Specifying E-Alliance contract dynamics through the $\mathcal{M}$OISE$^+$ reorganisation process*

Jomi Fred Hübner[1] †, **Olivier Boissier**[2] , **Jaime Simão Sichman**[3] ‡

[1]GIA / DSC / FURB
Braz Wanka, 238
89035-160, Blumenau, Brazil

[2]SMA / G2I / ENSM.SE
158 Cours Fauriel
42023 Saint-Etienne Cedex, France

[3]LTI / EP / USP
Av. Prof. Luciano Gualberto, 158, trav. 3
05508-900 São Paulo, SP, Brazil

jomi@inf.furb.br, Olivier.Boissier@emse.fr, jaime.sichman@poli.usp.br

Abstract. *When enterprises join to form virtual dynamic enterprises, they need to state their contracts. In this paper we propose to use the $\mathcal{M}$OISE$^+$ organisational model to describe constraints and obligations entitled in a contract. The model is suitable for this application since it was developed to deal with dynamic environments where the organisation, and thus the contracts, may change in a kind of reorganisation.*

1. Introduction

The main goal of the E-Alliance project is to provide an environment supporting collaboration activities across autonomous enterprises grouped into alliances [Carron and Boissier 2001, Andreoli et al. 2000]. Such an environment should support information sharing and collaborative decision-making across the partner organisations, preserving their autonomy. In order to get the application domain more concrete, it is considered the case of an alliance of autonomous printshops. In this case, the printshops are autonomous enterprises, fully responsible for their budget and for the planning and scheduling of their print jobs and resources. Printshops may create and join an alliance in order to accomplish or better accomplish customers' print requests that they cannot or do not want to satisfy alone. In this case, the goal is to support collaborative executions initiated by printshops willing to out-source some of their jobs, as a whole or in parts.

Before a group of printshops decide to collaborate, they need to previously agree on the responsibilities of each one in the alliance. Normally, this agreement is achieved by a negotiation process and, when finished, the result is explicitly stated in a contract. In this article, we propose the utilization of a Multi-Agent System (MAS) organisational

*This research was partially supported by USP/COFECUB project ORGMAS.
†Supported by FURB, Brazil; and CNPq, Brazil, grant 200695/01-0.
‡Partially supported by CNPq, Brazil, grant 301041/95-4; and by CNPq/NSF PROTEM-CC MAPPEL project, grant 680033/99-8.

description language for helping the agents in the negotiation process and also in the contract specification. More precisely, we state that the $\mathcal{M}$OISE$^+$ organisational model [Hübner et al. 2002, Hübner 2003] can be used in the definition of contract types which can be used as negotiation objects, i.e., the constraints and obligations of each part of the a contract.

2. The E-Alliance dynamics as an reorganisation process

The E-Alliance context can be seen as an open MAS where assignment of an organisation is useful to deal with the problems that could arise from the agents' autonomy, where we do not know what kind of enterprise will enter into the system (this motivation for organised MAS is well described in [Sierra et al. 2004, de Almeida Júdice Gamito Dignum 2003]). In this context, the organisation is a set of behavioural constraints that a group of agents (enterprises) adopts in order to control the agent's autonomy and easily achieve their global purposes [Dignum and Dignum 2001]. This approach is based on human societies that are successfully using organisation (e.g. social roles) to have a global coherent behavior. The definition of a proper organisation for a MAS is not an easy task, once the organisation could be too flexible (the organisation does not help the achievement of the global purpose) or too stiff (the organisation extinguishes the agent's autonomy). A initial good organisation is normally set up by the MAS designer, however it may become not suitable in dynamic environments like the E-Alliance. Thus, a reorganisations process is mandatory in this domain.

We suppose that the alliance starts with a very simple organisation: all agents have the printshop role and have communication links among them. During the alliance life, the agents may realise that their organisation is not playing its function: help the agent to achieve their common purpose. In this case, the alliance members need to change their organisation.

To take the reorganisation into account, we propose that the alliance has a *reorganisation* group of agents. One possible specification for this group is described in [Hübner et al. 2004] where the reorganisation process has the following steps: monitoring (when reorganise), design (manners of building a new organisation), selection (how to choose an organisation), and implementation (how to change the current running organisation). These steps are carried out by a group created from the `ReorgGr` specification which is defined in the Fig. 1.[1] The *soc* role is an abstract role (no agent can play it) which is the root of the $\mathcal{M}$OISE$^+$'s role hierarchy, thus every role defined in a $\mathcal{M}$OISE$^+$ organisation is a specialisation of *soc*. The *OrgManager* authority on the *soc* means therefore an authority on all roles. The *Monitored* is another abstract role which is specialised by roles whose agents will be monitored by a *Monitor*. In other words, all agents that will be monitored must have a *Monitored* sub-role. The *Designer* is an abstract role that contains the common properties for designers (*ReorgExpert* and *OrgParticipant*). The *Reorg* is also an abstract role which allows us to easily distinguish the *OrgManager* and the other roles in this group. Thus we can state, for example, that the *Reorg* and therefore all its

[1]The graphical notation used in Fig. 1 follows the $\mathcal{M}$OISE$^+$ model. The model and the notation is not described in detail in this paper, the reader is referred to [Hübner et al. 2002, Hübner 2003] for more information. Throughout the text, we will introduce the $\mathcal{M}$OISE$^+$ elements using examples and informal definitions.

sub-roles has permission to communicate with the *OrgManager* role. It is also useful to state that all roles are allowed to communicate with `ReorgGr`'s members.

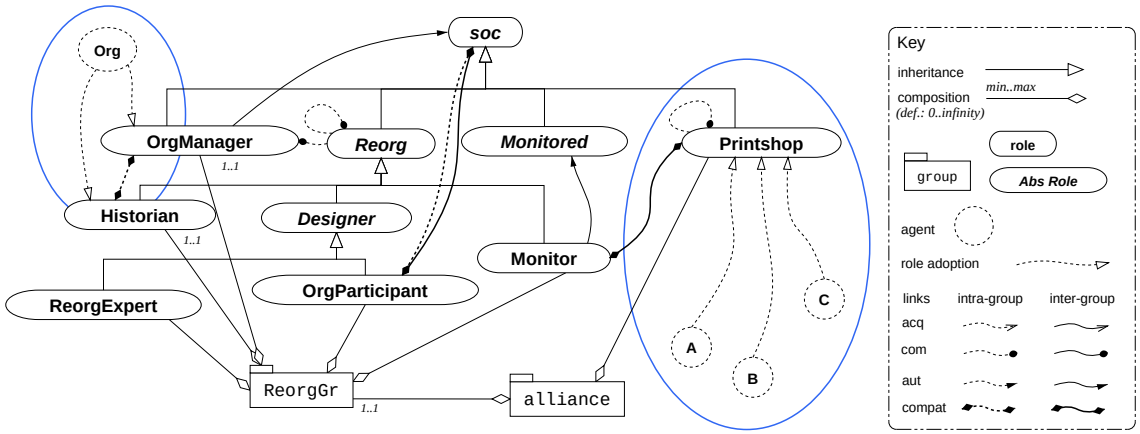

Figure 1. E-Alliance first organisation.

The general description of the not abstract roles and their structural position follows:

1. ***OrgManager***: one, and only one, agent can play this role. This agent has authority on the *soc* agents and so on all agents (this authority will be useful for the execution of the *OrgManager* missions). The agent playing this role have to know the current state of the MAS's organisation and has the permission to change it.
2. ***Historian***: the agent that plays this role maintains the history of the organisation — a kind of information very useful for the monitoring and design phases. Every change (called social event in the sequel) in the organisation (role adoption, commitment with missions, goal achievement, role creation, link creation, change in the cardinalities, etc.) is registered. To be informed of these events, the *Historian* will ask the *OrgManager* to inform him of all social events he has realised. The agent witch adopts this role could be the same that adopts the *OrgManager* role, since they are compatible. In $\mathcal{M}$OISE$^+$, if two roles ρ_1 and ρ_2 are compatible, then some agent playing ρ_1 is allowed to also play ρ_2. If two roles are not compatible, one agent can not play both.
3. ***Monitor***: the agents playing this role will monitor the organisation and identify a situation where the reorganisation must be performed. The inherited communication link to the *Historian* and the authority on the *Monitored* can be used for this aim. In the alliance case, it is added a compatibility links among *Monitor* and *Printshop*, so every Printshop can also be a Monitor.
4. ***ReorgExpert***: the agents playing this role have the ability (and the obligation) to analyse the current organisation, identify their problems, and propose alternatives. These agents must not participate in other groups of this MAS since it is not compatible with any other role . They are invited to participate to this group just for the reorganisation process as a kind of outside analysts which are able to see the organisation from a global point of view.
5. ***OrgParticipant***: each agent who plays a role in the MAS is permitted to play this role since it is compatible with the *soc* role. These agents have practical

knowledge about the way this organisation works. Conversely the *ReorgExpert*, they are inside analysts and see the organisation from a local point of view.

The next two sections describe our proposal by two examples of reorganisation. Hence the reorganisation is a process with four phases (monitoring, design, selection, and implementation), the sections are organised according them.

3. The E-Alliance first reorganisation

Monitoring. Suppose that four agents (named "A", "B", "C", and "Org") adopted the Organisational Specification (OS) described in the Fig. 1 and the agent *A* receives a request to produce 5000 copies of a book for 7/21/2005. The *A*'s scheduler realises that s/he is not able to accomplish this request alone. Of course s/he can use the current OS in order to cooperate with its partners in the alliance. However, once the agents' role (*Printshop*) has no obligation nor constraint, we can easily note that this initial OS is too broad and does not collaborate to *A*'s goal. Thus *A* adopts the role *Monitor* and therefore has the permission to start the monitoring and eventually a reorganisation process.

Design. All the e-alliance members are invited by the *OrgManager* to participate in the reorganisation process as *OrgParticipants*. However, in this example, only the *OrgParticipant A* proposes a new organisation. Its proposal is inside a circle in the partial organisational specification depicted in the Fig. 2, only the most relevant roles are shown in this figure. *A* intends to create a group specialised for the printing books task, the `GrForBook` group. The three roles (*Manager, Printer*, and *Binder*) of this new group and their links form the structure of the organisation, the Structural Specification (SS) in the $\mathcal{M}$OISE$^+$ terminology. This group has the goal of printing books and this task must be done as stated in the scheme *printBook*, i.e., print *Pqty* books and after bind them. A scheme is a global-goal [Castelfranchi 1998] decomposition tree where each goal is assigned to a mission that an agent will commit to. A set of schemes forms the Functional Specification (FS) of the organisation. The relation between the SS and the FS is done by deontic relations, i.e., obligations and permissions of roles on missions. Thus, the Deontic Specification (DS) state, for example, that some agent which accept the *Binder* role is obligated to accomplish the *bind* goal and it can do it only when the *print* goal is already accomplished. Deontic relations also have a penalty. In the *A*'s proposal, in case the *Binder* does not accomplish its mission (*mb*)'s goals, it has to pay $Bcost/4$ to the *Manager*. The penalty for the *Printer* is $Pcost/3$.

After this new OS is implemented, *A* negotiates with other alliance agents in order to convince them to adopt roles and missions in the `GrForBook` (this negotiation is not covered by the reorganisation process which stopped in implementation of the new organisation). As the result of this negotiation, for example, *A* adopts the role *Manager*, and *B* adopts the roles *Printer* and *Binder* (these two roles are compatibles). The agent *A* then instantiates the print books scheme (since s/he has the permission for the scheme's root goal) and the agent *B* is obliged to adopt the missions *mp* (print) and *mb* (bind). *A* also has to assign values to the variables of the new scheme instance: $Pqty = 5000$, $Pcost = 400$, $Bqty = 5000$, Bcost=200, Ptc="start at 5/20/2005, end at 6/18/2005", and *Btc* ="start at 6/20/2005, end at 7/10/2005".

Another example with the same OS may be the case where *A* adopts the role *Manager, B* adopts the roles *Printer* and *Binder*, and *C* adopts the role *Binder*. In this

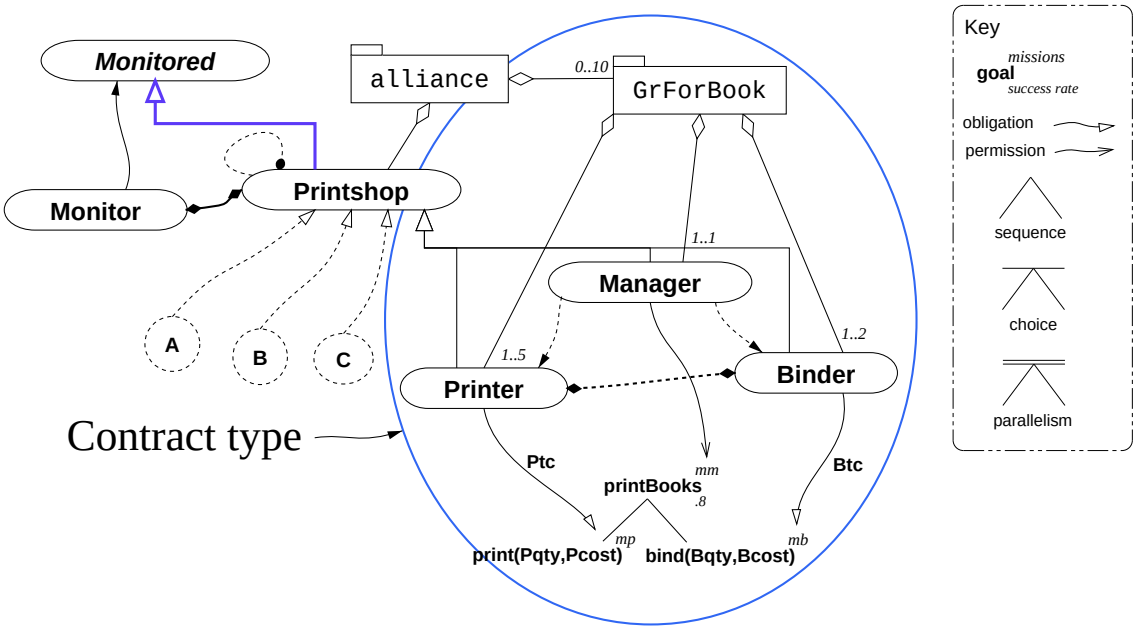

Figure 2. Partial view of the E-Alliance second organisation.

structure, *A* can either decide to create two instances of the print book scheme (see pb^1 and pb^2 in the Tab. 1) or only one scheme (pb^3 in Tab. 1). The commitment to goals like *bind(0,0)* is necessary since the *Binder* is obligated to the mission *mb*.

In the context of the E-Alliance project, the new OS represents a *contract template* that, when instantiated (agents playing roles and missions), states (*i*) the expected behaviour of each part and also (*ii*) the way the contract will be carried out (through the print books scheme). Since there is this contract, the agents will likely spend less time for coordinating their activities.

4. The E-Alliance second reorganisation

Monitoring. In a second scenario, the alliance society has installed a *Monitor* agent. This agent periodically looks the success rate of the print books scheme and, when this rate goes bellow 60% (less than 60% of the scheme execution did not finished with the global goal satisfied), it triggers the reorganisation process.

Design. To design a solution for the low success rate, the *OrgManager* invite two external agents (E_1, E_2) to play the role *ReorgExpert* and the agents *A*, *B*, and *C* to play the role *OrgParticipant*. These agents build the following proposals:

1. The agent E_1, using a diagnostic system, proposes to increase by 30% the penalties of current DS (see Tab. 2). This proposal's focus is on the DS. The argument is that the print book scheme does not work because the *Printer* and *Binder* give their missions up.
2. The agent E_2, searching in an OS database, proposes a completely new hierarchical OS (see Fig. 3). This proposal's focus is on all the OS. The argument is that the *Printer* and *Binder* are permitted to communicate with each other and that the *Manager* has few control on the scheme performance.

Table 1. Example of adoption of roles, missions, and goals.

agent	role	mission	goal
		scheme pb^1	
A	*Manager*	*mm*	
B	*Printer*	*mp*(Ptc="start at 6/1/2005, end at 6/30/2005")	*print(2000,300)*
B	*Binder*	*mb*(Btc="start at 6/30/2005, end at 7/10/2005")	*bind(2000,100)*
C	*Binder*	*mb*	*bind(0,0)*
		scheme pb^2	
A	*Manager*	*mm*	
B	*Printer*	*mp*(Ptc="start at 5/20/2005, end at 6/18/2005")	*print(3000,400)*
B	*Binder*	*mb*	*bind(0,0)*
C	*Binder*	*mb*(Btc="start at 6/20/2005, end at7/10/2005")	*bind(3000,150)*
		scheme pb^3	
A	*Manager*	*mm*	
B	*Printer*	*mp*(Ptc="start at 5/20/2005, end at 6/18/2005")	*print(5000,700)*
B	*Binder*	*mb*(Btc="start at 6/30/2005, end at 7/10/2005")	*bind(2000,100)*
C	*Binder*	*mb*(Btc="start at 6/20/2005, end at 7/10/2005")	*bind(3000,150)*

Table 2. E_1**'s modification proposal on the DS.**

role	relation type	mission	time	penalty
Manager	*permission*	*mm*	*Any*	—
Printer	*obligation*	*mp*	*Ptc*	$(Pcost/3) * 1.3$
Binder	*obligation*	*mb*	*Btc*	$(Bcost/4) * 1.3$

3. The agent *A*, through its experience in the print book scheme, proposes to increase the number of allowed binders (from 1..2 to 1..5) in the `GrForBook`. This proposal's focus is on the SS, more precisely, on the binder role.
4. The agent *B*, through its experience as *Printer*, proposes to increase the details of the print book scheme (see Fig. 4). This proposal's focus is on the FS. The argument is that the *Manager* needs to better know how the *Printer* and *Binder* are performing their tasks.

Selection. Since we have four proposals, we will firstly describe a voting system used to select a proposal (it is based on an early developed selection scheme [Lugo et al. 2001]). The execution of this reorganisation phase uses the organisation history maintained by the *Historian.* The first OS of one MAS is called os_1, the second os_2, and so forth. Each reorganisation process changes the current OS version os_v to os_{v+1}.

Table 3. Proposals classification.

proposal	focus	*position*	*fe*	*da*	*cost*
E_1	DS	0	0.5*0+0+0=0	0	2*(0+0)=0
E_2	OS	-18	0.5*0+0+0=0	0	2*(6+3)=18
A	SS, GrForBook	12.15	0.5*1+4+0=4.5	7.65	2*(0+0)=0
B	DS, printBook	8	0.5*0+0+10=10	0	2*(0+1)=2
no change		0	0	0	0

(a) SS (b) FS

role	relation type	mission	time	penalty
Master	*permission*	*mm*	*Any*	—
Slave	*obligation*	*mp*, *mb*	*TC*	*Pcost*/2

(c) DS

Figure 3. E_2**'s proposal.**

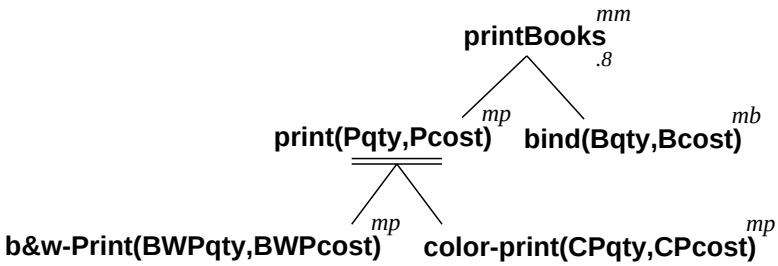

Figure 4. B**'s modification proposal on the FS.**

The first goal is to classify the proposals. It is accomplished by the *OrgManager* using three values: the experience of the proposer in its proposal's focus (SS, FS, or DS), the success of its previously chosen proposals, and the cost of the proposal.

The *focal experience* of an agent *a* in an os_v considering its proposal *p* is given by

$$fe(a,p,os_v) = \begin{cases} 0 & \text{if } v = 0 \\ (\theta\ fe(a,p,os_{v-1}))+ & \text{if } v \neq 0 \\ fre(a,p,os_v)+ & \\ fme(a,p,os_v) & \end{cases} \quad (1)$$

where *a* is an agent that has participated in the *os* in some of its version, from os_1 to os_v; θ represents the relevance of participation in old versions of os_v. For example, if θ is 1, the experience in old version has the same value than experiences in the version *v*; if θ is 0, old experiences has no value. $fre(a,p,os_v)$ gives the number of roles the agent *a* played in the system when the OS version is os_v. This function considers the proposal's focus: if the focus is a role, the function *fre* counts only the times *a* played this role; if the focus is a group, the *fre* counts the times this group's roles are played by *a*; if the focus is the

DS, *fre* counts the times *a* played the role of the deontic relation; and finally, if the focus is in the FS, $fre = 0$. $fme(a, p, os_v)$ gives the number of missions *a* committed to when the OS version is os_v. Analogously to *fre*, this function also considers the plan's focus. Usually, the experience of the *ReorgExpert* agents is zero since they do not participate in the alliance. The exception is the case where the very reorganisation process is being revised.

The *diagnostic ability* of an agent *a* is given by

$$da(a) = \sum_{os_v \in OS_a} \epsilon(startTime(os_v) - endTime(os_v)) \tag{2}$$

where *a* is an agent that proposed a modification in the organisation and its proposal was implemented; the set OS_a contains all OSs *a* has built, i.e., the OSs after the implementation of a *a*'s proposal; ϵ represents the relevance of the period that some OS has been active (from its implementation — *startTime* — to the moment it is subject of the reorganisation process — *endTime*). Thus, the diagnostic ability on an agent is based on the amount of time its proposed OSs was active.

The *cost* of one proposal *p* is given by

$$cost(p) = \phi \quad (rolePlayersRemoved(p) + missionPlayersRemoved(p)) \tag{3}$$

where ϕ is the relevance of the cost for the MAS; $rolePlayersRemoved(p)$ gets the number of roles that agents will lose in case *p* will be implemented; $missionPlayersRemoved(p)$ gets the number of missions that agents will lose in case *p* will be implemented. Thus, the cost essentially measures the problem that the implementation of a proposal will cause on the agents. The inclusion of a new role, for instance, has no cost.

Finally, the classification of the alternatives is done in accordance to the value given by the following function:

$$position(p, a, os_v) = fe(a, p, os_v) + da(a) - cost(p) \tag{4}$$

In the alliance example, to classify the proposals the *OrgManager* set $\theta = 0.5$ (participation in old versions has the half value of the experiences in the current version of the OS — os_2), $\epsilon = 0.05$ (each day counts as 0.05), and $\phi = 2$ (the cost of each change goal in the current OS is 2). The proposals listed in Tab. 3 considers that the *Historian* has registered the following OE history:

1. os_1 (Fig. 1) started at 2/1/2005.
2. the agents *A*, *B*, and *C* adopted the role *Printshop*.
3. os_1 ended at 5/1/2005 when the agent *A* had proposed a new OS (Sec. 3.).
4. os_2 (Fig. 2) started at 5/1/2005. Since this OS version was built on the previous version without extinguishing it, the agents *A*, *B*, and *C* has maintained their *Printshop* role.
5. *A* as *Manager*, *B* as *Printer*, and *B* as *Binder* performed the print book scheme twice.
6. *B* as *Manager*, *A* as *Printer*, and *C* as *Binder* performed the print book scheme once.

Table 4. Election results.

proposal	votes	voter's *ge*	voter's *da*	(*ge*+*da*+1)
B	E_1	0	0	1.00
	E_2	0	0	1.00
	B	0.5*1+7+10	0	18.50
	C	0.5*1+4+4	0	9.50
	total			*30.00*
A	*A*	0.5*1+5+6	7.65	20.15
	total			*20.15*

7. *A* as *Manager*, *B* as *Printer* and *Binder*, and *C* as *Binder* performed the print book scheme twice.
8. *C* as *Manager*, *A* as *Printer*, and *B* as *Binder* are performing the print book scheme.
9. os_2 ends at 10/1/2005 when the reorganisation process begun. This OS was active for 153 days.

In the next step, the *OrgManager* collects the proposals, adds the "no change" proposal, classifies and sends them to the *ReorgExpert* and the *OrgParticipant* agents. Each agent may vote on one proposal. The final score of each proposal *p* is given by

$$votes(p) = \sum_{a \in E(p)} ge(a, os_v) + da(a) + 1 \tag{5}$$

where $E(p)$ is the set of agents which vote in *p* and *ge* (general experience) is given by

$$ge(a, os_v) = \begin{cases} 0 & \text{if } v = 0 \\ \theta\ ge(a, os_{v-1}) + & \text{if } v \neq 0 \\ re(a, os_v) + me(a, os_v) & \end{cases} \tag{6}$$

where $re(a, os_v)$ gives the number of roles the agent *a* played in the system when the OS version is os_v; and $me(a, os_v)$ gives the number of missions *a* committed to when the OS version is os_v. Notice that the experience and the diagnostic ability of the voter increases the value of its vote. The proposal with the greatest $votes(p)$ value will be selected and implemented by the *OrgManager*.

In the alliance example, the result of the "election" are shown in Tab. 4. Thus the *B*'s proposal will be implemented by the *OrgManager*.

5. Conclusions

This paper has presented a general view of the reorganisation problem under the $\mathcal{M}$OISE$^+$ point of view applied to virtual alliances. The main contributions are (*i*) the proposal to use an organisational model to represent the contract, while being a set of constraints, among the enterprises and (*ii*) to use a reorganisation process to model the dynamics in the alliance. In the latter contribution, we proposed a voting system to deal with the selection phase of the reorganisation. This voting system considers the alliance history to measure each agent expertise both in the alliance and in the reorganisation processes.

The $\mathcal{M}$OISE$^+$ organisational model has been shown as a good support for the specification of an MAS's organisation which intends to reorganise itself because (*i*), as

an organisational description, it gives useful information for the monitoring and design phases and (*ii*), as a specification tool, it allows us to define the reorganisation process with valuable properties: (*a*) the openness for many types of monitoring, design, and selection; (*b*) the definition of special roles like the *OrgManager* and *Monitored*; and (*c*) the specification of the reorganisation through the $\mathcal{M}$OISE$^+$ enable any $\mathcal{M}$OISE$^+$ agent to understand and participate in the reorganisation. The $\mathcal{M}$OISE$^+$ is already implemented and available at `http://www.lti.pcs.usp.br/moise`.

References

Andreoli, J. M., Castellani, S., and Munier, M. (2000). AllianceNet: Information sharing, negotiation and decision-making for distributed organizations. In *Proceedings of the First International Conference on Electronic Commerce and Web (CEcWeb'2000)*.

Carron, T. and Boissier, O. (2001). Towards a temporal organizational structure language for dynamic multi-agent systems. In *Pre-Proceeding of the 10th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW'2001)*.

Castelfranchi, C. (1998). Modeling social action for AI agents. *Artificial Intelligence*, (103):157–182.

de Almeida Júdice Gamito Dignum, M. V. F. (2003). *A model for organizational interaction: based on agents, founded in logic*. PhD thesis, Universiteit Utrecht.

Dignum, V. and Dignum, F. (2001). Modelling agent societies: Co-ordination frameworks and institutions. In Brazdil, P. and Jorge, A., editors, *Proceedings of the 10th Portuguese Conference on Artificial Intelligence (EPIA'01)*, LNAI 2258, pages 191–204, Berlin. Springer.

Hübner, J. F. (2003). *Um Modelo de Reorganização de Sistemas Multiagentes*. PhD thesis, Universidade de São Paulo, Escola Politécnica. `http://www.inf.furb.br/~jomi/pubs/2003/Hubner-tese.pdf`.

Hübner, J. F., Sichman, J. S., and Boissier, O. (2002). A model for the structural, functional, and deontic specification of organizations in multiagent systems. In Bittencourt, G. and Ramalho, G. L., editors, *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence (SBIA'02)*, LNAI 2507, pages 118–128, Berlin. Springer.

Hübner, J. F., Sichman, J. S., and Boissier, O. (2004). Using the $\mathcal{M}$OISE$^+$ for a cooperative framework of MAS reorganisation. In Bazzan, A. L. C. and Labidi, S., editors, *Proceedings of the 17th Brazilian Symposium on Artificial Intelligence (SBIA'04)*, LNAI 3171, pages 506–515, Berlin. Springer.

Lugo, G. G., Hübner, J. F., and Sichman, J. S. (2001). Representação e evolução de esquemas sociais em sistemas multi-agentes: Um enfoque funcional. In Martins, A. T. and Borges, D. L., editors, *Anais do III Encontro Nacional de Inteligência Artificial*, pages 1237–1246, Fortaleza, Brazil. SBC.

Sierra, C., Rodríguez-Aguilar, J. A., Noriega, P., Esteva, M., and Arcos, J. L. (2004). Engineering multi-agent systems as electronic institutions. *European Journal for the Informatics Professional*, V(4).