet} is a set of domain objects denoted DO_i.

A domain object DO_i consists of:

- a unique identifier, denoted do_i,
- a set of attributes, and
- a type.

An attribute is a pair (name, value). Each attribute models a property of the domain object. To illustrate the use of attributes, let us assume that a negotiator is modeled by a domain object de-noted DO_{neg}. The attributes of DO_{neg} are pairs ("firstName", "John"), ("lastName", "Smith"), and ("represents", "ACME Corp.").

Domain objects are generated by an *Analysis Domain Function* (ADF). An ADF is a function whose image is an analysis domain.

Formally,

$$f \text{ is an ADF} \Leftrightarrow \begin{cases} f \text{ is a function on analysis domains } D_{orig_i} \\ \mathrm{Im}(f) = \{DO\},\ DO \text{ are domain objects} \end{cases}$$

When two or more analysis domains exist for an ADF, the function is said to be multi-variable. A special analysis domain, denoted $\varnothing$, is defined by $card(\varnothing) = 0$. The existence of the analysis domain $\varnothing$ allows to distinguish *transformer functions* from *generator functions*.

An ADF f is a generator function iff only one origin domain of f exists that is $\varnothing$. A generator function creates an analysis domain without the need of pre-existing data in the form of an analysis domain. A generator function may, for instance, generate the number π, or retrieve association tables from a multiversion contract model.

An ADF f is a transformer function iff at least one origin domain of f is different from $\varnothing$. A transformer function transforms an existing analysis domain into another analysis domain. A transformer function may for instance transform an analysis domain modeling association tables into another analysis domain representing the number of version of paragraphs.

A new programming language, named Analysis Domain Language (ADL), is used to define ADFs. ADL is a dialect of XML — the eXtensible Markup Language. ADL is based on four elements: *Metaobjects*, *ObjectSets*, *Tags*, and *Functions*. Metaobjects correspond to domain objects. ObjectSets correspond to analysis domains. Tags are basic elements of processing. Functions correspond to ADFs. ADL is described in details in [Picard 2002].

The ADL language allows to define new ADF processing domain objects modeling new aspects of the negotiation process. The choice of a facet of a negotiation process corresponds to the choice of an ADF. The result of the execution of an ADF is an analysis domain, i.e. a set of domain objects. An ADF defines a facet of the negotiation process to be analyzed, generating domain objects modeling the given facet.

An analysis criterion is an ADF with two analysis domains, each of them containing only one domain object. The resulting domain contains only one domain object modeling the distance between two domain objects.

Given an analysis domain and an analysis criterion operating on this analysis domain, it is possible to generate a hierarchical classification. The chosen hierarchical classification algorithm, named *single-link hierarchical classification algorithm*, is based on ultrametrics 0. However, analysis criteria only have to be metrics, as it has been proven that an ultrametric can be derived from each metric 0.

In Figure 4, a graphical representation of the multi-facet analysis process is given. A selected ADF (left-side arrow in Figure 4) extracts data from the multiversion contract and generates a set of domain objects to be analyzed. Then, a chosen analysis criterion (right side arrow) is used to process the classification of the previously generated objects. As the result, a hierarchical object classification is obtained.

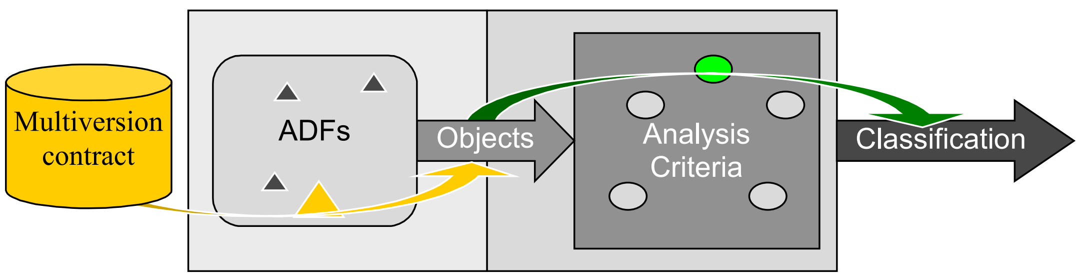

Figure 4. Multi-facet Analysis Process

As domain objects may model complex views of a negotiation process, and the interests of a given negotiator may be different from interests of other negotiators, many analyses may be performed on the same domain objects. Having an analysis domain modeling association tables of all multiversion members, a negotiator may be interested in influence of a given multiversion member on others, while another negotiator may be interested in the degree of controversy measured as a number of versions of each multiversion member. For this reason, the concept of *parametric analysis* is proposed. An analysis is parametric if various criteria may be used to perform various analyses of a given analysis domain.

In the proposed model, analyses are classifications. A classification groups domain objects according to their proximity. The concept of proximity can be considered as the similarity between items. The more two items are similar, the closest their are. Therefore, an analysis criterion is a metric on a given analysis domain.

Analysis criteria are a subset of transformer functions, so they are ADFs. Therefore they can be defined in ADL.

To illustrate the above technique, consider an example of a negotiation process analysis. In this example, we want to evaluate the weight of the various contract members in the negotiation process. We assume that the more a given member has been modified, the higher the interest of this member is. An analysis domain function is used to generate an analysis domain that consists of a set of domain objects denoted DO_i. Each object DO_i represents the level of interest of a multiversion member. The set of attributes of each object is restricted to only one attribute whose name is *numberOfVersion* and whose value is set to the number of versions of the corresponding multiversion member. For simplicity, we assume that only five contract members exist, and DO_1, DO_2, ... DO_5 are the various objects corresponding to the multiversion members. The values of the analysis criterion d for all pairs of domain objects are given in Table 1.

Table 1. Values of the analysis criterion *d* for all pairs of domain objects

x	y	d(x,y)
DO_1	DO_2	7
DO_1	DO_3	5
DO_1	DO_4	5
DO_1	DO_5	7
DO_2	DO_3	7
DO_2	DO_4	7
DO_2	DO_5	3
DO_3	DO_4	1
DO_3	DO_5	7
DO_4	DO_5	7

The hierarchical classification built on the basis of theses values is presented in Figure 5.

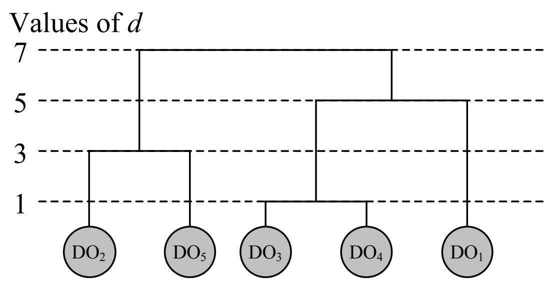

Figure 5. Classification of the level of interest of multiversion members according to the analysis criterion *d*

This hierarchical classification can be seen at various detail levels. Having a given threshold *T*, various partitions of the hierarchical classification may be processed. In Figure 6 (respectively Figure 7), the partition obtained with threshold *T*=4 (respectively *T*=6) is presented.

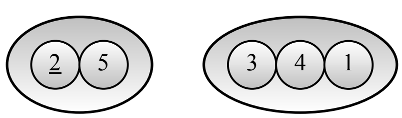

Figure 6. Partition with threshold T=4

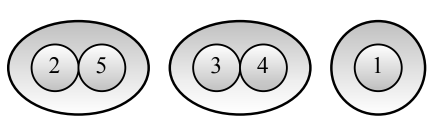

Figure 7. Partition with threshold T=6

Assume that the domain object DO_1 corresponds to the multiversion member concerning the price, denoted mvp1. The domain object DO_2 corresponds to the multiversion member concerning legal clauses, denoted mvp2. The domain object DO_3 corresponds to the multiversion member concerning delivery date, denoted mvp3. The domain object DO_4 corresponds to the multiversion member concerning delivery address, denoted mvp4. The domain object DO_5 corresponds to the multiversion member concerning the number of items, denoted mvp5. Assume that the number of versions of mvp1 is higher than the number of versions of mvp3, which is higher than the number of versions of mvp2.

At a high level of analysis, when threshold T=6, mvp1 and mvp3 are in the same class, while mvp2 is in another class. One can deduce that the price and the delivery date are of similar importance, while the importance of legal clauses is different. As the number of versions of mvp1 is higher than the number of versions of mvp2, one can conclude that the class consisting of mvp1, mvp3, and mvp4 contains parts of higher importance than the class consisting of mvp2 and mvp5. The price, the delivery date, and the delivery address are thus of higher importance than legal clauses and the number of items.

At a low level of analysis, when threshold T=4, mvp3 and mvp4 are in the same class, while mvp1 is in another class. As the number of versions of mvp1 is higher than the number of versions of mvp3, one can conclude that the price is of very high importance, while the delivery date and address are only important. Having such a knowledge, negotiators can focus on the price negotiation which is the main issue in the current negotiation process. Such an analysis can be provided with various aspects of the negotiation process. Not only contract member importance may be analyzed but also, for instance, the involvement of negotiators or the importance of contract versions.

Hierarchical classifications allow analysis domains to be partitioned at various level of granularity by the threshold operation. The threshold provides the granularity of the obtained partition. The higher the threshold is, the lower the number of classes in the obtained partition is. In the context of negotiation analysis, this characteristics of the threshold operation is a key feature as it allows:

- various analysis levels; when the threshold is low, the generated partition consists of many classes, representing a fine-grained analysis. When the threshold is high, the generated partition consists of a few classes, representing a high-level analysis, giving an overview of the analyzed negotiation facet;
- fast focusing on details; starting from a high-level analysis, a negotiator can select a few

classes in the partition which are of special interest. These classes can further be analyzed in details by the application of a threshold operation with a lower threshold. The repetition of this technique allows to focus quickly on interesting details.

4. Conclusions

The multi-facet analysis approach to mass e-negotiations presented in this paper provides a solution to the problem of mass distributed negotiations via Internet, allowing a high number of geographically dispersed negotiators to work on real-life contracts.

Two ideas that are the basis of the multi-facet analysis approach to mass e-negotiation are: first, synthetic views of the negotiation process are needed in mass negotiation of complex contracts, because of the high amount of data, second, relationships between negotiators' proposals contain information that can be analyzed more easily than clauses contents, because attributes can be non-aggregable and their semantics is not always known.

An important feature of the multi-facet analysis approach to e-negotiation is its extensibility. Extensibility is an inherent requirement for the classification mechanism. New facets can easily be analyzed because of the use of ADL to extract and classify data. The multiversion contract model is also extensible because the structure of contracts is not fixed in the model. Therefore, advanced contract structures (e.g. tree structured contracts) may be built using the concepts of multiversion members and multiversion composite members proposed in the multiversion contract model.

The multi-facet analysis approach to e-negotiation opens new directions of research. An interesting example is application of the proposed approach to mobile computing, allowing mobile negotiators, which are potentially off-line, to analyze the negotiation process. The proposed multiversion contract model captures various important facets of the negotiation process (such as contract member sharing) in small size structures — association tables. These structures can be send efficiently over limited-bandwidth network and can be stored in memory-limited devices like mobile phones or PDAs. Negotiators could therefore analyze some aspects of the negotiation process without having to download the whole multiversion contract.

Another example is the use of software agents. Using the analysis mechanism, advanced behavior models can be build. Psychological and social models for negotiating agents may base on data retrieved from the analysis of various facets of the negotiation process. An agent may for example have a "collaborative" behavior, i.e. may look for negotiators having similar proposals to build a group of negotiators in order to increase its weigh in the negotiation process. The problem of ontologies would be minimized in this case, as the analyses "encapsulate" the meaning of various facets of the negotiation process.

5. References

[Andreoli 2001] Andreoli, J.M. and Castellani S. Towards a Flexible Middleware Negotiation Facility for Distributed Components. In *Proc. of 12th International Workshop on Database and Expert Systems Applications*, Munich, 2001, pp732-736.

[Cellary 1990] Cellary, W., Jomier G. Consistency of Versions in Object-Oriented Databases. In *Proc. of the 16th VLDB Conference*, Brisbane, Australia, August 1990.

[Cranefield 1999] Cranefield, S. and Purvis M. UML as an Ontology Modeling Language. In *Proc. of the IJCAI-99 Workshop on Intelligent Information Integration*, Stockholm, 1999.

[Duda 1973] Duda, R. and Hart P. *Pattern Classification and Scene Analysis*, New-York, Wiley, 1973.

[Fallside 2001] Fallside, D. C. XML Schema Part 0: Primer. W3C Recommendation, May 2 2001, http://www.w3.org/TR/xmlschema-0/

[Jennings 2001] Jennings, N. R., Faratin P., Lomuscio A. R., Parsons S., Sierra C., and Wooldridge M., Automated Negotiation: Prospects, Methods and Challenges. International Journal of Group Decision and Negotiation, 10(2), 2001.

[Lewicki 2001] Lewicki, R. J., Saunders D. M., and Minton J. W. *Essentials of Negotiation*, McGraw-Hill, 2001.

[Picard 2002] Picard, W. *Multi-facet Analysis of e-Negotiations*. PhD thesis, École Nationale Supérieure des Télécommunications, Paris, France, December 2002.

[Rumbaugh 1998] Rumbaugh, J., Jacobson I., and Booch G. *The Unified Modeling Language Reference Manual*, Addison-Wesley, 1998.

[Ströbel 2001] Ströbel, M. Communication Design for Electronic Negotiations on the Basis of XML Schema. In *Proc. of the 10th World Wide Web Conference*, Hong Kong, ACM Press New York, 2001, pp.9-20.