Why choosing *GeNCA* to develop your negotiation application?*

Marie-Hélène Verrons* Philippe Mathieu*

*Laboratoire d'Informatique Fondamentale Lille – CNRS UMR 8022, University of Lille
59 650 VILLENEUVE D'ASCQ, FRANCE
{verrons,mathieu}@lifl.fr

Abstract

E-negotiations are more and more popular as the use of Internet grows. The need for Negotiation Support Systems (NSS) is thus more and more obvious. Various NSS have been designed to help developers in building their negotiation applications, such as Inspire, the SilkRoad project, Magnet or the GNP platform. These NSSs are often dedicated to e-commerce, which is a subset of existing negotiation applications. That's why we have designed *GeNCA*, our *Generic Negotiation of Contracts API*, that aims to facilitate the design and implementation of contract-based negotiation applications, not only in the e-commerce field but also in non-commercial purposes. In this paper, we show that *GeNCA* is the best suited model for developing a negotiation application. To do so, we propose a set of criteria to classify the NSSs and compare them.

1 Introduction

Automated negotiation is more and more used to fasten and enhance results of traditional negotiations between humans. Various Negotiation Support Systems (NSS) have been designed to help developers in building their negotiation applications, such as Inspire [13] developed by the InterNeg Group at Carleton University, the SilkRoad project [20] of IBM, Magnet [10] developed by the university of Minnesota or the GNP platform [8] of the Montreal university. These NSSs are often dedicated to e-commerce, which is a subset of existing negotiation applications. That's why we have designed *GeNCA* [1], our *Generic Negotiation of Contracts API*, that aims to facilitate the design and implementation of contract-based negotiation applications, not only in the e-commerce field but also in non-commercial purposes.

GeNCA is based on a three-level architecture, that separates the communication part between agents, the negotiation part and the negotiation strategy part of an application. As a matter of fact, the way agents communicate doesn't play a role in the way negotiation

*These works are partially supported by the TAC program of the *contrat plan Etat-Région Nord-Pas de Calais*

is made, and different communication ways can be used in a same application executed on different environments.

It is also important to separate the decision making level from the two other levels, to allow a user to choose which negotiation strategy he will use without disturbing the remaining of the application. Moreover, the negotiation strategy is intrinsically linked to the negotiation application, and it is obvious that negotiating a ton of potatoes is not the same as negotiating a slot-time for an appointment, nor the same as negotiating the exclusive use of a shared resource for an hour.

The negotiation level of *GeNCA* contains a general negotiation protocol and a management of conflicting negotiations that allows to process them either sequentially or simultaneously. Parameters to specialise the protocol are set up in a file. Among these parameters, we can cite the number of agreements needed to confirm the contract, answer delay and default answer, number of rounds in the negotiation process, retraction possibility and number of renegotiations allowed.

In this article, we show that *GeNCA* [15] is well suited for developing a negotiation application in comparison with Inspire, SilkRoad, Magnet, GNP, works done at HP Laboratories [5] and Zeus [17] developed by British Telecommunications.

To compare these negotiation platforms, we introduce some criteria such as negotiation cardinality (the number of agents who negotiate), genericity of the platform (is it aimed for different kinds of negotiations ?), counter-offer possibility, retraction possibility, simultaneity of several negotiations (is it possible to negotiate at the same time different contracts ?), automatic renegotiation, possibility to set up parameters, default answer definition (which answer consider when the agents don't answer ?) and separation of the communication level (is it possible to set up how agents will communicate ?). We show that *GeNCA* is the only NSS that integrates each criterion defined here.

We also put forward the advantages of *GeNCA*, such as its genericity (*GeNCA* can be used for a variety of negotiation applications [16]), its three-level architecture, its conflicting negotiations management system (negotiations can be processed sequentially or in parallel), and the possibility to interact with human agents (a human user can negotiate when he's there and let its agent negotiate for him when he's gone for example for lunch).

The remainder of the paper is organized as follows: section 2 gives an overview of *GeNCA* and section 3 presents the NSSs that will be compared with *GeNCA*. Section 4 gives the criteria used to make the comparison presented in section 5. Section 6 concludes the paper.

2 *GeNCA* overview

The aim of our work is to conceive a general negotiation model (called *GeNCA*: *Generic Negotiation of Contracts API*), and to give an implementation of it. This allows a user wishing to develop a negotiation application not to have to do the whole job but to have a model that will facilitate his work.

Contracts involve resources that are either common to each agent (for example time-slots)

or individual to each agent (it is for example the case of goods in auctions). Each agent can initiate a negotiation over its own resources or common ones. Negotiation takes place between agents either to share resources or to allocate them. Several participants are involved in the negotiation and the resources can be shared between or given to a subset (perhaps empty) of these participants.

Our proposition is based on a three-level architecture, that separates the communication part between agents, the negotiation part and the negotiation strategy part of an application. As a matter of fact, the way agents communicate doesn't play a role in the way negotiation is made, and different communication ways can be used in a same application executed on different environments.

We equally show that it is important to separate the negotiation strategy from the two other levels, to allow a user to choose which negotiation strategy he will use without disturbing the remaining of the application. Moreover, the negotiation strategy is intrinsically linked to the negotiation application. In order to help the user in defining a strategy, we provide a priority list for resources and one for participants. Users can then order resources and participants according to their preferences.

2.1 *GeNCA* negotiation level

The negotiation level contains all the objects needed for agents to negotiate, and of course contains the negotiation protocol used in *GeNCA*. The protocol we propose here aims to define the messages that agents can send to each others with the operational dynamics associated. This negotiation protocol (Figure 1) is characterised by successive messages exchanged between an initiator (the agent who initiates the negotiation) and participants (the agents who participate to the negotiation) as in the Contract Net Protocol framework [19]. We first describe the phases that compose our negotiation protocol, and then we present the different kinds of applications that can be achieved with this protocol. The internal objects needed to the implementation of *GeNCA* are described in [14].

2.1.1 Negotiation protocol phases

We distinguish three phases for our negotiation process : the first one is the proposal phase which initiates the negotiation process. Then, there is an optional phase named conversation phase. This phase consists of rounds of proposals and counter-proposals in order to converge to an acceptable contract for everyone. Finally, there is the final decision phase where the contract is either confirmed, either cancelled.

Proposal phase In this phase, the initiator proposes a contract to a set of participants and waits for their answer. In response to the proposal, each participant answers if he agrees or rejects it.

Conversation phase This phase is necessary if there was not enough participants who agreed the contract proposal. A conversation is then started between the initiator and participants during which modification proposals are exchanged. Following these proposals, the

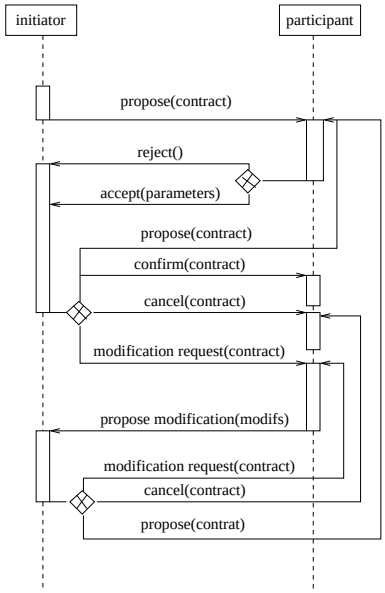

Figure 1: Negotiation protocol of *GeNCA*

initiator proposes a new contract to participants, and a new proposal phase is entered.

Final decision phase This final decision phase comes to either a confirmation or a cancellation of the contract. This decision is taken by the initiator in response to participants' answers.

2.1.2 Applications achievable with this protocol

In this subsection, we present the type of applications achievable with this protocol, as it is aimed to be general.

As we mentioned before, this protocol is inspired of the Contract-Net, and it adds an optional phase of conversation. As the protocol describes messages exchanged between agents but especially the order of messages and agents' turn to talk, and not what is the content of the message (for example, always a price), it allows many different applications to use it, which is not the case of many protocols such as the one used in ZEUS which is dedicated to marketplaces.

For example, you can use it in a "take it or leave it offer" form if you don't use the conversation phase. If you want to make auctions applications, you can implement English auctions as well as Dutch auctions. For English auctions, the initiator proposes his articles and participants answer giving a price as argument of the accept message if they are interested in the article, or rejecting the proposal otherwise. If no participant has proposed a satisfying price for the initiator, a conversation phase is entered where each modification consists of a new bid. The process finishes when a satisfying price has been proposed or when nobody rebids or the maximum number of turns predefined by the initiator has been reached.

For Dutch auctions, the initiator proposes an article with a high price, and if no participant accepts the proposal, the initiator proposes again the article with a lower price without asking for a modification from participants. The process finishes when a participant accepts the contract, or when the price reaches the minimum price wished by the initiator, or when the maximum number of rounds defined by the initiator is reached.

This protocol is not adapted to negotiations that have to be processed on several levels, for example, for negotiating to buy a car, you can first negotiate the colour, then the price and so on. This protocol is not adapted to combined negotiations [4], where contracts need to be linked. For example, you can't create two contracts and say both of them must be taken or none. If you want several resources from the same person, you put them in a single contract, but if you want several resources from several persons, you'll need one contract per person/resource but you can't specify that all contracts must be taken or none. Despite the protocol could fit it, negotiation with argumentation [18] is not included in *GeNCA*. The protocol could be adapted since the parameters of acceptance or modifications could be arguments.

2.2 *GeNCA* properties

The negotiation protocol enables contract-based negotiation between one initiator and several participants. Our implementation of this protocol in *GeNCA* allows several negotiations to take place simultaneously, thus finally negotiations take place between several initiators and several participants, that is to say many-to-many negotiation. The protocol also allows several rounds of counter-proposals, which makes it easier to find an agreement that meets the most preferences of each participant. Parameters to specialise the protocol are set up in a file. Among these parameters, we can cite the number of agreements needed to confirm the contract, answer delay and default answer, number of rounds in the negotiation process, retraction possibility and number of renegotiations allowed.

We propose a management of negotiations mechanism which allows to negotiate contracts on disjoint sets of resources in parallel and to negotiate contracts having conflicts on resources sequentially.

This model also allows to automatically renegotiate contracts that cannot be met any longer.

Our model has been implemented by a Java API also called *GeNCA* that has been used to achieve different applications [16].

The package we provide implements the whole negotiation level and gives default implementations for the interfaces of the communication and strategic levels.

Implementations of the communication level we give allows the use of the Magique [2] and Madkit [3] platforms, the use of threaded agents acting in a round-robin way and the use of e-mails.

Default strategies provided with the package are quite simple but can be easily refined. They take into account priorities given to resources and to persons in order to choose which contract to accept in case of conflict, and which resources to propose in case of modification request.

The package also provides a graphical interface for negotiation, which allows the user to create a contract, to visualise the messages sent and received by the agent, to answer a contract proposal if the manual mode is chosen, to visualise contracts taken by the agent, to have a view on the negotiations being conducted on resources and to retract a previously chosen contract.

In our package, the human user has two ways to use its agent. Manually, it is then a decision-helping tool which shows the state of all current negotiations, and, in this case, it is the user who answers a contract proposal. Automatically, this time, the agent is hidden and answers proposals by itself without human interventions.

Further informations can be found on *GeNCA*'s web site:

`http://www.lifl.fr/SMAC/projects/genca`

3 Other negotiation platforms

3.1 Inspire

Inspire [13] is developed by the InterNeg Group at Carleton University. It is aimed to facilitate multi-attributes negotiations with discrete values. To do so, a weight is given to each attribute against the others and the user gives his preferences over the different values by attribute. Then a ranking is done over the different offers containing couples (attribute, value) that the user can modify. Offers are made between 2 human users via e-mail or INSS server.

When an agreement is found, the system checks if there are other better solutions (in a Pareto-optimal sense) and indicates them to users that can then agree a new solution.

3.2 SilkRoad

The SilkRoad project [20] aims to facilitate the design and implementation of negotiation support systems for specific application domains. SilkRoad facilitates multi-attribute negotiations in e-business scenarios through a specific design methodology and a generic system architecture with reusable negotiation support components. A negotiation support system built on the basis of the SilkRoad architecture model acts as an intermediary between the actual negotiating agents (which might be software agents or humans) and thereby provides rule-driven communication and decision support. This project has common points with ours, like the possibility to have either software or human agents and the genericity of the system.

3.3 Magnet

Multi AGent NEgotiation Testbed [11] is a testbed for multi-agent negotiation, implemented as a generalised market architecture and developed at the university of Minnesota. It provides a support for a variety of types of transaction, from simple buying and selling of goods to complex multi-agent contract negotiation. A session mechanism enables a customer to issue a call-for-bids and conduct other business. The negotiation protocol for planning by contracting

consists of three phases : a call-for-bids, bidding and bid acceptance. In contrast, our protocol enables the initiator of the call-for-bids to make counter-proposals until an agreement is reached. In MAGNET, there is an explicit intermediary into the negotiation process and agents interact with each other through it, whereas all agents directly interact with each other in our negotiation process.

3.4 GNP

Morad Benyoussef et al. [8] want to create a Generic Negotiation Platform for auctions in marketplaces in a B2B context. This platform collects offers to sell and offers to buy and matches them. Users interact with GNP only via a web browser and it is possible to specify negotiation rules.

From a formal point of view, authors claim in [9] :

> *"we have identified a number of operations that are common to different negotiation processes. Some of these operations are :*
>
> - *define attributes and default values for the formalized concepts;*
> - *setup the end conditions for rounds, phases and the whole negotiation;*
> - *define the information to be displayed to or hidden from the players."*

As us, they have identified common points to different kinds of negotiation and they made use of it to propose a generic negotiation platform. They equally think that the negotiation process must be split from the other parts of the software and that rules governing negotiation must not be hard-coded. We are thus very close to these works on GNP, but our platform aims to be still more general in that we didn't restrict ourselves to auctions and marketplaces. Moreover, we propose different communication modes between agents whereas GNP is only used via a web browser. Our negotiation protocol is broader than the one of GNP as we renegotiate automatically contracts that have to be moved. A GNP advantage is that it proposes different negotiation templates, which makes it easier to instantiate predefined negotiation kinds.

3.5 A Generic Software Framework for Automated Negotiation

At HP Laboratories, Claudio Bartolini et al. [6, 7, 5] want to create a general framework for automated negotiation dedicated to market mechanisms. In this paper, they define two roles : participant and negotiation host. A participant is an agent who wants to reach an agreement, while the negotiation host is responsible for enforcing the protocol and rules of negotiation. Rules of negotiation include posting rule, visibility rule, termination rule ... It is the negotiation host who is responsible for making agreements. This framework proposes a general negotiation protocol parametrised with rules to implement a variety of negotiation mechanisms. It has common properties with our, like enabling one-to-one, one-to-many and many-to-many negotiations, or like parametrisation.

3.6 Zeus

ZEUS [17] is a generic Java API achieved by British Telecom in order to easily conceive cost-based negotiation applications between autonomous agents. Zeus proposes a negotiation protocol between two agents (an initiator and a participant) and on a single resource per contract. The protocol consists of a call-for-bids, and no mechanism of counter-proposal is provided. Moreover, it is possible to negotiate simultaneously different contracts on the same resource, that we don't allow. Another difference with our protocol is that retraction is not possible with Zeus. Once a contract is taken you can't retract yourself. Moreover, Zeus provides only cost-based strategies, and so is less generic than our protocol which is not dedicated to cost-based contracts. Although it is possible to add an interaction protocol in Zeus, it is a difficult thing to do, as says S. Thompson in the mailing list of Zeus in April 2002. On the other hand, *GeNCA* negotiation protocol is parametrisable via XML files, which simplifies modifications.

4 Evaluation criteria

In this section, we define the different criteria that we use to compare *GeNCA* with the other negotiation platforms. These criteria have been defined after studying different negotiation applications to characterize them.

4.1 Cardinality

Negotiation cardinality is an important feature for MAS. Its purpose is to know how many agents negotiate together. Different kinds of negotiation cardinality exist [12] , from one-to-one to many-to-many. Kasbah is an example of one-to-one negotiation : one buyer negotiates an article with one seller at a time. This form of negotiation is useful when only two persons are involved in the negotiation. But when a negotiation involves many participants with an initiator, it is a one-to-many negotiation. This is the case in English or Dutch auctions, for examples. Continuous double auctions are many-to-many negotiations. There are several buyers and several sellers at the same time trying to buy or sell articles.

4.2 Genericity

Genericity means that the framework allows you to process different kinds of negotiations, such as auctions as well as multi-attribute negotiations for example, and not only one specific application like Dutch auctions.

4.3 Retraction and Automatic renegotiation

Many times, during negotiations, some contracts can't be met any longer and have to be negotiated again. It is the case when appointments are negotiated. But you can't always question a contract that has been taken. For example in auctions, when an article is sold, it is

definitely sold, you can't retract yourself. That's why we define a parameter called retraction allowed, used to know whether it is possible or not to retract yourself from a contract previously taken.

If retraction is allowed, one may want to renegotiate automatically the contract for which retractions have been sent, without human intervention.

4.4 Default answer

When an agent doesn't answer, how should it be considered? What is the default answer for a participant? Several choices appear: either act as if he rejects the proposal, or as if he accepts it, or just remove him from the participants list. Acting as if he accepts the proposal might be surprising but it can be useful for example in appointment taking. Of course, in auctions it might not be a solution.

4.5 Simultaneity

The management of negotiations is an important criterion in a negotiation application. Negotiations can be processed sequentially, in parallel or in a mixed way, depending on the constraints of the application. In the case of parallel negotiations, no restriction is made on the resources, they can already being negotiated for another contract. For sequential negotiations, each one is processed one after another. The mixed solution is to process simultaneously only the negotiations which involve disjoint sets of resources. The other negotiations will wait for their turn.

4.6 Counter offers

When the first proposal isn't accepted by the participants, can they give their point of view, make counter-offers, or not? For example, in the Contract-Net Protocol [19], it is not possible for contractors to make counter-offers.

This criterion also indicates if the initiator tries to come to an agreement alone or if he takes into account participants' desires.

4.7 Parameters set up

This criterion indicates if it is possible to fix some parameters of the negotiation or not. For example, it can be useful to easily change the retraction possibility, the way negotiations are processed (simultaneity), etc.

4.8 Separation of the communication level

The communication level defines the way agents use to communicate with each other. It can be by sending e-mails or using sockets or else using communication primitives given by the MAS they're living in. Separation of the communication level means that the framework can be used by several of these kinds of communication ways. This criterion is thus used to know if it is possible to use the framework in any context or if it is linked to a particular use, like a web application.

5 Comparison

In this section, we compare the different NSSs presented in Section 3 with *GeNCA* according to the criteria we presented in the previous section.

5.1 Cardinality

GeNCA and HP's framework are the only ones to enable many-to-many negotiation. Magnet then proposes one-to-many negotiations and the other platforms only propose one-to-one negotiations.

The advantage provided by many-to-many negotiation is that it enables one-to-many and one-to-one negotiation. *GeNCA* and HP's are thus the most general platforms in terms of negotiation cardinality. If more than two persons are to negotiate together, only Magnet, HP's and *GeNCA* can be used to develop the application. No other platform will fit it. This greatly reduces the possible choices for a developer.

5.2 Genericity

Magnet, HP's platform and *GeNCA* are the only generic platforms as they propose different kinds of negotiation and not only CNP for Zeus, auctions for GNP and multi-attribute negotiation for Inspire and SilkRoad. For example, it is possible with *GeNCA* to achieve an appointment taking application, auctions, a negotiation game or a timetable creation application [16].

This is one of the most important criteria when someone has to choose a platform for developing a negotiation application. It shows which platforms can support (almost) any kind of negotiation. So if someone has to develop many negotiation applications of different types, she needs to know how to use only one platform to develop them all. Otherwise, she'll need to study one specific platform for each application and so spend more time for achieving her job.

5.3 Retraction and Automatic renegotiation

As HP's platform and GNP are dedicated to auctions and Zeus contains only the Contract-Net Protocol, retraction and obviously automatic renegotiation are not proposed by them. Magnet and Inspire propose retraction but not automatic renegotiation. Nothing enables us to say whether SilkRoad proposes them or not. *GeNCA* is the only platform to propose both of these criteria.

So if your application not only needs to offer retraction possibility but also automatic renegotiation, you'll have to choose *GeNCA* to develop it.

5.4 Default answer

In our approach, the default answer that will be taken into account if a participant doesn't answer to a proposal can be defined by the user (either an acceptance or a rejection of the proposal). GNP, Magnet and SilkRoad consider a rejection and Inspire, Zeus and HP's platform don't take it into account.

So if your application needs to be able to specify whether a lack of answer is considered as a rejection or as an acceptance, the only platform that enables you to do that is *GeNCA*.

5.5 Simultaneity

All platforms enable to process all negotiations in parallel and for some of them it is not possible to take into account conflicts between contracts. *GeNCA* proposes to negotiate sequentially contracts that have not disjoint sets of resources, the user opts for the management he prefers.

So if conflicts over resources are not a problem for your application, any of these platforms fits. But if you are to negotiate critical resources for which you want to be sure that negotiations will be processed sequentially, *GeNCA* is the best suited platform.

5.6 Counter-offers

Only Zeus and Magnet don't let the possibility for making counter-offers. This criterion is largely embedded in the platforms, so it can't make the difference between them. This equally shows that it is an important feature in negotiation platforms, and those which don't integrate it are obsolete.

5.7 Parameters set up

GNP, HP's, Inspire and *GeNCA* are the only platforms that allow to set up parameters for negotiation. That is to say that only these platforms allow you to customize them for your application. It is thus possible to specialize the platform in order to make it ideal for your needs.

5.8 Separation of the communication level

Only *GeNCA* is not provided with a single communication way for users. So you can use it whichever communication way your agents use. That is to say that *GeNCA* is conceived so that you can plug it into an existing application that doesn't provide negotiation by specifying how your entities communicate with each other.

5.9 Overall comparison

Criteria	*GeNCA*	Inspire	SilkRoad	Magnet	GNP	HP	Zeus
cardinality	$n \to m$	$1 \to 1$	$1 \to 1$	$1 \to m$	$1 \to 1$	$n \to m$	$1 \to 1$
genericity	yes	no	no	yes	no	yes	no
retraction	yes	yes	?	yes	no	no	no
automatic renegotiation	yes	no	?	no	no	no	no
default answer	chosen	no	rejection	rejection	rejection	no	no
simultaneity	yes	yes	yes	yes	yes	yes	yes
counter-offers	yes	yes	yes	no	yes	yes	no
parameters set up	yes	yes	no	no	yes	yes	no
separation of communication level	yes	no	no	no	no	no	no

Table 1: Comparative table of some negotiation platforms according different criteria. ? is used when no information allow us to confirm or infirm the criterion.

Table 1 sums up the value of each criterion for the different platforms quoted above and so provides a global view on the features of each platform.

This table shows that *GeNCA* is the only platform that enables you to choose your communication way and the default answer and to renegotiate automatically your contracts. No other platform give you these possibilities. Moreover, *GeNCA* possesses all other criteria, making it the best alternative if you have to develop a negotiation application.

6 Conclusion

In this article, we showed that *GeNCA* is a real alternative for developing negotiation applications. We first described *GeNCA*, our *Generic Negotiation of Contracts API*, that aims to facilitate the design and implementation of contract-based negotiation applications, not only in the e-commerce field but also in non-commercial purposes. *GeNCA* is based on a three-level architecture, that separates the communication part between agents, the negotiation part and the negotiation strategy part of an application. This makes *GeNCA* truly generic. *GeNCA* allows many-to-many negotiations, retraction, automatic renegotiation, counter-offers, its protocol can be specialised by parameters and conflicting negotiations can be processed either in parallel or sequentially. Moreover, you can specify how agents communicate with each other, which makes it possible to use *GeNCA* in an already existing application.

Then, we presented six other negotiation platforms: Inspire, SilkRoad, Magnet, GNP, HP's platform and Zeus. In order to compare these platforms with *GeNCA*, we described several criteria that we think are important to characterise a negotiation platform. These criteria are cardinality, genericity, retraction, parameters set up, automatic renegotiation, default answer, simultaneity, counter-offers and separation of the communication level. With these criteria, we showed that *GeNCA* is the best alternative to develop a negotiation application as it is the only one which has all of these criteria. As a matter of fact, *GeNCA* is the only platform to propose automatic renegotiation, to choose the default answer when someone doesn't answer and to choose the way agents communicate. Moreover, it is possible with *GeNCA* to negotiate sequentially critical resources whereas with other platforms all resources are negotiated in parallel.

References

[1] http://www.lifl.fr/SMAC/projects/genca.

[2] http://www.lifl.fr/SMAC/projects/magique.

[3] http://www.madkit.org.

[4] Samir Aknine. New Multi-Agent Protocols for M-N-P Negotiations in Electronic Commerce. In *National Conference on Artificial Intelligence, AAAI, Agent-Based Technologies for B2B Workshop*, Edmonton, Canada, July 2002.

[5] C. Bartolini, C. Preist, and N. R. Jennings. Architecting for reuse: A software framework for automated negotiation. In *Proc. 3rd Int Workshop on Agent-Oriented Software Engineering*, pages 87–98, Bologna, Italy, 2002.

[6] Claudio Bartolini and Chris Preist. A Framework for Automated Negotiation. Technical Report HPL-2001-90, HP Laboratories Bristol, 2001.

[7] Claudio Bartolini, Chris Preist, and Nicholas R. Jennings. A generic software framework for automated negotiation. Technical Report HPL-2002-2, HP Laboratories Bristol, 2002.

[8] Morad Benyoucef, Rudolf K. Keller, Sophie Lamouroux, Jacques Robert, and Vincent Trussart. Towards a Generic E-Negotiation Platform. In *Proceedings of the Sixth International Conference on Re-Technologies for Information Systems*, pages 95–109, Zurich, Switzerland, 2000.

[9] Morad Benyoucef, Rudolf K. Keller, Sophie Lamouroux, Jacques Robert, and Vincent Trussart. Towards a Generic E-Negotiation Platform. In *Proceedings of the Sixth International Conference on Re-Technologies for Information Systems*, pages 95–109, Zurich, Switzerland, 2000.

[10] J. Collins, B. Youngdahl, S. Jamison, B. Mobasher, and M. Gini. A Market Architecture for Multi-Agent Contracting. In *2nd Int'l Conf on Autonomous Agents*, pages 285–292, Minneapolis, May 1998.

[11] John Collins, Maksim Tsvetovatyy, Bamshad Mobasher, and Maria Gini. MAGNET : A Multi-Agent Cntracting System for Plan Execution. In *Workshop on Artificial Intelligence and Manufacturing: State of the Art and State of Practice*, pages 63–68, Albuquerque, NM, August 1998. AAAI Press.

[12] Robert H. Guttman and Pattie Maes. Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce. In *Proceedings of the Second International Workshop on Cooperative Information Agents (CIA'98)*, Paris, France, July 1998.

[13] Gregory E. Kersten and Sunil J. Noronha. Supporting International Negotiation with a WWW-Based System. Technical Report IR-97-49, International Institute for Applied Systems Analysis (IIASA), August 1997.

[14] Philippe Mathieu and Marie-Hélène Verrons. A generic model for contract negotiation. In *Proceedings of the AISB'02 Convention*, London, UK, 2002.

[15] Philippe Mathieu and Marie-Hélène Verrons. A Generic Negotiation Model for MAS using XML. In *Proceedings of the ABA workshop Agent-based Systems for Autonomous Processing, held by the IEEE International Conference on Systems, Man and Cybernetics.*, Washington, USA, October, 5-8 2003. IEEE Press.

[16] Philippe Mathieu and Marie-Hélène Verrons. Three different kinds of negotiation applications achieved with GeNCA. In *Proceedings of the International Conference on Advances in Intelligent Systems - Theory and Applications (AISTA) In cooperation with the IEEE Computer Society*, Centre de Recherche Public Henri Tudor, Luxembourg-Kirchberg, Luxembourg, 15-18 novembre 2004.

[17] H.S. Nwana, L.C. Lee D.T. Ndumu, and J.C. Collis. ZEUS : A Toolkit for Building Distributed Multi-Agent Systems.

[18] S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[19] R. G. Smith. The Contract Net Protocol : high-level communication and control in a distributed problem solver. *IEEE Transactions on computers*, C-29(12):1104–1113, December 1980.

[20] M. Ströbel. Design of Roles and Protocols for Electronic Negotiations. *Electronic Commerce Research, Special Issue on Market Design*, 1(3):335–353, 2001.