Using a normative organisational model to specify and manage an institution for multi-agent systems

Benjamin Gâteau [a]

[a] CITI/CRP Henri Tudor – G.-D. of Luxembourg
benjamin.gateau@tudor.lu

Abstract

Nowadays multi-agent applications are more and more openness. That brings the risk to deal with to autonomous agents i.e. agents not respecting the society rules. To insure a coherent behaviour the application require tools to control and regulate the system overall functioning. Moreover they should provide the system with mechanisms to enforce global laws on the autonomous agents operating in it. This paper presents an institution multi-agent layer called $\mathcal{S}$YNAI. Implemented with different agents, the institution functioning is itself specified as a normative organisation model making explicit how the overall system should be controlled. Using an iTV game application, we illustrate how such a specification is useful to help the agents to function in the system.

1 Introduction

Nowadays, multi-agent technologies' applications are faced to an increasing openness. Being composed of heterogeneous and autonomous agents, they require tools to control and regulate the system overall functioning. Moreover they should provide the system with mechanisms to enforce global laws on the autonomous agents operating in the system.

In this paper we present $\mathcal{S}$YNAI, a multi-agent layer dedicated to the rights and duties management and enforcement of autonomous agents within an organisation. This layer belongs to the electronic institution environment called MA$\mathcal{B}_{\text{ELI}}$[1]. It is composed of generic institutional agents, *supervisors*, aiming at controlling and enforcing the domain agents functioning according to the specified normative organisation expressed with the normative organisation model called $\mathcal{M}$OISE^{Inst}. Whereas supervisor agents are dedicated to the system control, the domain agents implement the application functionalities. The supervisor agents themselves operate under the control of a normative organisation that structures and constrains their control behaviour on the domain agents.

All along the paper, we illustrate the use of MA$\mathcal{B}_{\text{ELI}}$ with an iTV game issued from the European ITEA Jules Verne Project. We show how this multimedia game can be modelised and controlled with such a platform.

This paper is organised as follows: section 2 presents an motivations overview for using an explicit normative organisational model to specify $\mathcal{S}$YNAI, the multi-agent institution platform with $\mathcal{M}$OISE^{Inst}, normative organisation model. Its use is illustrated with the iTV application. The succeeding sections present the other main component of the Electronic Institution namely $\mathcal{S}$YNAI, the arbitration system. We describe the institutional agents organisation and how the supervisors arbitrate an organisation by respecting their own organisation specification Finally, before concluding, section 5 compares our work to other approaches.

[1] Multi-Agent Based ELectronic Institution

2 Motivations

In the recent past, multi-agent technologies have been developed and deployed in different applications. Most of these efforts have been largely supported by the existence of multi-agent platforms like JADE [1] or FIPA-OS [12]. These platforms have demonstrated the generic services needs and utility for supporting the execution of multi-agent applications such as Agent Management System, Directory Facilitator. The recent developments in the domain (e.g. electronic commerce [2]) have shown the requirement to enrich those services to provide multi-agent applications with institution platforms. The main purpose was to insure and promote the user's trust in the system functioning by controlling agents during their transactions. In human societies institutions define the *game rules* [11]. These rules enclose all kinds of informal or formal constraints that human beings use to interact. Current multi-agent approaches to institution propose these rules modelling through normative systems [10] that are interpreted by agents that enforce the application's agents to follow them and not to violate them.

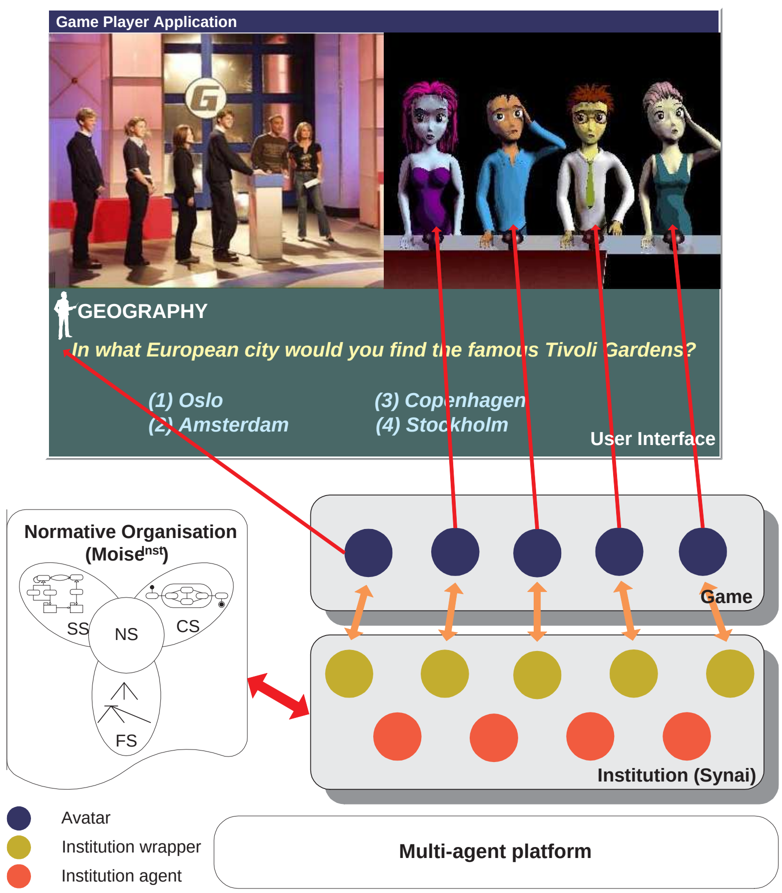


Figure 1: Global view of the e-Institution for iTV

In the same trend, the work described in this paper is applied to an Interactive Games application (see Fig. 1): a "questions – answers" TV game show opposing a real players' team present on the TV scene, to a televiewers' team interacting from home into the game with the help of the Avatars, i.e. software agents. Each Avatar is under its respective televiewer control. The quizmaster is also supported by a virtual assistant. His role is to regulate the game. As in all collective games, the aim is to promote a collective behaviour among the the same team's players. An explicit *organisation* states the roles involved in the game. A set of rules (*norms*) represents the game rules, the sanctions and rewards in use during the game. However, since avatars are autonomous agents, they can be autonomous with respect to these constraints, e.g. a televiewer is able to decide to answer whereas it is not his turn and to take the risk to be punished. An *institution* has been thus defined in order to control, regulate and reward or punish agents when they respect or not the

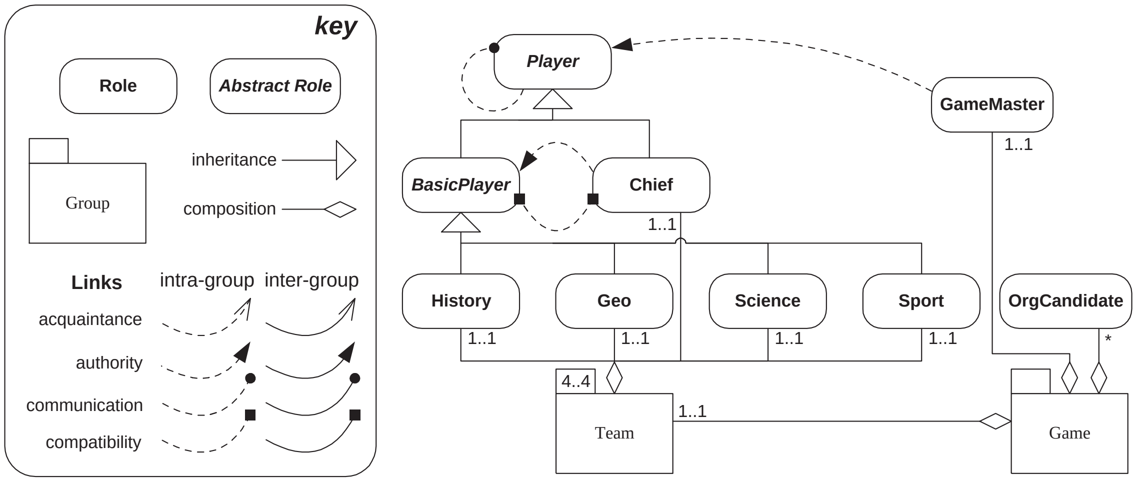


Figure 2: Avatars scenario Structural Specification

organisation and norms.

Two kinds of agents have been designed: *domain* agents, avatars controlled by the users, and *supervisor* agents aiming at managing the organisation and enforcing the game rules on the domain agents. They are organised into two layers: (i) the multiagent interactive game in which domain agents as avatars, operate on behalf of their user, (ii) $\mathcal{S}$YNAI[2] an institution multiagent platform dedicated to the organisation management and to its control by the mean of the supervisor agents. Both kinds of agents (supervisor and domain) are organised and constrained according to a normative organisation described with the $\mathcal{M}\text{OISE}^{Inst}$ normative organisation description language [5]. Agents are thus able to reason on the organisation and constraints. They have the possibility to decide to take it into account or not. The institution platform reads this specification in order to supervise and control the agents as well as be informed about its own organisation specification.

Before focusing on the presentation of the $\mathcal{S}$YNAI specification, we shortly describe $\mathcal{M}\text{OISE}^{Inst}$.

3 Normative organisation description language

$\mathcal{M}\text{OISE}^{Inst}$ [5, 6] is used to define what we call an organisation specification (OS) with the help of four specifications[3]: structural specification (SS), functional specification (FS), contextual specification (CS) and normative specification (NS).

3.1 Structural specification

The *structural specification* (SS) expresses a set of roles, groups and links that build the organisation structure (cf. Figure 2). For instance, a "Team" group is composed of the following roles: "History", "Geo", "Sport", "Science" and "Chief". These roles inherit from "BasicPlayer" or "Player" roles that are abstract, i.e. roles which are not adoptable by agents. *Cardinality* and *compatibility links* express constraints on the way agents play roles in groups. For instance, cardinality '1..1' on the composition link ensures constraints that, in a "Team" group instance, roles can be adopted by only one agent at the same time. A compatibility link between "BasicPlayer" and "Chief", allows the same agent to play those two roles or those roles specializations. Thus, according to this specification, one agent may have the possibility to play at most two of those five roles. In order to avoid that five agents play the five "Team" roles, we express a cardinality '4..4' for the group "Team", stating that any well formed instance of this group may contain four and only four agents.

[2]**SY**stem of **N**ormative **A**gents for **I**nstitution.

[3]A BNF definition of SS and FS are available in [7] and of CS and NS in [5]

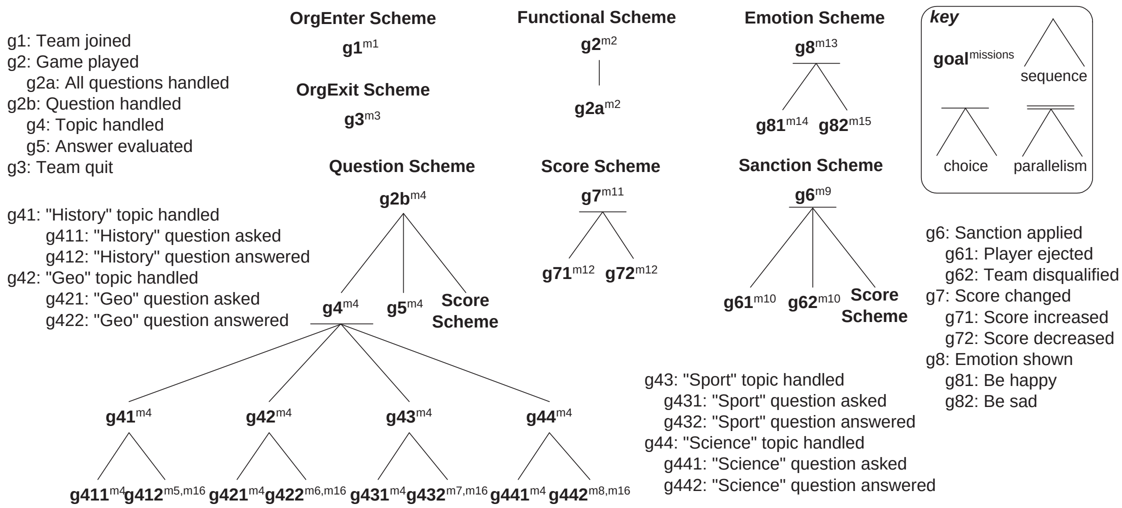


Figure 3: Avatars scenario Functional Specification

Communication and authority links structure the different roles. For instance, all roles inheriting from "Player" can communicate between them, and the "Chief" has the authority on all "BasicPlayer", which means that all roles inheriting from this role are under the "Chief" authority . "OrgCandidate" is the first role played by every agents coming in the organisation that is why it could be played by a lot of agents at the same time. "OrgCandidate" does not participate in the game (activity to answer question). According to available roles adoptable in the "Team", agents could change to join the group. "GameMaster" is the role played by the only one presenter assistant.

3.2 Functional specification

The *functional specification* (FS) specifies the global expected system functioning in terms of goals/subgoals that agents operating in it should achieve (cf. Figure 3). The goal decomposition trees are organised into different *social schemes* which may be reused within other social schemes. For instance the Question Scheme has "question handled" as root goal and its plan is a sequential achievement of goals "g4", "g5" and of "Score Scheme". The "OrgEnter Scheme" (resp. "OrgExit Scheme") defines the principal behaviours for entering (resp. leaving) an organisation. We also define a scheme relating to the customization of the sanctions by specifying that apply a sanction is a choice between the ejection of a player, the disqualification of the team or the modification of the score. At last, we can also define scheme relating to Avatars 3D rendering with goals to show a happy or sad face for instance making possible the norms definition relating to that.

3.3 Contextual specification

To tackle with the applications situatedness in evolving environment, a *contextual specification* (CS) captures design-time constraints on the organisation evolution as a set of contexts and transitions between them (cf. Figure 4).

A context expresses a state in which an agent playing a role has to respect specific rules (see below the norms expression). Transitions define change from one context to another context given different events occurrence. For instance, in our application, it is used to express the different game rounds that impose change to the rules. Here the CS starts with a synchronous state "Begin" which allows the televiewer to connect to the system. A macro-context "Game" is decomposed into three rounds sub-contexts. This global context will be used to define the basic game rules while the three round sub-contexts will be used to define the corresponding specific rules. The "Game" context is also decomposed into two sub-contexts defining the players turn. A round sub-context and a turn sub-context can be active at the same time. Let us notice that the macro-context is active in all its sub-contexts. The rules defined in the "Game" context are thus inherited in sub-contexts and are still valid. Finally the last state is the context in which Avatars quit their team.

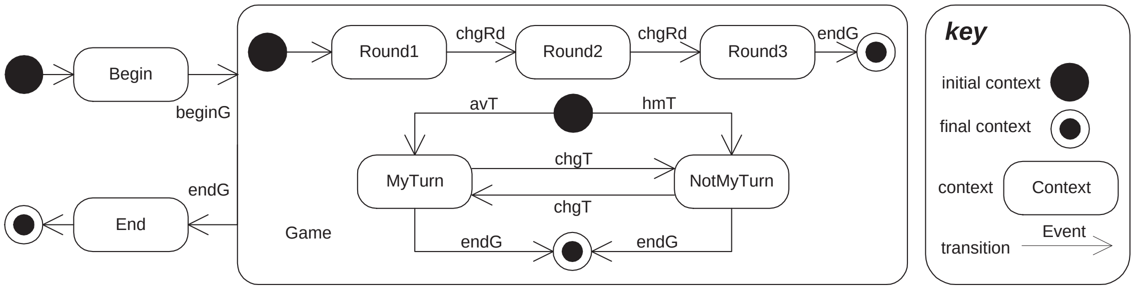


Figure 4: Avatars scenario Contextual Specification

3.4 Normative specification

Finally, the *normative specification* (NS) glues all specifications in a coherent and normative organisation. It expresses permissions, obligations and prohibitions of missions referring to the goals of the FS in the context of elements of the SS (roles or groups). Missions group goals into coherent sets according to the way the designer wants to assign them to roles or groups for their achievement. A norm in NS (cf. Figure 5) is specified with an *id*, a *context*, a *bearer*, a *deontic operator* referring to a *mission* and a *deadline*.

context	id	w.	condition	issuer	bearer	deOp	mission	deadline	sanction
Begin	N01	1	nb(Team)<max(Team)	Supervisor	OrgCandidate	O	m1	---	---
End	N02	1	---	Supervisor	Team	O	m3	---	---
Game	N03	1	---	Supervisor	OrgCandidate	F	m1	---	N17
Game	N04	1	---	Supervisor	Team	F	m3	---	---
Game	N05	1	---	Supervisor	GameMaster	O	m2	---	---
Game	N06	1	---	Supervisor	GameMaster	O	m4	---	---
Game	N07	1	---	Supervisor	Team	P	m13	---	---
Game	N08	2	---	Supervisor	Team	F	m16	---	N18
Round1	N09	3	---	Supervisor	Team	P	m16	< answer_delay	---
Round2	N11	1	---	Supervisor	History	P	m5	< answer_delay	---
Round2	N12	1	---	Supervisor	Geo	P	m6	< answer_delay	---
Round2	N13	1	---	Supervisor	Sport	P	m7	< answer_delay	---
Round2	N14	3	---	Supervisor	Science	P	m8	< answer_delay	---
Round3	N10	1	---	Supervisor	Chief	P	m16	< answer_delay	---
NotMyTurn	N15	1	---	Supervisor	Team	F	m16	---	---
NotMyTurn	N16	1	---	Supervisor	Team	F	m14	---	---
Game	N17	1	violated(N02)	Supervisor	GameMaster	O	m9	---	---
Game	N18	1	violated(N08)	Supervisor	GameMaster	O	m11	---	---

Figure 5: Avatars scenario Normative Specification

The Avatars scenario NS displayed on the Fig. 5 uses functions defined in the $\mathcal{M}$OISE^{Inst} meta-model. The N1 validation condition ($nb(Team) < max(Team)$) is composed of two functions representing the agent number already in the Team group and the maximum of agents allowed in the Team. This norm expresses the fact that the Team must not be full in order to allow an agent entry. Concerning the norms N17 and N18, the function $violated()$ return $true$ if the norm in parameter is not respected. The detection is done by $\mathcal{S}$YNAI agents. We explain how in section 4. This specification can define norms as well as their sanction. A sanction is a norm with a violation condition. The norms issuer is the role which supervises the norm respect. Users who specify their own application modelling do not know how the arbitration works. That is why they have to set the issuer up to "Supervisor" role. The $\mathcal{S}$YNAI layer decides automatically what agents supervise what norms.

Our model does not provide solution to check if the norms are coherent ones compared to the others. In our example, potential conflicts can occur between norms N15 and N09 because they oblige agents playing role in the "Team" group to accomplish and to not accomplish the mission m16. We have the same between norms N15 and N14, N8 and N9, and N8 and N14. To avoid agents having to make a choice between conflictual norms to respect, we specify a priority order denoted by a *w.* in the table. 1 is the higher priority. To abrogate conflictual norms we decrease N9 and N14 priority order.

The norms allow us to define and constrain the game functioning as well as what happens at the beginning and at the end of the game. The four first norms in Fig. 5 define when it is possible to join and to leave the team. Global game rules are expressed as functioning norms. For instance *Prohibition for "Player" role to answer a question during the game* represented by N08 authorizes concerned roles during rounds to answer questions. N09 and N14 oblige the "Player" and the "Chief" roles to answer all questions during the first and third rounds. Four norms for each role in the second round allow concerned roles to answer question.

4 $\mathcal{S}$YNAI Institution platform normative organisation

4.1 Requirements

Domain agents play the game by acting in an Organisation specified by the designer in the OS described in the previous section. As depicted on the left of Fig. 6, an Organisation is an Organisation Specification instantiation which means that agents adopt roles and commit on mission according to the OS. This specification aims at constraining their behaviour. However being autonomous (under an user control) they can decide to not respect the specification stated in the OS. An agent can adopt a role in the Organisation which is not authorize in the OS. If the Organisation is not consistent with the OS, the Organisation is considered as incoherent.

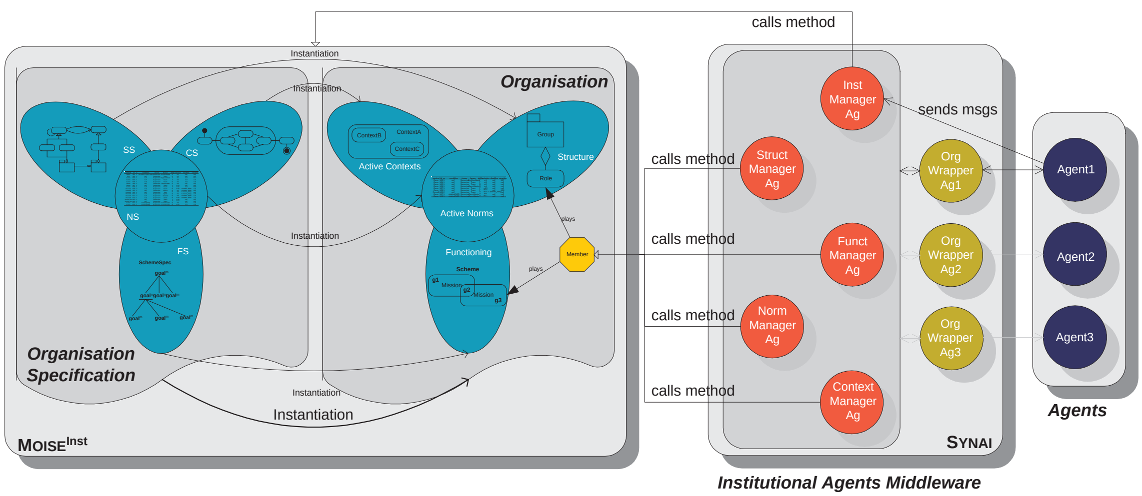


Figure 6: Supervisor agents in $\mathcal{S}$YNAI

As motivated in this paper beginning, we need an arbitration system able to supervise the Organisation execution and avoid incoherences. That means managing and controlling the functioning of the Organisation by the way of different events corresponding to the agents entry/exit, roles adoption/leaving, context change, missions commitment, goals achievement, etc. Event are basic $\mathcal{M}$OISE^{Inst} elements. They are defined in the $\mathcal{M}$OISE^{Inst} meta-model and in the user CS model.

To satisfy these requirements, we have defined an institution layer namely $\mathcal{S}$YNAI which filters actions executed in the Organisation by the agents. It aims at checking the OS respect. Receiving requests from agents (messages composed by an event and others parameters), it detects if they violate or not constraints stated in SS, FS and NS (cf. Fig. 6). For instance it verifies that an agent plays compatible roles or that it is authorized to commit on mission according to the role it is playing.

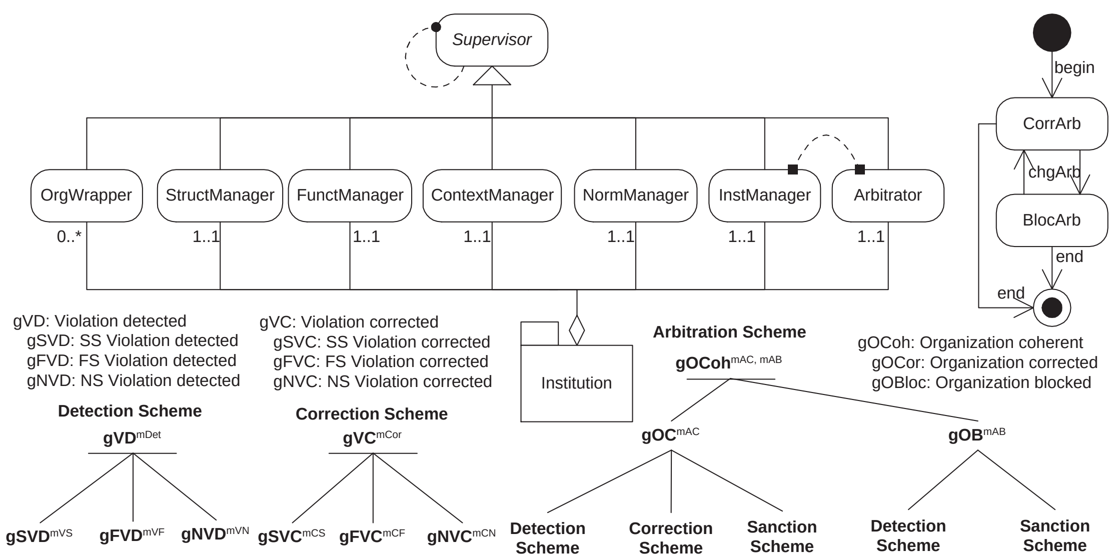
A set of different supervisor agents composes $\mathcal{S}$YNAI. Four different agents manage each entity deriving from the OS specification: *StructManagerAg* for the SS entity, *FunctManagerAg* for the FS entity, *Con-*

textManagerAg for the CS entity and *NormManagerAg* for the NS entity. The *InstManagerAg* is able to manage the Organisation. Each domain agent is supported by an *OrgWrapperAg* which is a facilitator for the domain agent to access and interact with the supervisor agents. $\mathcal{S}$YNAI agents are sensitive to events and treat them differently according to their role. The interpretation of an event coming with a message triggers action and another message sending.

4.2 Institution normative organisation

In order to supervise the organisation and the norms respect, supervisor agents have to understand the $\mathcal{M}$OISE^{Inst} model. This model advantage is the ability to model both the organisation and its arbitration. Supervisor agents are organised the same way as domain agents i.e. according to the OS specified with $\mathcal{M}$OISE^{Inst} in order to structure and to define their rights and duties (see Fig. 7).

Structural specification : The SS is composed of the only group “Institution” grouping the roles that supervisor agents would play in order to manage $\mathcal{M}$OISE^{Inst} model specifications for domain agents. “InstManager” and “Arbitrator” roles are compatible. All roles inheriting from “Supervisor” role can communicate with each other (communication link from “Supervisor” to itself). The cardinality ‘1..1’ except for “OrgWrapper” ensures that only and only one supervisor agent with play a role in this group.



context	*id*	*w.*	*condition*	*issuer*	*bearer*	*deOp*	*mission*	*deadline*	*sanction*
CorrArb	NA01	1	---	InstManager	Arbitrator	O	mAC	---	---
CorrArb	NA02	1	---	Supervisor	Arbitrator	O	mCor	---	---
CorrArb	NA03	1	---	Supervisor	StructManager	O	mCS	---	---
CorrArb	NA04	1	---	Supervisor	FunctManager	O	mCF	---	---
CorrArb	NA05	1	---	Supervisor	NormManager	O	mCN	---	---
BlocArb	NA06	1	---	Supervisor	Arbitrator	O	mAB	---	---
---	NA07	1	---	Supervisor	Arbitrator	O	mDet	---	---
---	NA08	1	---	Supervisor	StructManager	O	mVS	---	---
---	NA09	1	---	Supervisor	FunctManager	O	mVF	---	---
---	NA10	1	---	Supervisor	NormManager	O	mVN	---	---

Figure 7: Organisation Specification of $\mathcal{S}$YNAI

Functional specification : The FS defines the arbitration system main goal which is to keep the organisation in a coherent state. This consists in a choice between correcting the violation (*gOC* goal) or blocking the violation intention (*gOB* goal). This choice defines an arbitration strategy. As expressed in the arbitration scheme, the arbitration steps are: violation detection, violation correction or not (according to the arbitration strategy) and culprit sanction. Constraints come from the SS (cardinalities, links, etc.), from the FS (mission cardinalities) and from the NS (norms). The CS does not constraint agents and so cannot be violated. A violation detection is either a NS violation, or a FS violation or a NS violation. The *gVD* goal ("Violation Detected") plan expresses that. The same choice defines the *gVC* ("Violation Corrected") plan.

Contextual specification : The CS defines the contexts that are used for the arbitration strategies choice related to the goals *gOC* or *gOB* achievement. During the Organisation working, an event can be created and causing the arbitration strategy change: correct violations or block violations.

Normative specification : The NS norms (cf. the table of the Fig. 7) express that the organisation must be kept in a coherent state by correcting violations in the "CorrArb" context (NA1 to NA5) and by blocking actions with violation intention in the "BlocArb" context (NA6) and express that the detection must be done in whatever context (NA7 to NA10).

InstManagerAg plays "InstManager" and "Arbitrator". Each supervisor agent plays the role corresponding to its capabilities: *StructManagerAg* plays "StructManager", *FunctManagerAg* plays "FunctionalManager" and so on.

4.3 Detection of constraint violation

In order to supervise an Organisation execution, the $\mathcal{S}$YNAI's agents have to respect their own Organisation Specification and achieved their goals. As seen before, their root goal is to keep the organisation in a coherent state. For that and first of all they have to detect violations. According to events contained in message, institutional agents behave in a certain way by executing some actions and sending messages. The Fig. 8 depicts the interaction diagram between $\mathcal{S}$YNAI agents in order to treat a violation detection.

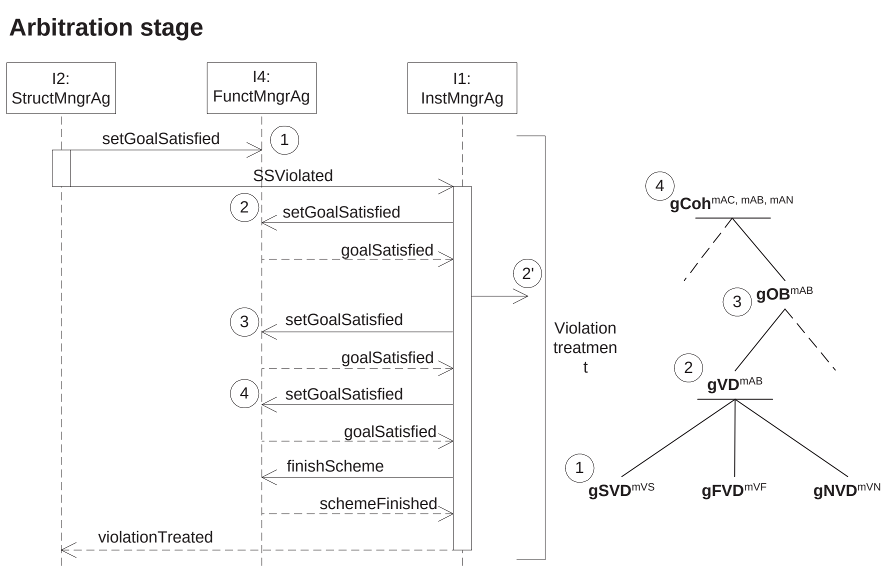


Figure 8: Interaction diagram between $\mathcal{S}$YNAI agents for a violation treatment

We consider here that a structural violation is in progress. So when *StructManagerAg* detects a violation, it achieves a goal and for that it acts on the Organisation via *FunctManagerAg* by sending a message with *setGoalSatisfied* event and goal "gSVD" as parameter (step 1). At the same time, it notifies *InstManagerAg* that a violation happened. *InstManagerAg* considers that goal "gVD" is achieved as the plan is executed (step 2). In order to accomplish mission *mAB* a sanction must be applied. Step 2 is the creation and the execution of a Sanction Scheme by *InstManagerAg* playing "Arbitrator" role. Then *InstManagerAg* is allowed to achieve goal "gCB" because of violation detection and sanction. At last goal "gCoh" is also achieved because it corresponds to the arbitration strategy choice (step 4). Therefore mission *mAB* is accomplished and Arbitration Scheme is finished (*finishScheme* event). The arbitration is terminated for this violation and *InstManagerAg* informs the supervisor which detected the violation by sending *violationTreated* event.

5 Related work

In this paper we introduced $\mathcal{S}$YNAI that could be compared with others Electronic Institution arbitration models coming from the MAS domain.

OMNI (Organizational Model for Normative Institutions) [13] is an organisational model split in several dimensions (normative, organisational and ontological) and levels (abstract, concrete and implementation). The concrete organisational model is composed of roles and interactions structures implemented into social model specifying roles played by agents and into interactions model specifying the actual interactions between agents. Norms used to specify roles and interactions structures are defined within the normative dimension. OMNI differentiates institutional agents bringing particular services (matchmaking, reputation, identification, notary, monitoring, etc.) in order to execute institutions and external agents obliged to respect organisational and normative constraints. *Police Agents* are in charge of norms execution and violation detection.

ISLANDER is an Institution Definition Language (IDL) [3] specifying scenes and protocols in an Electronic Institution. Compared to $\mathcal{M}$OISE^{Inst} the role hierarchy specification is minimal in the sense that we can only define roles and inheritance and compatibility between roles. The agents functioning definition is not possible. This model is more focused on interactions and protocols specification taking part to the scenes definition. The agents have to follow the protocols to evolve in a scene. There are no sanctions defined. AMELI [4] is the ISLANDER specification execution framework. It provides a social layer which controls and helps the agents to participate in an e-institution with specialized governors. According to the specification available, only the interactions between agents can be controlled.

$\mathcal{M}$OISE^{+} [8] is an organisational model specifying agents' structure, functioning and set of deontic expressions. Its separation into three distinct specifications brings more flexibility. This model allows us to define well-structured and precise organisations. However there is no contexts or scene definition in which specific deontic expressions can be applied. $\mathcal{S}-\mathcal{M}$OISE^{+} [9] is a platform managing $\mathcal{M}$OISE^{+} organisations. It provides to agents evolving in the society personal "OrgBoxes" as organisation partial view. It serves as interface between heterogeneous agents and the organisation. Even so, there is just one "OrgManager" for controlling agents access into the organisation. Besides, the deontic expressions are enforced but not controlled. For instance, an obligation violation is hardly detectable.

To conclude, contrary to $\mathcal{M}$OISE^{Inst}, none of these models take into consideration the whole essential specification points of view (structural, functional, contextual and normative). They allow an arbitration system modelling. Arbitration could be done if norms can be controlled. Norms provide enough information to supervise them and to detect if the norm is respected or not. In works above-mentioned nothing is said about the norm violation detection.

6 Conclusion and perspectives

We have proposed in this paper the $\mathcal{S}$YNAI platform being part of MA$\mathcal{B}_{\text{ELI}}$ which is composed of supervisor agents organised with the $\mathcal{M}$OISE^{Inst} meta-model. This meta-model is considered as an institution organisation specification especially through each society roles rights and duties description as well as these rights and duties arbitration.

Two kinds of agents will evolve in the electronic institution: the domain agents and the supervisor agents. With $\mathcal{M}$OISE^{Inst} we expressed authority roles that $\mathcal{S}$YNAI agents will play, as well as the missions related to their ability to detect norms violations and to punish culprit domain agents.

There was no intention to impose a unique domain agents definition due to the heterogeneity objective. However we can specify the supervisor agents functionalities operating in $\mathcal{S}$YNAI. But the events definition and the way they are treated is not perfect. Next steps are to transform the events definition into an $\mathcal{M}$OISE^{Inst} Ontological Specification and messages exchange defined inside supervisors into an Interactional Specification.

References

[1] Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. Jade - a fipa-compliant agent framework. In *4th International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents*, London - UK, 1999.

[2] Frank Dignum. Agents, markets, institutions, and protocols. In Frank Dignum and Carles Sierra, editors, *Agent-Mediated Electronic Commerce - The European AgentLink Perspective*, volume 1991 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, 2001. ISBN 3-540-41671-4.

[3] Marc Esteva, David de la Cruz, and Carles Sierra. Islander: an electronic institutions editor. In *1rst international joint conference on autonomous agents and multiagent systems (AAMAS'02)*, volume 3, pages 1045–1052, Bologna - Italy, 2002. ACM Press. ISBN 1-58113-480-0.

[4] Marc Esteva, Bruno Rosell, Juan A. Rodriguez-Aguilar, and Josep Ll. Arcos. Ameli: An agent-based middleware for electronic institutions. In *3rd international joint conference on Autonomous Agents & Multi-Agent Systems (AAMAS'04)*, volume 1, pages 236–243, New York City - USA, 2004. ACM Press. ISBN 1-58113-864-4.

[5] Benjamin Gâteau, Olivier Boissier, Djamel Khadraoui, and Eric Dubois. $\mathcal{M}$OISE^{Inst}: An organizational model for specifying rights and duties of autonomous agents. In *1st International Workshop on Coordination and Organisation (CoOrg 2005)*, Namur - Belgium, 2005.

[6] Benjamin Gâteau, Olivier Boissier, Djamel Khadraoui, and Eric Dubois. Controlling an interactive game with a multi-agent based normative organizational model. In *Proceedings of the Coordination, Organization, Institutions and Norms in agent systems workshop (COIN) at the 17th European Conference on Artificial Intelligence (ECAI)*, Riva del Garda - Italy, 2006.

[7] Benjamin Gâteau, Djamel Khadraoui, and Eric Dubois. Architecture e-business sécurisée pour la gestion des contrats. In *3ème Conférence sur la Sécurité et Architectures Réseaux (SAR)*, La Londe - France, 2004.

[8] Jomi Fred Hübner, Jaime Simo Sichman, and Olivier Boissier. A model for the structural, functional, and deontic specification of organizations in multiagent systems. In *16th Brazilian Symposium on Artificial Intelligence (SBIA'02)*, volume 2507 in *LNAI*, pages 118–128. Springer, 2002.

[9] Jomi Fred Hübner, Jaime Simo Sichman, and Olivier Boissier. $\mathcal{S}-\mathcal{M}$OISE^{+}: A middleware for developing organized multi-agent systems. In *Proceedings of the International Workshop on Organizations in Multi-Agent Systems: From Organizations to Organization Oriented Programming in MAS (OOOP)*, volume 3913 in *LNCS*. Springer, 2005.

[10] A. Jones and J. Carmo. *Handbook of Philosophical Logic*, chapter Deontic logic and contrary-to-duties, pages 203–279. Kluwer, 2001.

[11] Douglass C. North. *Institutions, Institutional Change and Economic Performance*. Political Economy of Institutions and Decisions. Cambridge University Press, 1990. ISBN 0521397340.

[12] Stefan Poslad, Phil Buckle, and Rob Hadingham. The fipa-os agent platform: Open source for open standards. In *PAAM*, 2000.

[13] Javier Vázquez-Salceda, Virginia Dignum, and Frank Dignum. Organizing multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 11(3):307–360, November 2005. ISSN: 1387-2532.