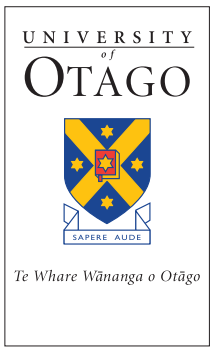

Establishing Dynamic Trust in Virtual Organization by Means of MAS

Noria Foukia
Pierre-Etienne Mallet

The Information Science Discussion Paper Series

Number 2006/06
March 2006
ISSN 1172-6024

University of Otago

Department of Information Science

The Department of Information Science is one of six departments that make up the School of Business at the University of Otago. The department offers courses of study leading to a major in Information Science within the BCom, BA and BSc degrees. In addition to undergraduate teaching, the department is also strongly involved in postgraduate research programmes leading to MCom, MA, MSc and PhD degrees. Research projects in spatial information processing, connectionist-based information systems, software engineering and software development, information engineering and database, software metrics, distributed information systems, multimedia information systems and information systems security are particularly well supported.

The views expressed in this paper are not necessarily those of the department as a whole. The accuracy of the information presented in this paper is the sole responsibility of the authors.

Correspondence

This paper represents work to date and may not necessarily form the basis for the authors' final conclusions relating to this topic. It is likely, however, that the paper will appear in some form in a journal or in conference proceedings in the near future. The authors would be pleased to receive correspondence in connection with any of the issues raised in this paper, or for subsequent publication details. Please write directly to the authors at the address provided below. (Details of final journal/conference publication venues for these papers are also provided on the Department's publications web pages: http://www.otago.ac.nz/informationscience/pubs/). Any other correspondence concerning the Series should be sent to the DPS Coordinator.

Department of Information Science
University of Otago
P O Box 56
Dunedin
NEW ZEALAND

Fax: +64 3 479 8311
email: dps@infoscience.otago.ac.nz
www: http://www.otago.ac.nz/informationscience/

Establishing Dynamic Trust in Virtual Organization by means of MAS

Noria Foukia
Information Science
University of Otago
P O Box 56, Dunedin NZ
0064 3 479 8386
nfoukia@infoscience.otago.ac.nz

Pierre-Etienne Mallet
Information Science
University of Otago
P O Box 56, Dunedin NZ
0033 616126130
pmallet@infoscience.otago.ac.nz

ABSTRACT
This paper presents an implementation of the first stage of a Virtual Organization (VO) life cycle, which is the VO's creation. This implementation is based on previous work by one of the authors describing a framework which facilitates the establishment of VO agreements. In accordance with the framework, the implementation makes the VO's creation fully automated, thereby reducing its duration considerably. This is beneficial for the VO, which should only exist for the limited period needed to satisfy its goal. The VO is implemented as a Multi-Agent System (MAS), where autonomous agents negotiate the agreement leading to the VO's establishment. The Opal FIPA-compliant MAS platform was used to implement the VO agents. Different scenarios and evaluations provide a clear demonstration of the implementation, showing how agents dynamically negotiate the establishment of the agreement and how opportunistic agents' behavior affect the trust level during the negotiation process.

Categories and Subject Descriptors
C.2.4 [**Distributed Systems]**

General Terms
Design, Experimentation, Security.

Keywords
Virtual Organization, Trust, Autonomy, Agent

1. INTRODUCTION AND MOTIVATION
The concept of Virtual Organization (VO) and its implication to business is well recognized. In broad terms a Virtual Organization can be defined as a temporary alliance of autonomous, diverse, and separately owned organizations where the participants share resources, information and knowledge in order to meet common objectives [1][9].

This definition encompasses several issues :

- Common objective: in the business world this mainly means that the objective of the members of the VO is to access one another's markets in order to maximize their profit.
- Individual autonomy of members: this gives them the benefits of utilizing the resources of a large organization while keeping their independence and autonomy during their memberships.
- Temporary alliance: a VO is usually established without long and cautious preparations and in a relatively short time. New members can extend the VO or old members can leave if their value for the VO does not meet their expectations.
- Separate ownership: a VO is composed of different legal entities that are often geographically dispersed.
- Strategic collaboration: the strategic intent of the VO is based on close collaboration between the separate members of the VO. This collaboration is possible if an acceptable level of trust is established between members. Establishing trust is necessary for VO's creation when members might have to act with uncertainty because they are in possession of ambiguous and incomplete information [7].
- Open Environment: VOs are open systems, in the sense that they are inhabited by members of whom nothing may be known in advance [13].

On-demand creation of VO allows to quickly exploit fast changing opportunities before these opportunities disappear. It allows their members to take advantage of often short-term agreements and cooperation with strategic partners.

[10] proposed three stages in a VO's creation, namely the creation, the execution and the dissolution. Since one issue of a VO is to become operational quickly, the time needed for the creation stage must be minimal [2]. It will also reduce the time of VO's life. Under such time constraint conditions, during the creation stage each potential partner has to ensure that he has enough information about the other VO's participants to establish an acceptable level of trust, and that the other participants fulfill his requirements.

In the VO, trust gives greater flexibility and can lead to competitive advantage [12]. But, trust is not easy to establish because it depends on the human intuition and the current context. However, there are some factors that can help to establish trust in VO:

- The reliability: for instance being able to deliver a product on time or in accordance with the terms specified in the agreement. Lack of reliability indicates that a participant is not keeping his word and maybe he is not acting according to the common objectives defined in the agreement. By exploiting individual objectives and not obeying agreed policies the participant breaks one fundamental principle of the VO.
- The responsibility: every member in the VO is responsible of its actions and also responsible of making the necessary

corrections when needed [3]. The responsibility can be expressed by acknowledging the problem areas and deficiencies and by producing compensation to these deficiencies (simple apology, acceptance of sanction in case of non performance,…). Compensation is important in order to restore broken trust before it is permanently lost. The compensation has to be correctly specified in the agreement and, during the execution stage of the VO, corresponding enforcement mechanisms should take care of its application in case of non conformance to the agreement.

In order to express the subjective and intuitive nature of trust in VOs four concepts have been taken from a previous work [1] and are explained below:

- **Expectation**: expresses the belief and wish with a confidence of fulfillment of obligations that an entity A has about the future action and behavior of an entity B.
- **Obligation**: expresses a contractual commitment or responsibility for an entity A to provide a commodity or a service under specified conditions (payment, profit, action and/or penalty for failing to comply,...).
- **Agreement**: is concluded when there is an explicit and declarative compatibility and a mutual assent of the parties (at least two) on given obligations and expectations.
- **Suspicion Level**: measures when the previously agreed trust is revoked. The conditions of revocation are specified in the agreement. The value of the suspicion level is the result of past transactions between VO's participants and factors of the current transaction of the agreement negotiation: lack of cooperation, opportunistic proactive behaviors, are factors for consideration in determining the value of the suspicion level. These factors are qualified as lack of goodwill [1].

In [1] we described a dynamic trust model for VO creation on the basis of these four concepts. In this new work, multi-agent (MA) technology has been integrated to help create the VO. The objectives are:

- Automation of the VO's creation using MA technology. This will reduce time of VO's creation and therefore the time of VO's life. Under such time constraint, economic partners could have no prior knowledge of each other. Collaboration between partners is also automated and could be used to obtain information about a new negotiating partner. This can help to determine the initial level of subjective trust attributed to this partner. The conditions of collaboration are specified in each partner's local policies.
- Support for dynamic aspect of trust during the agreement negotiation: trust is a dynamic social relationship that evolves as participants interact with each other over time and depending on the situation. In VO, partnerships and participant relationships are constantly changing. Agents are defined as acting on behalf of their representatives (here VO's members) to establish the VO. But, during the agreement negotiation they have the capability to delegate other agents to acquire new and available evidence when changing trust conditions require it.

This paper is structured as follows: Section 2 briefly introduces the framework presented in [1]. Section 3 describes our implementation using OPAL MA platform [4][8]. Section 4 develops a scenario and provides an evaluation of the implementation. Section 5 describes related works before the conclusion.

2. FRAMEWORK DESCRIPTION

This section summarizes the framework for the VO agreement's establishment that was presented in [1]. The framework provides a formalism to ensure that no important aspect of the agreement structure (social, economical, …) is missed during the negotiation. This framework helps to make the term of the contract unambiguous by providing additional checking if needed. This framework also forms the basis of the implementation of our OPAL MAS for trust negotiation in VO. The MAS allows the total automation of the VO's creation which reduces the time needed for setting up the VO. Another issue addressed by the framework is the capability to integrate the dimension of dynamics of trust during VO's creation. Indeed the notion of trust is not static but changes during the negotiation process depending on a more realistic notion of trust: the intuitive and subjective nature of human trust which evolves during the interaction when new evidences are produced during the interaction that will affect their point of view. Therefore, in the framework the level of trust that one party A grants to another party B may rely on already known factors which can be derived from past interactions between A and B, notoriety and know-how attributed to A and B. But the trust relationship between A and B also requires re-evaluation during the first stage of the VO cycle. Re-evaluation is done by collecting evidences during the current negotiation. Examples of evidence are, opportunistic behaviors (maximum profit), lack of conformance (bad quality), lack of reliability (high delivery delay) and affect directly the current level of trust.

The structures defined for the agreement creation between two entities A and B, are described below:

- participants A and B that wish to create a VO;
- suspicion levels sl_A and sl_B attributed to the participants A and B;
- The local policies L of a participant: each party defines also a set of local policies which control the good operation of the agreement negotiation and also control the conditions to release the sensitive and private information;
- negotiation history $H_{A,B}$ maintained locally by the participants A and B.

The negotiation process of the VO's creation is done through exchange of offers and counter-offers between at least two parties A and B. Offers and counter-offers are written as formal declarative rules including participants' obligations, participants' expectations, a set of verified attributes and a set of requested attributes. Expectations, verified attributes and requested attributes can be empty depending on the participants decision and on the current interaction. Formula 1 shows a typical agreement proposal made by participant A to participant B:

$$P_{A,B} = < E_{A,B} , O_{A,B} , \{s_A\}, \{s_B\} > \quad \textit{(Formula 1)}$$

where:

$E_{A,B}$: (Expectation from A to B) list of wishes including obligations that A requires from B;

$O_{A,B}$: (Obligations from A to B) set of points the entity A commits to fulfill;

$\{s_A\}$: verified attributes, credentials or evidences that A must prove to B;

$\{s_B\}$: requested attributes that B must prove to A.

When A initiates the negotiation with B, it sends a first proposal $P_{A,B}$ like in Formula 1. When receiving A's proposal, B generates its counter-proposal $P_{B,A}$, and sends it back to A. The negotiation continues with a sequence of proposals and counter-proposals. It ends when one party accepts the other's proposal or when it withdraws from the negotiation.

Each party also defines a set of policies which control the successful operation of the agreement negotiation and also control the conditions to release the sensitive information. These policies depend on the level of suspicion that each party is attributing to the other party during the negotiation (this suspicion level changes dynamically during the process of the agreement negotiation). Local policies are employed:

- To compute the suspicion level;
- To generate the parameters of expectations and obligations;
- To release the sensitive information;
- To generate the counter-proposal, stop the negotiation or accept the agreement.

An agreement between A and B is represented by the pair

$<O_{A,B}, O_{B,A}>$ which is the set of obligations that A accepted to undertake as part of the agreement and the set of obligations that B accepted to undertake as part of the agreement. Before reaching the final agreement $<O_{A,B}, O_{B,A}>$, at each offer (respectively counter-offer) that B receives from A (respectively that A receives from B), B (respectively A):

- calculates the suspicion level sl_A (respectively sl_B) attributed to A (respectively B) according to its local policies;
- generates its expectations and obligations according to its local policies;
- verifies that the proposed obligations and expectations fulfill its own requirements by using the fulfill() function that takes proposed and local expectations and obligations as input, and returns a decision whether the input fulfils the local requirements. This is described in Formula 2:

$$\mathbf{fulfill}(O_{A,B}, E_{A,B}, O_{B,A}, E_{B,A}) \rightarrow \mathbf{true/false} \qquad \textit{(Formula 2)}$$

- checks whether there is a sensitive attribute that it has to present to A (respectively B). Sensitive attributes are subject to enforcement policies that may require some evidences and credentials to be first presented by A (respectively B), otherwise a non-sensitive attribute is freely released in the counter-offer;
- generates a counter-offer.

The agreement is reached when:

$$\mathbf{fulfill}(O_{A,B}, E_{A,B}, O_{B,A}, E_{B,A}) \rightarrow \mathbf{true} \textit{ and}$$

$$\mathbf{fulfill}(O_{B,A}, E_{B,A}, O_{A,B}, E_{A,B}) \rightarrow \mathbf{true}$$

3. IMPLEMENTATION

In MAS, individual agents act in an autonomous and flexible manner in order to achieve their objectives. In open systems, agents can join and leave the system at any time. From these two features, we think that autonomous agents are good candidate to represent members of a VO. The Otago Agent Platform (OPAL) was used to implement the VO agents [4]. Opal is a highly modular FIPA-compliant MAS platform enabling the development of a MAS where agents can cooperate exchanging messages. It is based on Java Agent Services (JAS) public review specification which follows the FIPA [5] Abstract Architecture and provides standard Java interfaces for common agent concepts.

Opal combines the efficiency and robustness of MicroAgents called KEA (Kea is an alpine parrot [4]). and the openness of JAS that allows it to communicate with agents hosted by any other FIPA-compliant platform. These MicroAgents are reactive agents with basic capabilities like registering to a group of agents, searching for roles within the platform, or eventually they could be attributed reasoning capabilities.

3.1 Agent's description

An agent of the VO framework should be viewed as a high level agent (VO Agent) composed of several low level ones. Each low level agent is dedicated to a single task. The different agents and tasks are shown in Figure 1 and described below:

- **MessageProcessor**: this agent processes the incoming messages from the VO agent.
- **InformationSeeker**: this agent is queried by other Negotiators to answer queries about specific information (for instance a suspicion level attributed by collaborating VO Agents).
- **Negotiator**: this agent is created by the VO Agent to conduct a negotiation with another agent.

The implementation provides one MessageProcessor and one InformationSeeker within a high level agent and it provides the capability of having one Negotiator for each current negotiation. In practice, this allows each participant in the VO's creation to negotiate with multiple participants. This is used for instance if a vendor is trying to determine the better price offer from several manufacturers by negotiating the same proposal in parallel with each of them. Each Negotiator is independent from the others and acts autonomously on behalf of its corresponding VO Agent. The agent/vendor might also benefit from conducting several asynchronous negotiations by terminating all the negotiations if one of them leads to an agreement.

The Negotiator implements the update_sl(), fulfill(), attribute_release() and counter_proposal() functions that have been defined in the framework.

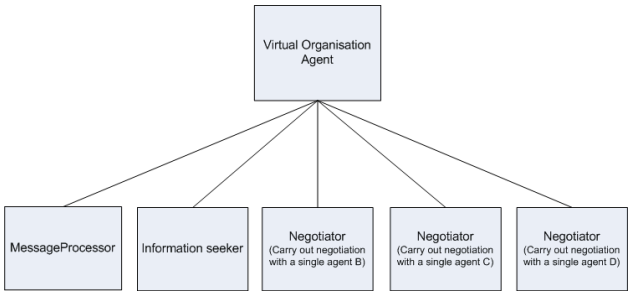

Figure 1. Architecture of a VO agent.

During the negotiation, a local policy in ***L*** can specify that an agent requires verification or recommendation from collaborating agents (federation of partners) to establish/update the current suspicion level. In this situation, the Negotiator asks the InformationSeeker to retrieve additional information by contacting the collaborating agents.

The InformationSeeker's help is also requested by the Negotiator if the sensitivity of an attribute requires additional information retrieval by the InformationSeeker. This extends the capacities provided by attribute_release() in the framework description.

3.2 Action sequence of the negotiation

Figure 2 shows a first round negotiation between VO Agent A and VO Agent B.

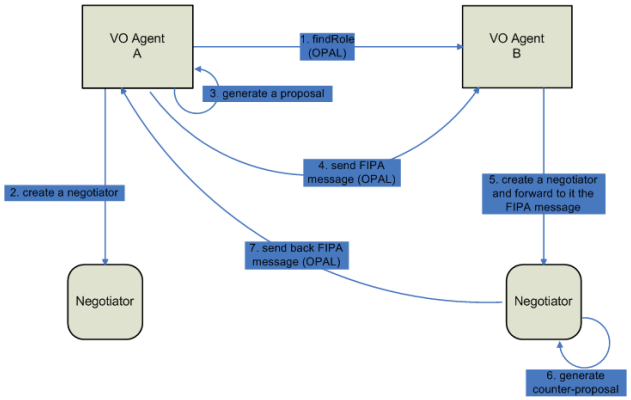

Figure 2. Sequence of actions for a first round negotiation.

1. VO Agent A checks first if VO Agent B is available by calling the FindRole method provided by OPAL;
2. VO Agent A creates a Negotiator;
3. VO Agent A creates a proposal on behalf of its representative (company, individual),
4. and sends it to VO Agent B as a FIPA message. This is also provided by OPAL. This message is processed by our specific MessageProcessor which checks the type and the sender of the message;
5. Since it is a new negotiation VO Agent B creates a Negotiator and the MessageProcessor sends the FIPA message to it;
6. Negotiator creates a counter-proposal,
7. and sends it as a FIPA message to VO Agent A that forwards it to its MessageProcessor. MessageProcessor checks the type and the sender and passes the message to the existing Negotiator.

3.3 Example scenario

In this scenario we imagine a federation of companies negotiating different products (raw and finished goods) which could exchange (buy and sell) their good by establishing virtual agreements. New members can join or leave the federation to establish a VO. The following illustrates a simple agreement negotiation between 2 members of the corporation: a manufacturer ***A*** that sells finished goods (in this case finished computers) and a supplier ***B*** that provides raw parts of the final computers (in this case monitors).

Each participant defines the minimum profit share it is willing to accept in its local policies ***L***. This information is secret and is not revealed to the other party.

1. **The class Policy contains policies defined by each participating agent as described in Figure 3.**

```
/* Threshold for the suspicion level */
private float slThreshold = (float)0.6 ;
/* The minimum suspicion level needed to trust an agent */
private float trustThreshold = (float)0.4 ;
/* Minimum profit accepted */
private double Im = 30 ;
/* Maximum fine accepted */
private int maxFine = 100000 ;
/* Fine required for the other negotiators */
private int requiredFine = 10000 ;
/* Default sl used when no history available for agent */
private float defaultSl = (float)0.5 ;
/* The required quality assurance */
private Set requiredSb = new HashSet() ;
/* The provided quality assurance */
private Set providedSb = new HashSet() ;
/* Accepted authorities */
private Set authorities = new HashSet() ;
/* Describe what this agent provide */
private Set providedServices = new HashSet() ;
/* Sensitive information */
private Set sensitive = new HashSet() ;
```

Figure 3. Example of policies defined for one agent.

Round 1: ***A*** initiates a negotiation by sending a message to ***B*** that contains the first agreement proposal:

$\mathbf{P_{A,B1} = <E_{A,B1}, O_{A,B1}, Ø, s_B >}$

The proposal contains:

- *A's* expectation $\boldsymbol{E_{A,B1}}$ ***= (xship, rmonitor, wfine)*** for ***B*** to update the stock of monitors and a sanction specified in ***wfine*** in the case of non-performance;
- *A's* obligation $\boldsymbol{O_{A,B1}}$***= (xshare, rprofit, 20%)*** to give ***B*** 20% of the profit;
- the quality assurance requirement $\boldsymbol{s_B}$ that ***B*** needs to prove to ***A*** in order to satisfy the compatibility and quality requirements posed by ***A*** on ***B's*** product;
- an empty set of *A's* verified credentials.

2. **The proposal sent by *A* is shown in Figure 4.**

```
Proposal:
Expectations
object : ship_monitor
fine : 10000
Obligations
share : 20.0
Requested quality assurance: ISO9001
Verified quality assurance:
```

Figure 4. Proposal sent by agent *A* to agent *B*.

When ***B*** receives ***A's*** proposal, it first calculates the suspicion level sl_A attributed to ***A***. Since there was no prior interaction between ***A*** and ***B***, the sl_A is set to a default value ***sladefault***.

update_sl(Ø , Ø, L_B)→ sladefault

The local policies of ***B***, L_B, are employed when calculating the suspicion level.

3. The function update_sl() is described in Figure 5.

```
private void update_sl(){
/* case of the first round and no negotiation history */
      if((round == 1) && (sl == 0)){
         sl = localPolicy.getDefaultSl() ;
         String[] info = {AgentInfo.SUSPICION_LEVEL};
         sl = (Float) infoSeeker.getInformation(participant, info, 5000,
5).get(AgentInfo.SUSPICION_LEVEL);
      } else
/* after one round of negotiation */
        if(round > 1){
            double myShare = (Double)
localProposal.getObligations().get(Obligations.SHARED_PROFIT);
            double hisShare = (Double)
receivedProposal.getObligations().get(Obligations.SHARED_PROFIT);
            double hisPreviousShare = (Double) ((Proposal)
            history.get(2 * (round -
1))).getObligations().get(Obligations.SHARED_PROFIT);

/*if the negotiator not too opportunistic: increase of profit share < 5 % */
         if(hisShare < (100 - myShare) && ((hisShare - hisPreviousShare) < 5)){
            sl += 0.08 ;
          }
/* if the negotiator is too opportunistic*/
         if(hisShare < hisPreviousShare)
            sl += 0.1;
            }
      }
```

Figure 5. Description of update_sl() of agent *B*.

Next, ***B*** generates its expectations $E_{B,A}$ and obligations $O_{B,A}$ according to the local policies L_B that indicate, for example, the minimum profit of 60%.

4. localPolicy.getIm() returns the specified value of the minimum profit share accepted by *B* and defined in *B*'s local policy.

Next, ***B*** checks whether the proposed obligations and expectations fulfill its own requirements. Since ***A*** proposed lower profit share than ***B*** expects, the ***fulfill*** function returns ***false.*** Assume that ***B*** accepts the requested sanctions ***wfine***.

fulfill ($O_{A,B1}$, $E_{A,B1}$, $O_{B,A}$, $E_{B,A}$) → false, where

$O_{B,A}$ = (xshare , rprofit , 40% ,wfine),

$E_{B,A}$ = (xship , rmonitor , wdisputeprofit)

5. The verification of fulfill ($O_{A,B1}$, $E_{A,B1}$, $O_{B,A}$, $E_{B,A}$) is shown in Figure 6.

Since there is a requested attribute s_B that ***B*** has to present to ***A***, ***B*** must first check whether the attribute is sensitive or not. Assume that the attribute is not sensitive and can be freely released.

attribute_release(sladefault, $P_{A,B1}$) → s_B

```
if(fulfill(receivedProposal.getObligations(),
receivedProposal.getExpectations(),
localProposal.getObligations(),
localProposal.getExpectations())){

/*build response message*/
Message response = new Message();
response.setReceiver(message.getSender());
response.setSender(message.getReceiver());
response.set(Message.ACT,Message.ACCEPT_PROPOSAL);
response.set(Message.LANGUAGE, "XML");
//convert the proposal to XML
XMLSerializer ser = new XMLSerializer();
String content = ser.serialize(receivedProposal);
response.set(Message.CONTENT, content);
response.set(Message.CONVERSATION_ID, round);
//send the response
((Platform)
SystemAgentLoader.findRoles(Platform.class)[0]).send(response);
System.out.println("We get a deal with"+message.getSender()+"Negotiation
finished");
closeNegotiation(true);
}
```

Figure 6.Verification that requirements match by fulfill.

6. The function attribute_release(proposal) is shown in Figure 7.

```
private Set attribute_release(Proposal proposal){
    Set release = new HashSet() ;
/* check out the requested attribute */
    Iterator iter = proposal.getRequested().iterator();
    while(iter.hasNext()){
       String req = (String) iter.next();
       System.out.println("att_release: "+req);
/* if attribute is sensitive */
       if(localPolicy.isSensitive(req)){
          /* update the suspision level */
          String[] info = {AgentInfo.SUSPICION_LEVEL};
          sl = (Float) infoSeeker.getInformation(participant, info, 5000,
5).get(AgentInfo.SUSPICION_LEVEL);
/* if  suspicion level too high, attribute is not released */
          if(sl <= localPolicy.getSlThreshold())
             release.add(req) ;
          }else if(localPolicy.isProvided(req))
          release.add(req);
    }
    return release ;
 }
```

Figure 7. Verification and release of attributes by attribute_release.

Next ***B*** generates a counter-proposal $P_{B,A1}$. The proposal indicates that:

- ***B*** is willing to supply the monitor if ***A*** accepts to take only 40% of the profit;
- ***B*** agrees to pay fine ***wfine*** if ***B*** does not perform on time;
- ***A*** will be a subject to a legal dispute ***wdispute*** if ***A*** does not provide the agreed profit share.

Next ***B*** sends the counter-offer and its quality certification s_B to ***A***.

counter_offer(sladefault, $P_{A,B1}$)→ $P_{B,A1}$

$P_{B,A1}$ = < $E_{B,A1}$, $O_{B,A1}$, s_B, Ø >, where

$E_{B,A1}$ = (xship , rmonitor , wdispute),

$O_{B,A1}$ = (xshare , rprofit , 40% , wfine)

7. **The function counter-offer(receivedProposal) generates the counter-offer. The counter-offer is serialized and sent to *A*.**

```
localProposal = counter_offer(receivedProposal);
...
String content = ser.serialize(localProposal);
Message response = new Message();
response.setReceiver(message.getSender());
response.setSender(message.getReceiver());
response.set(Message.ACT, Message.PROPOSE);
response.set(Message.CONTENT, content);
...
((Platform) SystemAgentLoader.findRoles(Platform.class)[0]).send(response);
```

Figure 8. Generation and sending of counter_offer.

Rounds 2 to N

When ***A*** receives ***B's*** counter-offer, ***A*** notices that ***B's*** profit expectation is higher than the minimum profit specified in ***A's*** local policies L_A (which indicates the minimum profit of 50%), so it makes a counter offer setting ***B's*** profit to a lower value.

The negotiation proceeds in a similar fashion until the agreement is reached. If ***B*** is acting suspiciously, for example, constantly exhibiting opportunistic behaviors by not lowering its profit expectation (or even rising the profit expectation each new round), ***A*** will increase the suspicion level sl_B. This behavior is not necessarily malicious, it is natural for one to wish for a higher profit. However, ***A*** can infer that ***B*** is likely trying to probe the maximum profit share that ***A*** can offer, which is a sign of "lack of good will". Such behavior may indicate potential future problems in the collaboration with ***B***. When sl_B reaches certain threshold Th (defined in the policies L_A), ***A*** may decide not to establish the agreement with ***B*** (by sending a negotiation failure message to ***B)*** and find a more cooperative partner to create a VO.

4. EVALUATION

This section describes two scenarios to validate the implementation. The first scenario is called isolation scenario and the second one is called collaboration scenario.

- In the isolation scenario 25 agents negotiate in isolation the agreement of the VO's creation. That is no agent is sharing information with any other agent. This is equivalent of a federation of agents where no one trusts the other ones.
- In the collaboration scenario, the 25 agents form a federation where they can exchange information when they negotiate the agreement for the VO's creation.

We define two kinds of agent populations:

- opportunistic agents: opportunistic agents try to maximize their wealth, at the expense of others if necessary. This is implemented by a specific agent (called ***OP***) which systematically increases its profit share if its first proposal of the profit share is immediately accepted by any other agent (called ***A***). That means that agent ***OP*** is trying to guess and reach the minimum profit share that agent ***A*** is willing to accept (specified in agent ***A***'s local policies $\mathbf{L_A}$) and agent ***OP*** will always try to get more than the minimum profit share specified in its local policy $\mathbf{L_{OP}}$).
- fair agents: fair agents do not try to increase their profit share beyond the minimum profit share value specified in the local policy. A fair agent (called ***FR***) will send a counter-proposal increasing its profit share only if the profit share proposed by ***A*** is below its minimum profit share value (specified in $\mathbf{L_{FR}}$).

4.1 Isolation scenario

The first experiment is conducted with different population sizes of opportunistic agents. There is no information exchange between agents apart from the proposals and counter-proposals sent between two agents to negotiate an agreement. When an agent finishes (reaches an agreement or not) a negotiation, it immediately looks for another available agent to begin a new negotiation. In the experiment, the suspicion levels are generated in the interval [0.0, 1.0]. 0 represents no suspicion and 1 represents maximum suspicion. As any agent, ***A*** identifies the behavior of an agent ***OP*** as opportunistic, it increases the local suspicion level attributed to ***OP*** by 0.1. Each agent defines in its local policy the minimum profit share it is willing to accept. If below this value the agent decides to terminate the negotiation without reaching an agreement. Each agent also defines the threshold (Th) of suspicion level to ban an agent. If it is above this value the agent decides to refuse any future negotiation.

Table 1 summarizes the averaged results obtained when 5, 10 and 20 opportunistic agents are present in a population of 25 agents. The results are collected after a period of 10 minutes. The same experiment is repeated 20 times for each population size and each column gives an average of the 20 times. Table 1 gives the average suspicion attributed to the opportunistic agents by the other agents after 10 minutes of operation: [sl_{OP}]; the average portion of successful agreements for the opportunistic agents ***OP*** after 10 minutes of operation: [$\%N_{OP}$]; for each agent ***OP*** the average number of agents which bans this ***OP*** after 10 minutes of operation: [BAN]/OP; the average suspicion attributed to the fair agents by the other agents after 10 minutes of operation: [sl_{FR}]; the average portion of successful agreements for the fair agents ***FR*** after 10 minutes: [$\%N_{FR}$]. The initial suspicion level is set to 0.2.

Table 1. Isolation with Opportunistic Behavior

OP	[sl_{OP}]	[$\%N_{OP}$]	[BAN]/OP	[sl_{FR}]	[$\%N_{FR}$]
5	0.46	45%	5.4	0.25	85%
10	0.51	40%	7.5	0.23	73%
20	0.46	18%	5.7	0.24	46%

4.2 Collaboration scenario

The conditions of the second experiment are similar to the first experiment. However, during a negotiation an agent ***A*** can exchange information with other collaborating agents when the agent ***A*** has a doubt of suspect behavior in the current negotiation. The information received from the collaborating agents is incorporated in agent ***A***'s parameters based on the level of trust attributed to each collaborator (this is an average of the information provided by the collaborators weighted by the trust level attributed to each collaborator (**TR**)). The trust rates vary in the interval [0.0, 1.0]. For the experiment the trust rates are randomly attributed so that two zones of trust are defined:

1. low trust: average trust rate < 0.6
2. high trust: : 0.6 ≤ average trust rate ≤ 1.0

Each agent defines also in its local policies the conditions to require the information from a collaborator. In the experiment, an agent ***A*** requires information from its collaborators when the current suspicion level attributed to a negotiating agent ***B*** reaches a certain value Slmax < Th. This gives agent ***A*** a some room to decide whether or not to ban b. Slmax is chosen so that Slmax = $\frac{Th}{2}$

Table 2 and Table 3 summarize the average results obtained when 5, 10 and 20 opportunistic agents are present in a population of 25 agents. Table 2 corresponds to 0.6 ≤ average trust rate ≤ 1 (high trust among agents). Table 3 corresponds to average trust rate < 0.6 (low trust among agents). The results are collected after a period of 10 minutes. The same experiment is repeated 20 times for each population size and each column gives an average of the 20 times. Table 2 and Table 3 collate the same information as Table 1.

Table 2. Collaboration with Opportunistic Behavior:TR ≥ 0.6

OP	[sl_{OP}]	[$\%N_{OP}$]	[BAN]/OP	[sl_{FR}]	[$\%N_{FR}$]
5	0.40	56 %	2.0	0.23	93 %
10	0.41	53 %	3.8	0.23	85%
20	0.36	33 %	0.9	0.24	74%

Table 3. Collaboration with Opportunistic Behavior: TR < 0.6

OP	[sl_{OP}]	[$\%N_{OP}$]	[BAN]/OP	[sl_{FR}]	[$\%N_{FR}$]
5	0.59	27 %	15.8	0.43	74 %
10	0.59	20 %	14.8	0.42	65 %
20	0.56	9 %	12.0	0.42	35 %

The analysis of these results allows the following observations:

- Comparing Table 1 and Table 2 it can be seen that collaboration tends to be advantageous to opportunistic agents which obtain a higher percentage of successful negotiations and are banned less by other agents. This is explained by the fact that at the beginning of the 10 minutes of experiment there is no prior and real knowledge attributed to the opportunistic behavior. If for example an agent ***A*** identifies an agent ***B*** as potentially opportunistic, ***A*** asks other collaborators within the VO for information about ***B***. Since the collaborators do not have any previous interactions with ***B***, and because the level of trust between collaborating agents is high (TR≥ 0.6), they send back their default suspicion level which is low (0.2). This lowers the suspicion level that ***A*** attributes to ***B***. Therefore the opportunistic behavior need a certain time before being identified. This is only true at the start, eventually opportunistic agents would be known and there will be more accurate information distributed through collaboration.
- Under collaboration, when the level of trust is low (TR < 0.6) as shown in Table 3, opportunistic agents are easily banned and their average percentage of successful negotiation is quite low. Also fair agent fails slightly more negotiations. This is due to the fact that the decision of each agent to ban or to conclude a negotiation with another agent, is mitigated by the low trust rate (TR).

4.3 Typical opportunistic behavior

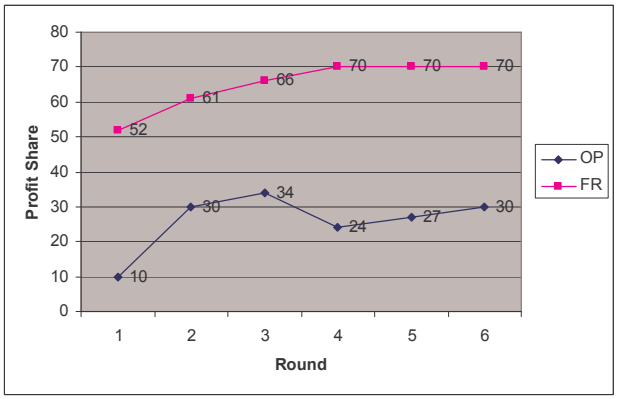

Figure 9. Example of negotiation with an opportunistic agent.

Figure 9 shows a typical negotiation between an opportunistic agent ***OP*** and a fair agent ***FR***. ***OP*** starts by proposing to give ***FR*** 10% of the profit. ***FR*** sends back a counter-proposal granting ***OP*** 52% of the profit. At round 3, ***OP*** and ***FR*** theoretically reached an agreement when ***OP*** should give ***FR*** 34% of the profit. Since ***OP*** is opportunistic, it is trying to get a better profit share. Instead of accepting to give the theoretical 34%, ***OP*** sends a new offer of 24%. ***FR*** accepts to give ***OP*** 70% of the profit since its local policy $\mathbf{L_{FR}}$, specifies 30% for the minimum accepted profit. After several rounds, ***OP*** notices that ***FR*** won't accept to decrease its profit, therefore ***OP*** agrees on 30%. At the end of the negotiation, ***OP*** made an increase of 4% in its profit: ***OP*** gave 30% instead of 34%.

5. RELATED WORKS

Different works refer to MAS for building VO. Mainly the proposed approaches do not address the aspect of dynamic trust in VO's creation. [14] implements a MAS that simulates an artificial marketplace. A negotiation protocol also based on agent expectation supports the formation of the VO. A penalty mechanism is added as an incentive to encourage an agent to fulfill its promise to the VO. The work does not discuss the aspect of trust between agents.

In [13] an agent metaphor is used as a conceptual model of VOs where interactions between an agent and a normative system, also represented as an autonomous agent, are considered. The normative system defines the roles that agents can play in the system. Agents negotiating contracts are subject to obligations issued by the normative system but they also create obligations directed towards each other. The normative agents can dynamically introduce new norms and ordinary agents can also add obligations, prohibitions and permissions. Interactions

between agents are governed by local conditional or production rules. The way that an agent makes its decision crucially depends on the way the conflicts between its rules are resolved. The model captures some kind of dynamics of a normative system by observing the effect of an ordinary agent on the normative system. However, [13] does not clearly discuss the effect of the agents' behavior on trust during the contracting process.

[6] proposes an agent-based model that allows agents to negotiate trust in situations where conflicts can arise between agents. The model reduces the uncertainty between negotiating agents by means of repeated interactions. These interactions are direct based on confidence or indirect based on reputation mechanisms. The trust model aims at choosing agents that are most reliable and honest in the long run. This long run can improve the reputation (that is included in the history) of a partner and will be considered during the negotiation. However, the long run is not suitable at the stage of the VO's creation because one premise of the VO is to minimize the time needed for the its creation.

Similar to our framework, [2] proposes a formal representation of VO contracts as a basis to check consistency and compliance of the agreement. Common issues are discussed such as: reducing the VO's time of operation, automating the VO's creation, monitoring the actual execution of contracts to notify the partners in case of non-compliance. However, [2] does not integrate the notion of trust and suspicion in the framework.

6. CONCLUSION AND FUTUR WORK

This paper has outlined our work in progress, which uses: 1) The framework presented in [1] to facilitate the process of establishing a VO and 2) The corresponding implementation using OPAL MA platform. The novelty of the approach is that it provides support for dynamic specification of trust during the VO creation process. In accordance with the framework, the implementation makes the VO's creation fully automated, thereby considerably reducing its duration. As a demonstration of our implementation, we developed two scenarios where agents can act in isolation or in collaboration when negotiating new agreements. The effect of isolation and collaboration on opportunistic agents is compared as well as the effect of trust rate in the collaboration. Future work includes describing new scenarios where agents will be involved in multi-party negotiations. These multi-party negotiations can already be supported by the implementation of the VO Agent. Another major issue would be to add an authentication scheme (based on certification) between agents to allow agents authenticate themselves when required by negotiating agents.

7. ACKNOWLEDGMENT

We would like to thank the authors of [1] for their contribution on the framework elaboration which is at the basis of the implementation presented in this paper.

8. REFERENCES

[1] Ryutov, T., Neuman, C., Zhou, L., and Foukia, N. Establishing Agreements in Dynamic Virtual Organizations. *In Proceedings of SECOVAL, The Workshop on the Value of Security through Collaboration, SecureComm 2005* (Athens, Greece / Hotel Amarilia / 5 - 9 September, 2005).

[2] Quirchmayr, G., Milosevic, Z., Tagg, R., Cole, J., and Kulkarni, S. Establishment of Virtual Enterprise Contracts. *In Proceedings of DEXA'02 conference* (September 2-6, 2002).

[3] Albu, G. Trust. Mediator, Federal Mediation and Conciliation Service, A quarterly publication of the Ohio State University Leadership Center (USA 2002).

[4] Nowostawski, M., Purvis, M., and Cranefield, S. OPAL A Multi-level Infrastructure for Agent-Oriented Development. *In Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002)*. ACM Press (2002) pp.88-89.

[5] FIPA. Foundation For Intelligent Physical Agents (FIPA). FIPA 2000 specifications. http://www.fipa.org/specifications/index.html, 2000.

[6] Ramchurn, S. D. *Multi-Agent Negotiation using Trust and Persuasion*. Ph.D. Thesis, University of Southampton, UK, 2004.

[7] Ramchurn, S.D., Huynh, T.D., and Jennings, N.R. Trust in Multi-Agent Systems. *The Knowledge Engineering Review* (2004).

[8] Weng, M., Nowostawski, M., and Purvis, M. Declarative Agent Programming Support for a FIPA-Compliant Agent Platform. *In Proceedings of the Third International Workshop on Programming Multi-Agent Systems (ProMAS 2005)* Fourth International Joint Conference on Autonomous Agents & Multi (Utrecht, The Netherlands 2005).

[9] Davulcu, H., Kifer, M., Pokorny, L.R., Ramakrishnan, C.R., Ramakrishnan, I.V., and Dawson, S. Modeling and Analysis of Interactions in Virtual Enterprises. *In Proceedings of the Ninth International Workshop on Research Issues on Data Engineering*. Information Technology for Virtual Enterprises, p.12, March 23-24, 1999.

[10] Tagg, R. Workflow in Different Styles of Virtual Enterprise. *In Proceedings of Workshop on Information Technology for Virtual Enterprises* (ITVE 2001). Gold Coast, Australia, January 2001.

[11] Goodchild, A., Milosevic, Z., and Herring, C. Business Contracts for B2B. *In Proceedings of the CAISE'00 Workshop on Infrastructure for Dynamic Business-to-Business Service Outsourcing* (Stockholm, Sweden, 5-6 June 2000).

[12] Barney J.B., and Hansen M.H. Trustworthiness as a Source of Competitive Advantage. *Strategic Management Journal*, Vol. 15, pp. 175-190. 15 (1994).

[13] Boella, G., Hulstijn, J., and Van der Torre, L. Virtual Organizations as Normative Multiagent Systems. In Proceedings of Hawaii International Conference on System Sciences (HICSS). (Hawaii, 2005, USA).

[14] Qinhe Zheng, Xiaoqin (Shelley) Zhang. Study of Virtual Organizations Using MultiAgent System. *In Proceedings of the Fourth International Joint Conference on Autonomous Agents & Multi Agent Systems* (Utrecht, The Netherlands 2005).