DISTRIBUTED MULTI-AGENT SYSTEMS FOR SUPPORTING COLLABORATIVE NETWORKED ENVIRONMENT

Adina-Georgeta CRETAN*

Abstract

This paper proposes a distributed multi-agent system to model and support parallel and concurrent negotiations among organizations acting in the same industrial market. The complexity of the approach is given by the dynamic environment in which multi-attribute and multi-participant negotiations take place. The metaphor Interaction Abstract Machines (IAMs) is used to model the parallelism and the non-deterministic aspects of the negotiation processes that occur in Collaborative Networked Environment.

Keywords: *Automated negotiation, web services, collaborative networked environment, computing platform, multi-agent systems.*

1. Introduction

The advent of the Internet and more recently the cloud-computing trend have led to the development of various forms of virtual collaboration in which the organizations are trying to exploit the facilities of the network to achieve higher utilization of their resources. We try to provide support to these collaboration activities and we propose negotiation as a fundamental mechanism for such collaborations.

The concept of "Virtual Enterprise (VE)" or "Network of Enterprises" has emerged to identify the situation when several independent companies decided to collaborate and establish a virtual organization with the goal of increasing their profits. Camarinha-Matos defines the concept of VE as follows: "A *Virtual Enterprise (VE)* is a temporary alliance of enterprises that come together to share skills and resources in order to better respond to business opportunities and whose cooperation is supported by computer networks"[1].

In this paper we present how organizations participate and control the status of the negotiations and how the negotiation processes are managed.

The starting point in the development of this work was the goal to support small and medium enterprises that are not able or are not willing to perform alone a large contract since in this situation the association in a virtual alliance provides the opportunity to subcontract the tasks of the contract to other partners within the alliance. To achieve this goal, research was dedicated to the development of a model to coordinate the negotiations that take place within an inter-organizational alliance. Our research was focused on the topics of virtual alliances, automation of the negotiations and of coordination aimed to provide the mechanisms for coordinating the negotiations that take place among autonomous enterprises that are grouped in a virtual alliance.

Assuming that the nature of the roles that may be played in a negotiation are similar in multiple approaches, the number of participants involved at the same time in the same negotiation is considerably different.

Depending on the number of participants involved in a negotiation, we may distinguish various negotiation types: *bilateral negotiation (one-to-one)*; *one-to-many negotiation; many-to-many negotiation*.

Taking into account the complexity of the negotiations modeled by multi-agent system, we can state that to conduct in an efficient fashion one or many negotiations that involve a large number of participants and to properly account for all negotiation dimensions, it is necessary to develop a coordination process that is defined outside of the specific constraints of a given decision mechanism or communication protocol.

The negotiation process was exemplified by scenarios tight together by a virtual alliance of the autonomous gas stations. Typically, these are competing companies. However, to satisfy the demands that go beyond the vicinity of a single gas station and to better accommodate the market requirements, they must enter in an alliance and must cooperate to achieve common tasks. The type of alliance that we use to define their association emphasizes that each participant to this alliance is completely autonomous i.e., it is responsible of its own amount of work and the management of its resources. The manager of a gas station wants to have a complete decision-making power over the administration of his contracts, resources, budget and clients. At the same time, the manager attempts to cooperate with other gas stations to accomplish the global task at hand only through a minimal exchange of information. This exchange is

* Associate Professor, PhD, Faculty of Economics and Business Administration"Nicolae Titulescu" University, Bucharest (e-mail: adinacretan@univnt.ro)

[1] Camarinha-Matos L.M. and Afsarmanesh H.,(2004), *Collaborative Networked Organizations,* Kluwer Academic Publisher Boston

minimal in the sense that the manager is in charge and has the ability to select the information exchanged.

When a purchasing request reaches a gas station, the manager analyses it to understand if it can be accepted, taking into account job schedules and resources availability. If the manager accepts the purchasing request, he may decide to perform the job locally or to partially subcontract it, given the gas station resource availability and technical capabilities. If the manager decides to subcontract a job, he starts a negotiation within the collaborative infrastructure with selected participants. ***In case*** that the negotiation results in an agreement, a contract is settled between the subcontractor and the contractor gas station, which defines the business process ***outsourcing*** jobs and a set of obligation relations among participants[2].

The gas station alliance scenario shows a typical example of the SME virtual alliances where partner organizations may be in competition with each other, but may want to cooperate in order to be globally more responsive to market demand.

The collaborative infrastructure, that we describe, should flexibly support negotiation processes respecting the autonomy of the partners.

We are starting with a presentation in Section 2 of a formal interaction model to manage multiple concurrent negotiations by using the metaphor Interaction Abstract Machines (IAMs). Section 3 presents an example of the proposed interaction model using the IAMs metaphor. Then, we are briefly describing in Section 4 the collaborative negotiation architecture. Finally, Section 5 concludes this paper.

2. Building the Negotiation Model

In this section we propose a formal model to settle and to manage the coordination rules of one or more negotiations, which can take place in parallel. We will introduce the metaphor of Interaction Abstract Machines (IAMs) to describe the negotiation model. We introduce the Program Formula to define the methods used to manage the parallel evolution of multiple negotiations.

2.1. The Metaphor Interaction Abstract Machines (IAMs)

The metaphor Interaction Abstract Machines (IAMs) will be used to facilitate modeling of the evolution of a *multi-attribute, multi-participant, multi-phase negotiation*. In IAMs, a system consists of different *entities* and each entity is characterized by a state that is represented as a set of *resources* [4]. It may evolve according to different laws of the following form, also called "*methods*":

A1@…@An <>- B1@…@Bm

A method is executed if the state of the entity contains all resources from the left side (called the "*head*") and, in this case, the entity may perform a transition to a new state where the old resources *(A1,…,An)* are replaced by the resources *(B1,…,Bm)* on the right side (called the "*body*"). All other resources of the entity that do not participate in the execution of the method are present in the new state.

The operators used in a method are:

- the operator @ assembles together resources that are present in the same state of an entity;
- the operator <>- indicates the transition to a new state of an entity;
- the operator & is used in the body of a method to connect several sets of resources;
- the symbol "T" is used to indicate an empty body.
- In IAMs, an entity has the following characteristics:
- if there are two methods whose heads consist of two sets of distinct resources, then the methods may be executed in parallel;
- if two methods share common resources, then a single method may be executed and the selection procedure is made in a non-deterministic manner.

In IAMs, the methods may model four types of transition that may occur to an entity: *transformation*, *cloning*, *destruction* and *communication*. Through the methods of type *transformation* the state of an entity is simply transformed in a new state. If the state of the entity contains all the resources of the head of a transformation method, the entity performs a transition to a new state where the head resources are replaced by the body resources of the method. Through the methods of type *cloning* an entity is cloned in a finite number of entities that have the same state. If the state of the entity contains all the resources of a head of a cloning method and if the body of the method contains several sets of distinct resources, then the entity is cloned several times, as determined by the number of distinct sets, and each of the resulting clones suffers a transformation by replacing the head of the method with the corresponding body. In the case of a *destruction* of the state, the entity disappears. If the state of the entity contains all the resources of the head of a transformation method and, if the body of the method is the resource T, then the entity disappears.

In IAMs, the *communication* among various entities is of type broadcasting and it is represented by the symbol "^". This symbol is used to the heads of the methods to predefine the resources involved in the broadcasting. These resources are inserted in the current entity and broadcasted to all the entities existent in the system, with the exception of the current entity. This mechanism of communication thus executes two synchronous operations:

- *transformation*: if all resources that are not pre-defined at the head of the method enter in collision, then the pre-defined resources are inserted in the entity and are immediately consumed through the application of

[2] Singh M.P., (1997) *Commitments among autonomous agents in information-rich environments*. In Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW), pp. 141–155

the method;

- *communication:* insertion of the copies of the pre-defined resources in all entities that are present in the system at that time instance.

2.2. Modelling the Negotiation Process

According to our approach regarding the negotiation, the participants to a negotiation may *propose* offers and each participant may decide in an autonomous manner to stop a negotiation either by *accepting* or by *rejecting* the offer received. Also, depending on its role in a negotiation, a participant may *invite* new participants to the negotiation. To model this type of negotiation, we will make use of the previously defined particles and we will propose the methods to manage the evolution of these particles.

As we have seen, a characteristic of negotiation is its multi-node image, which allows parallel development of several phases of negotiation. A possibility to continue a negotiation is to create a new phase of negotiation from an existing one. In this regard, the Figure 1 presents the possible evolutions of a ph0 phase of negotiation described by the *atom (s,ph0)*.

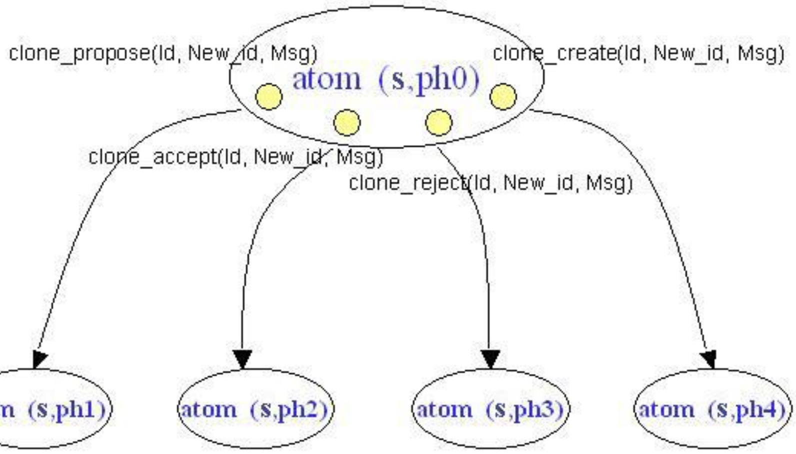


Fig. 1. Evolution of negotiation process by cloning an atom

In accordance with the aspects of negotiation for which changes are made, three new negotiation phases are possible:

- evolution of negotiated <u>attributes</u> and / or of their value from *atom(s,ph0)* to *atom(s,ph1)*: a participant sends a new proposal thus achieving either the *contraction* of the negotiation attributes, or their *extension*, by the introduction of new attributes to negotiate;
- evolution of the negotiation <u>status</u> perceived by one of the sequences sharing the new negotiation phase: one of the participants accepts - *atom(s,ph2)* - or refuses a proposal - *atom(s,ph3);*
- evolution of <u>participants</u> and of <u>dependences</u> among negotiations by the evolution of the number of sequences sharing the same negotiation phase: a sequence can invite a new sequence to share a new phase of negotiation *atom(s,ph4)*.

Through the use of the metaphor IAMs, the evolutions of the negotiation phases correspond to the evolutions at the atoms level. The evolution may be regarded as a process consisting of two stages: a *cloning* operation of the atom existent in the initial stage and a *transformation* operation within the cloned atom to allow for the new negotiation phase.

The *cloning* operation is expressed by a set of methods involving the particles *event* and these methods are used to facilitate the evolution of the negotiation.

We propose the following methods associated to the particles *event* to model the cloning of an atom where new message particles are introduced:

- The method *Propose* is associated to the particle event *clone_propose(Id, New_id, Msg)* and models the introduction of a new proposal (*clone_propose*), made by one of the participants to the negotiation.

This method is expressed:

name(Id) @ enable @ clone_propose(Id, New_ id, Msg)<>- (enable @ name(Id)) & (freeze @ name(New_ id) @ propose(Rname, Content))

The atom identified by the particle *name(Id)* is cloned. The new proposal contained in the particle *propose(Rname, Content)* will be introduced in the new atom *name(New_id)*.

- The method *Accept* is associated to the event particle *clone_accept(Id, New_id, Msg)* and models the case when one of the participants has sent a message of acceptance of an older proposal (*clone_accept*).

This method is expressed:

name(Id) @ enable @ clone_accept(Id, New_ Id, Msg) <>- (enable @ name(Id)) & (freeze @ name(New_ Id) @ accept(Rname))

The atom identified by the *name(Id)* is cloned. The acceptance message contained in the particle *accept(Rname)* will be introduced in the new atom *name (New_id)*.

- The method *Reject* is associated to the event particle *clone_reject(Id, New_id, Msg)* and models the denial of an older proposal (*clone_reject*) made by one of the participants.

This method is expressed:

name(Id) @ enable @ clone_reject(Id, New_ Id, Msg) <>- (enable @ name(Id)) & (freeze @ name(New_ Id) @ reject(Rname))

The atom identified by the particle *name(Id)* is cloned. The refusal message contained in the particle *reject(Rname)* will be introduced in the new atom *name(New_ id).*

- The method *Create* is associated to the event particle *clone_create(Id, New_id, Msg).* This method models the invitation of a new sequence (*clone_create*) made by one of the participants for sharing the newly created negotiation phase.

This method is expressed:

name(Id) @ enable @ clone_create(Id, New_Id, Msg) @ <>- (enable @name(Id)) & (freeze @ name(New_Id) @ create(Rname, Type))

The atom identified by the particle *name(Id)* is cloned, and a particle *create(Rname, Type)* is introduced in the new atom *name(New_ id)* that will further generate the occurrence of a new representation particle for the new sequence participating in the negotiation.

These methods are described in a generic way. Thus, new particles may be added depending on how the current sequence builds negotiation graphs.

By these methods of the event particles, the duplication of an atom has been modeled, in which new message particles are introduced (Figure 2). In the new atom, the representation particles for the current negotiation phase remain identical with those of the first atom.

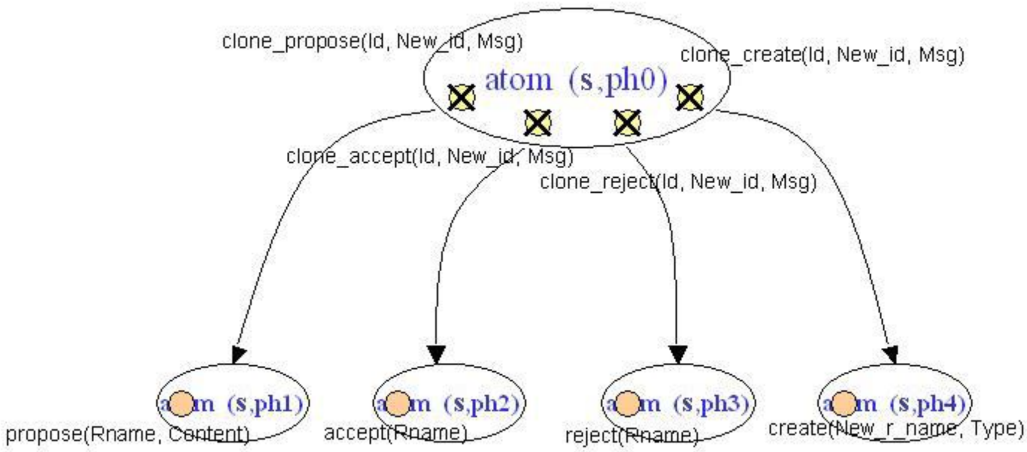


Fig. 2. Evolution of negotiation process by transformation of an atom state

According to our approach, the evolution of the negotiation process takes place by changing or creating a new negotiation phase. This phase can evolve according to: *i)* status; *ii)* attributes negotiated; *iii)* number of sequences participant in the negotiation, as in the following:

- The sequence of the statutes is the following: *i)* the sequence *s* finds itself in the *initiated* state at the creation of a first atom and of a first phase of negotiation; *ii)* the sequence *s* switches to the *undefined* state at the moment of emission or reception of a message; *iii)* if a participant accepts or declines a proposal, the *s* associated sequence may pass to a *success* or *failure* statute.
- Referring to the negotiated attributes (*Issues*), the different messages contribute to the evolution of the multitude of attributes and their values.
- The introduction of a new sequence in the current negotiation is modeled by inserting a new particle of representation on the current negotiation phase, which models the instant image of a new sequence on the current phase of negotiation.

In order to model these evolutions at the level of a negotiation phase, the message-type particles described above have been defined. The *message* particles participate in the transformation methods, which change the negotiation phase of an atom by replacing the representation particles of the negotiation sequences involved in the creation or in the reception of the exchanged messages.

In the following, we propose the basic forms of the *transformation* methods. Depending on the particular constraints of the negotiation, other transformation methods and other particles can be defined for modeling the foreseen constraints.

- The *transformation* method associated to a *propose(Rname, Content)* particle contributes to the local evolution of a negotiation phase regarding the status and the attributes negotiated. This evolution takes place by replacing, in the existing atom, all representation particles that are involved (depending on the method) with the new particles that have the status changed to *undefined*. Further, the set of the negotiated attributes (*Issues*) contains the new proposal expressed in the *Content* of the message particle.

freeze @ localr(Rname1, S1, I1) @ extr(Rname, S2, I1) @ propose(Rname, Content)<>- enable@ localr(Rname1, undefined, I) @ extr(Rname, undefined, I)

The atom changes the state by consuming the *propose()* particle as well as two representation particles to create new representation particles that describe the new proposal received.

- The transformation method associated to an *accept(Rname)* particle leads to the local evolution of a negotiation phase in terms of status. Evolution is achieved by replacing, within the corresponding atom, the representation particles involved, with the new particles whose status has been changed from *initiated* or *undefined* into *success* :

freeze @ localr(Rname1, S1, I1) @ extr(Rname, S2, I1) @ accept(Rname) <>- localr(Rname1, success, I1) @ extr(Rname, success, I1)

The atom changes the state by consuming the *accept()* particle and two representation particles to create the new representation particles whose status is changed in *success*.

- The transformation method associated to a *reject(Rname)* particle. This method is similar to the *accept(Rname)* particle, except for the fact that the evolution of the negotiation phase is achieved by changing the status of the representation particles concerned from *initiated* or *undefined* into *failure* :

freeze @ localr(Rname1, S1, I1) @ extr(Rname, S2, I1) @ reject(Rname) <>- localr(Rname1, fail, I1) @ extr(Rname, fail, I1)

The atom changes the state by consuming the *accept()* particle and two representation particles to create the new representation particles which have changed the status into *failure*.

- The transformation method associated to a *create(New_r_name, Type)* particle contributes to the evolution of a negotiation phase in terms of number of sequences that participate to this negotiation phase. This evolution is achieved by introducing, in the corresponding atom, a new representation particle:

freeze @ create(Rname, type) <>- extr(Rname, init, ∅) @ enable

As this sequence is just invited in the negotiation, its status is *initiated* and its set of the negotiated attributes is the empty set.

Thus, the negotiation phases and the evolution of these phases have been described using representation particles, event particles and message particles.

Given the fact that IAMs metaphor achieves a non-deterministic execution of the methods, we have introduced the c*ontrol* particles (see section 3.3) in order to counter this disadvantage and achieve a coherent execution of a negotiation process.

The evolution of all negotiation atoms and the negotiation phases take place in parallel.

To model the coordination of the execution of the negotiation process within a sequence, we used the communication mechanism among the existing negotiations. This type of particles that are part of the communication process among different negotiation atoms communicate to all negotiation atoms a certain result.

In the negotiation processes, the messages hold meta-information regarding the content of the messages that describe the proposals in terms of the value of different attributes of the negotiation object. We assume that all the negotiation participants use the same language and ontology.

Next section presents an example of modeling a negotiation composed of a set of negotiation sequences.

3. Example - modeling the negotiation process using the IAMs metaphor

In this example, a simple negotiation scenario will be presented, whose negotiation process corresponds to an exchange of proposals leading to an agreement.

In the proposed scenario, we consider a carpentry workshop of the p1 participant. The participant p1 decides to outsource a job (the assembling for 10K LM at a cost less than 2€/LM, within less than 5 days) to another carpentry workshop of a p2 participant.

The negotiation N that occurs between p1 and p2 is a bilateral negotiation.

It is described by two sequences: N(t) = $\{s_1, s_2\}$ with $s_1 \in$sequences(t,p_1), $s_2 \in$sequences(t,p2) and role(t,p1,N)=initiator, role(t,p2,N)=guest.

The scenario modeled subsequently takes place in three distinct steps:

- **step1**: after a first proposal made by participant p1 to participant p2, the participant p2 decides to send a new proposal;
- **step2**: the participants decide to agree on the second proposal;
- **step3**: the agreement is established and the negotiation stops.

Modeling takes place as it follows:

Step 1

Figure 3.a) shows the s_1 , *view*(s_1)= (p_1,N,R_1) negotiation sequence with an a1 atom corresponding to a negotiation phase described by two representative particles, a local one and an external one, and two control particles, (*enable* and *name(a1)*). This atom can be considered as being the proposal made by p1 to p2 at the beginning of the negotiation.

Going on with the scenario, there is assumed that within the *a1* atom of the s_1 negotiation sequence, the *clone_propose(a1, a2, cost=18K delay=3)* event was introduced in order to announce a new proposal.

By using the *Propose* method, the s_1 negotiation sequence will contain two atoms (see Figure 3.b)): a1 atom that has changed by consuming the *clone_propose* particle and an a2 new atom, which is the a1 clone (representation particles not involved in the method remain unchanged in the two atoms). The expression of this method is:

name(Id) @ enable @ clone_propose(Id, New_ id, Msg)<>- (enable @ name(Id)) & (freeze @ name(New_ id) @ propose(Rname, Content))

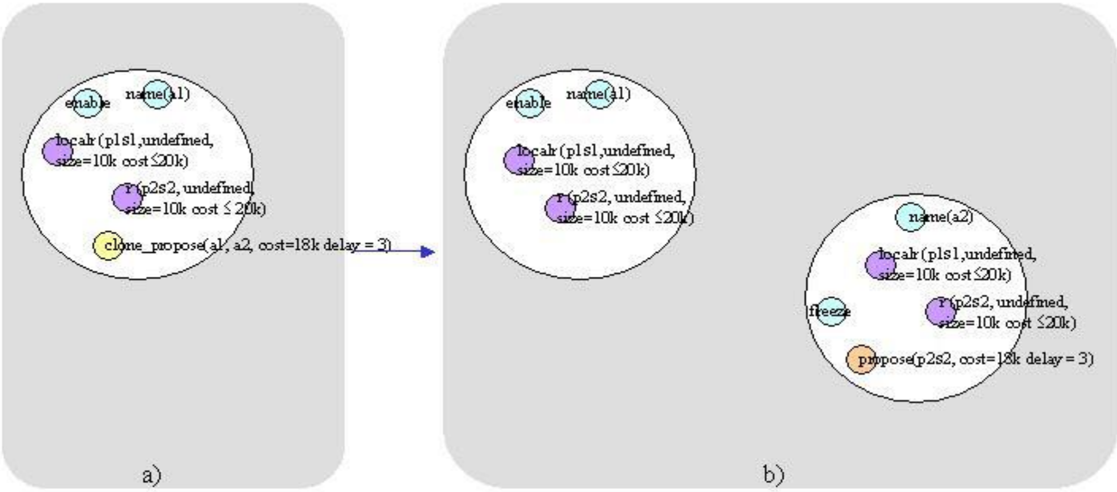


Fig. 3. a) – Proposal of p1 to p2 contained in a1atom, b) – Transformation of a1 atom in a2

From now on, the negotiation is described through the two negotiation atoms, in which the negotiation process can evolve independently.

The following methods will help us to model the fact that the (*name*, *freeze*, *propose*) new particles introduced in the a2 atom will make that the negotiation phase attached to this atom have a different evolution in comparison to that of a1 atom.

For the *propose* particle, the following method will be used:

freeze @ localr(Rname1, St1, I1) @ extr(Rname2, St2, I1) @ propose(Rname2, Content)<>- enable@ localr(Rname1, undefined, I) @ extr(Rname2, undefined, I)

This method exchanges the representation particles, which preserve the old values of the attributes, with the new representation particles that describe the new proposals received.

Thus, to move from an *extr(p2s2, undefined, size=10K cost≤20K)* representation particle, and from a *propose(p2s2, cost=18K delay=3)* message particle to an *extr(p2s2, undefined, size=10K cost=18K delay=3)* representation particle, the existence of a computational-type particle called *construct(I1,Content,I)* has been supposed, which calculates this transformation (see Figure 3.c)).

The expression of this method is:

freeze @ localr(Rname1, St1, I1) @ extr(Rname2, St2, I1) @ propose(Rname2, Content) @ {construct(I1,Content,I)} <>- enable @ localr(Rname1, undefined, I) @ extr(Rname2, undefined, I)

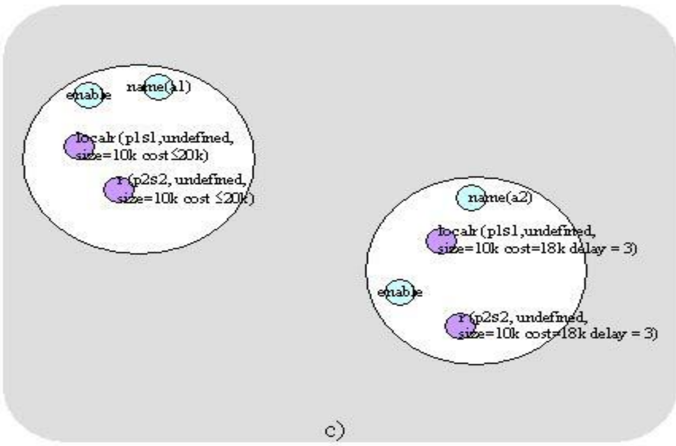


Fig. 3. c) Evolution of a1 and a2 atoms

Step 2

Further, we assume that p1 participant examines the two proposals and decides to accept the second proposal. Acceptance is modeled as follows:

i) the atom containing the proposals to be accepted is duplicated by *Accept* method associated to an event *clone_accept()* particle;

name(Id) @ enable @ clone_accept(Id, New_ Id, Msg) <>- (enable @ name(Id)) & (freeze @ name(New_ Id) @ accept(Rname))

ii) the clone atom evolves by consuming the *accept()* message particle;

freeze @ localr(Rname1, St1, I1) @ extr(Rname, St2, I1) @ accept(Rname) <>- localr(Rname1, success, I1) @ extr(Rname, success, I1)

Thus, in Figure 3.d) the s1 negotiation sequence of a p1 participant is composed of three negotiation phases. The third a3 atom contains a negotiation phase in which *success* is the negotiation status.

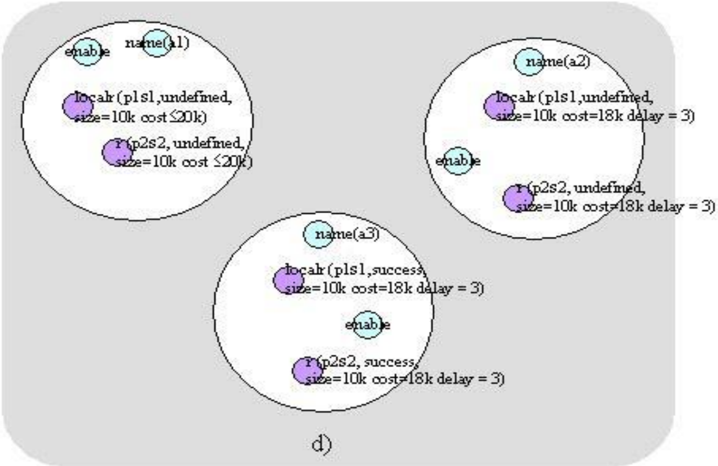


Fig. 3.d) Negotiation phases of the s1 sequence of a p1 participant

Step 3

The a3 atom can thus be perceived as the image of a negotiation phase on which the two participants have agreed. The existence of this agreement has to imply the fact that the negotiation has to come to an end.

In Figure 3.e), a3 atom changes its state and, at the same time, communicates to the other atoms to stop. The expression of this method is:

localr(Rname1, success, I1) @ extr(Rname2, success, I1) @ ^stop(I1) <>- ready(I1)

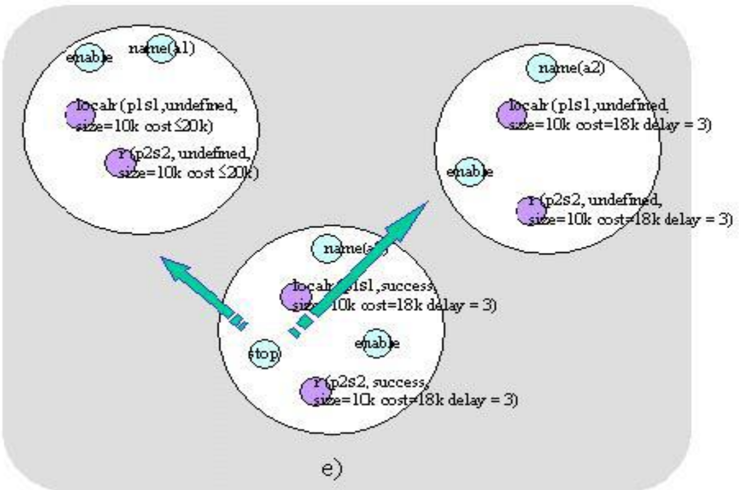
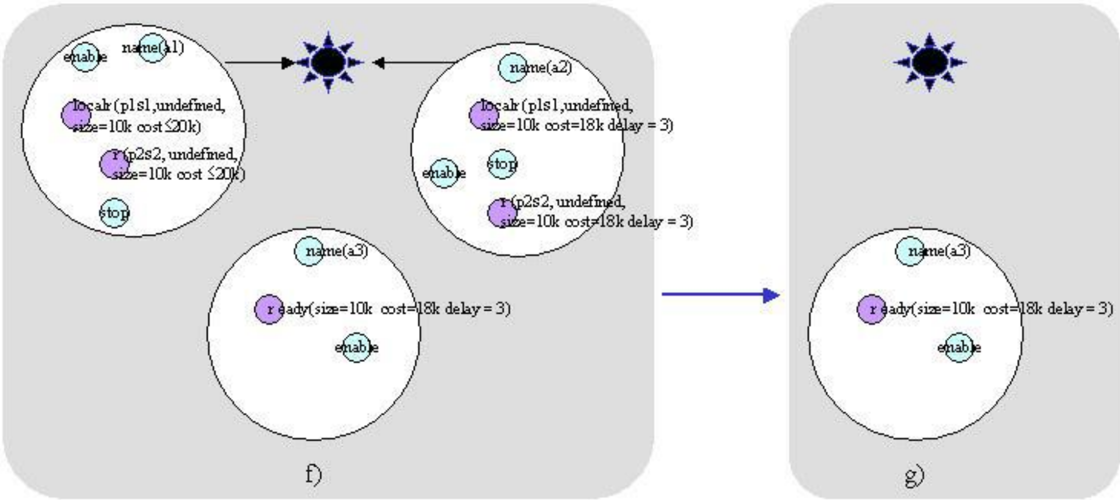


Fig. 3.e) Transformation of a3 atom state

The method *stop* <>- *#t* makes that, at a certain moment, the atoms containing *stop* particles be dissolved (see Figure 3. f.), the only active atom being the one containing the agreement (see Figure 3.g)).



Fig, 3. f) Dissolution of a1 and a2 atoms containing stop particle, **g)** The active atom: a3 atom containing the agreement

Thus, in this section, the negotiation process by using the IAMs metaphor has been defined, and the negotiation composed of a set of negotiation sequences has been modeled. Each of these sequences was represented by a set of negotiation atoms, which, at their turns, each of them administrates independently a private negotiation phase as well as the set of the snapshots associated to the phase.

A snapshot describes the status of the negotiation and the set of the negotiated attributes. Also, a negotiation is characterized by the role of participants and the policy of coordination attached to it. These policies can be described by using different methods. The proposed model of the negotiation process allows describing the evolution of a negotiation by a parallel development of negotiation atoms associated to methods modeling the coordination policy.

In the next sections, the coordination model will be defined by using this describing model of the negotiation process. Our objective is to achieve a correspondence between the coordination policy that a sequence has to satisfy (static model), and the set of methods modeling the evolution in time of a negotiation sequence (dynamic model).

Given that each negotiation is composed of a set of negotiation sequences, the coordination process of one or more negotiations will be structured into modules (services) that correspond to the sequences involved in the negotiation.

In the next section, we will briefly describe the architecture of the negotiation system in which the interactions take place.

4. The Collaborative Negotiation Architecture

The main objective of this software infrastructure is to support collaborating activities in virtual enterprises. In VE partners are autonomous companies with the same object of activity, geographically distributed.

Taking into consideration, the constraints imposed by the autonomy of participants within VE, the only way to share information and resources is the negotiation process.

Figure 4 shows the architecture of the collaborative system:

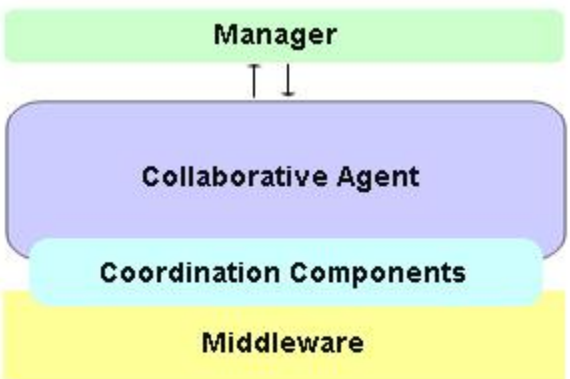


Fig. 4. The architecture of the collaborative system

This infrastructure is structured in ***four*** main ***layers***[1]: Manager, Collaborative Agent, Coordination Components and Middleware. A first layer is dedicated to the Manager of each organization of the alliance. A second layer is dedicated to the Collaborative Agent who assists its gas station manager at a global level (negotiations with different participants on different jobs) and at a specific level (negotiation on the same job with different participants) by coordinating itself with the Collaborative Agents of the other partners through the fourth layer, Middleware[2]. The third layer, Coordination Components, manages the coordination constraints among different negotiations which take place ***simultaneously***.

A Collaborative Agent aims at managing the negotiations in which its own gas station is involved (e.g. as initiator or participant) with different partners of the alliance.

Each negotiation is organized in three main steps: initialization; refinement of the job under negotiation and closing[3]. The initialization step allows to define what has to be negotiated (Negotiation Object) and how (Negotiation Framework)[4]. A selection of negotiation participants can be made using history on passed negotiation, available locally or provided by the negotiation infrastructure[5]. In the refinement step, participants exchange proposals on the negotiation object trying to satisfy their constraints[6]. The manager may participate in the definition and evolution of negotiation frameworks and objects[7]. Decisions are taken by the manager, assisted by his Collaborative Agent[8]. For each negotiation, a Collaborative Agent manages one or more negotiation objects, one framework and the negotiation status. A manager can specify some global parameters: duration; maximum number of messages to be exchanged; maximum number of candidates to be considered in the negotiation and involved in the contract; tactics; protocols for the Collaborative Agent interactions with the manager and with the other Collaborative Agents [9].

Conclusions

This paper proposes an intelligent mechanism for modeling and managing parallel and concurrent negotiations. The business-to-business interaction context in which our negotiations take place forces us to model the unexpected and the dynamic aspects of this environment. An organization may participate in several parallel negotiations. Each negotiation may end with the acceptance of a contract that will automatically reduce the available resources and it will modify the context for the remaining negotiations. We have modeled this dynamic evolution of the context using IAMs metaphor that allows us to limit the acceptance of a negotiation to the available set of resources. The proposed negotiation infrastructure aims to help the different SMEs to fulfil their entire objectives by mediating the collaboration among the several organizations gathered into a virtual enterprise.

A specific feature that distinguishes the negotiation structure proposed in this work from the negotiations with imposed options (acceptance or denial) is that it allows the modification of the proposals through the addition of new information (new attributes) or through the modification of the initial values of certain attributes (for example, in the case of gas stations the gasoline price may be changed).

In the current work we have described in our collaborative mechanism only the interactions with the goal to subcontract or contract a task. A negotiation process may end with a contract and in that case the supply schedule management and the well going of the contracted task are both parts of the outsourcing process.

In order to illustrate our approach we have used a sample scenario where distributed gas stations have been united into virtual enterprise. ***Take into consideration this*** scenario**,** one of the principal objectives was related to the generic case and means that this proposed infrastructure can be used in other activity domains.

Regarding research perspective continuation, we will focus on the negotiation process and the coordination process taking into consideration the contracts management process. In this way the coordination can administrate not only the dependence between the negotiations and the contracts which are formed and with execution dependences of those contracts.

[1] Cretan A., Coutinho C., Bratu B. and Jardim-Goncalves R., (2011), *A Framework for Sustainable Interoperability of Negotiation Processes.* Paper submitted to INCOM'12 14th IFAC Symposium on Information Control Problems in Manufacturing.

[2] Bamford J.D., Gomes-Casseres B., and Robinson M.S., (2003), *Mastering Alliance Strategy: A Comprehensive Guide to Design, Management and Organization*. San Francisco: Jossey-Bass, pp. 27-38

[3] Sycara K., (1991), *Problem restructuring in negotiation*, in Management Science, 37(10), pp.24-32.

[4] Smith R., and Davis R., (1981), *Framework for cooperation in distributed problem solving*. IEEE Transactions on Systems, Man and Cybernetics, SMC-11, pp. 42-57.

[5] Zhang X. and Lesser V., (2002), *Multi-linked negotiation in multi-agent systems*. In Proc. of AAMAS, Bologna, pp. 1207 – 1214.

[6] Barbuceanu M. and Wai-Kau Lo, (2003), *Multi-attribute Utility Theoretic Negotiation for Electronic Commerce*. In AMEC III, LNAI, pp. 15-30.

[7] Keeny R. and Raiffa H., (1976), *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. JohnWilley & Sons.

[8] Bui V. and Kowalczyk R., *On constraint-based reasoning in e-negotiation agents*. In AMEC III, LNAI 2003, pp. 31-46.

[9] Faratin P., (2000), *Automated service negotiation between autonomous computational agent*. Ph.D. Thesis, Department of Electronic Engineering Queen Mary & West-field College.

References

- Camarinha-Matos L.M. and Afsarmanesh H., *Collaborative Networked Organizations,* 2004 Kluwer Academic Publisher Boston.
- Singh M.P., *Commitments among autonomous agents in information-rich environments*. In Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW), pp. 141–155, May 1997.
- Bamford J.D., Gomes-Casseres B., and Robinson M.S., *Mastering Alliance Strategy: A Comprehensive Guide to Design, Management and Organization*. San Francisco: Jossey-Bass, 2003.
- Sycara K., *Problem restructuring in negotiation*, in Management Science, 37(10), 1991.
- Smith R., and Davis R., *Framework for cooperation in distributed problem solving*. IEEE Transactions on Systems, Man and Cybernetics, SMC-11, 1981.
- Zhang X. and Lesser V., *Multi-linked negotiation in multi-agent systems*. In Proc. of AAMAS 2002 July, Bologna, pg. 1207 – 1214.
- Barbuceanu M. and Wai-Kau Lo, *Multi-attribute Utility Theoretic Negotiation for Electronic Commerce*. In AMEC III, LNAI 2003, pg. 15-30.
- Keeny R. and Raiffa H., *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. JohnWilley & Sons, 1976.
- Bui V. and Kowalczyk R., *On constraint-based reasoning in e-negotiation agents*. In AMEC III, LNAI 2003, pp. 31-46.
- Faratin P., *Automated service negotiation between autonomous computational agent*. Ph.D. Thesis, Department of Electronic Engineering Queen Mary & West-field College, 2000.
- Cretan A., Coutinho C., Bratu B. and Jardim-Goncalves R., *A Framework for Sustainable Interoperability of Negotiation Processes.* Paper submitted to INCOM'12 14th IFAC Symposium on Information Control Problems in Manufacturing, 2011
- Cretan, A., Coutinho, C., Bratu, B., and Jardim-Goncalves, R., *NEGOSEIO: A Framework for Negotiations toward Sustainable Enterprise Interoperability.* Annual Reviews in Control, 36(2): 291–299, Elsevier, ISSN 1367-5788, 2012. http://dx.doi.org/10.1016/j.arcontrol.2012.09.010
- Vercouter, L., *A distributed approach to design open multi-agent system*. In 2nd Int. Workshop Engineering Societies in the Agents' World (ESAW), 2000
- Muller H., *Negotiation principles.* Foundations of Distributed Artificial Intelligence, 1996.
- Hurwitz, S.M., *Interoperable Infrastructures for Distributed Electronic Commerce.* 1998,
- Hu, J., Deng, L., "An Association Rule-Based Bilateral Multi-Issue Negotiation Model," in *Computational Intelligence and Design (ISCID),* Fourth International Symposium, Volume 2, pp. 234 – 237, 2011.
- Luo, X., Miao, C., Jennings, N., He, M., Shen, Z. and Zhang, M.," KEMNAD: A Knowledge Engineering Methodology for Negotiating Agent Development," *Computational Intelligence* ., 28 (1), 51-105, 2012 .
- Robinson W., and Volkov V., *Supporting the negotiation life cycle.* Communications of the ACM, 1998.
- Ossowski S., *Coordination in Artificial Agent Societies*. Social Structure and its Implications for Autonomus Problem-Solving Agents, No. 1202, LNAI, Springer Verlag, 1999.
- Schumacher M., *Objective coordination in multi-agent system engineering – design and implementation.* In Lecture Note in Artificial Intelligence, No. 2093, Springer Verlag, 2001.
- Sycara, K., Dai, T.,"Agent Reasoning in Negotiation," *Advances in Group Decision and Negotiation*, Volume 4, Handbook of Group Decision and Negotiation, Part 4, pp. 437-451, 2010.
- Kraus S., *Strategic negotiation in multi-agent environments.* MIT Press, 2001