Dynamic Responsibilities Assignment in Critical Electronic Institutions - A Context-Aware Solution for in Crisis Access Right Management

Bonhomme, Cédric; Feltus, Christophe; Petit, Michaël

Published in:
Proceedings of the The sixth International Conference on Availability, Reliability and Security ("ARES 2011 - The International Dependability Conference"), Vienna, Autria.

DOI:
10.1109/ARES.2011.43

Publication date:
2011

Document Version
Early version, also known as pre-print

Dynamic Responsibilities Assignment in Critical Electronic Institutions - A Context-Aware Solution for in Crisis Access Right Management

Cédric Bonhomme[†], Christophe Feltus[†‡], Michaël Petit[‡]
[†]Public Research Centre Henri Tudor, Luxembourg-Kirchberg, Luxembourg
[‡]PReCISE Research Centre, Faculty of Computer Science, University of Namur, Belgium
cedric.bonhomme@tudor.lu, christophe.feltus@tudor.lu, mpe@info.fundp.ac.be

***Abstract*— Nowadays critical IT infrastructures constitute the pillars of our economy. Being able to react quickly and in real time is a crucial challenge for the security officers in charge of maintaining those infrastructures operationally. Our state of the art in this field has highlighted that many architectures exist to dynamically support the reaction after the detection of an incident infrastructure. Those architectures are mostly elaborated based on a multi-agent system approach that offers the possibility to work in a decentralized and heterogeneous environment. However, in the meantime, we have observed that those architectures are based on a static assignment of functions to agents and that, as a consequence, isolating an agent or breaking the communication channel between two of them could create serious damage on the management of the crisis. In this paper, we propose an innovative approach for making the assignment of functions to agents in the critical architecture dynamic. Our approach exploits the concept of agent responsibility that we assign dynamically to those agents depending on the crisis type and severity. Simultaneously we explain the dynamic assignment of the access rights necessary to perform the obligation linked to these new responsibilities. This dynamic assignment of responsibilities is illustrated based on the architecture defined in the ReD project.**

Keywords: access right, security, multi agent system, crisis management reaction architecture, agent responsibility.

I. INTRODUCTION

Responsibilities and access rights management in critical infrastructures are crucial activities in order to avoid security failures [1]. Those critical infrastructures are defined by (i) the importance of the services that they furnish to society, like for instance power distribution, telecommunication, health rescue, or specific administrative resources and (ii) the dependability between two or more of those services.

In our previous work [2, 3], we have defined a security decision-reaction architecture for heterogeneous distributed network. This architecture was firstly composed of a Multi-Agent System (MAS) that offers the advantage to react quickly and efficiently to an attack while being adapted for heterogeneous and distributed networks, and secondly of a decision support system that helps agents to make decisions based on utility preference values. The preference choice is achieved by taking uncertainty into account through Bayesian networks and influence diagrams. These main architecture objectives provide the logical and technological bases for the monitoring and for the reaction after the occurrence of an incident on the network. Although it permits to cover the entire conceptual layer from the incident detection at the very low technical layer up to the escalation of the incident to upper layer based on the decision mechanisms, our solution did not consider the normative specifications related to the responsibilities and accountability of the agents involved in it (including the technical and the human agents), and did not provide the possibility of adapting the agent responsibility during the occurrence of a crisis.

In this paper, we propose to face that problem by enhancing the architecture with a dynamic assignment of responsibility to agents. Introducing the agent responsibility is a relevant topic because it permits to address many challenges at the same time: (i) agents are either human or software (ii), agents are issued from different fields (telecom, power distribution, etc.) and applications (iii) agents' responsibilities in heterogeneous systems are formalized with responsibility models from those heterogeneous systems and consequently, a limitation of interoperability between those models may arise.

To address that matter, we have enhanced the reaction architecture with a mechanism that permits to dynamically assign responsibility to agent and to modify the assignment according to crisis situations.

In order to limit access to all information by all agents on the network, the access rights granted to the agents are dependent on their assigned responsibilities.

The paper is structured as follows: the next section presents ReD [9], an agent based architecture to respond to incidents. Section III introduces the responsibility of agents and Section IV integrates that agent responsibility in ReD through a real case study. Finally the last section concludes the paper.

II. REACTION ARCHITECTURE

The reaction architecture presented in this section is based on the ReD project [4]. The ReD (Reaction after Detection) project defines and designs a solution to enhance the detection/reaction process and improves the overall resilience of IP networks. The architecture is composed of software components and of human agents that obligations concern the monitoring of the software component.
The main components of the ReD architecture are:

- PDP (Policy Decision Point) receives the new security policies and deploys them at the enforcement points (PEP);

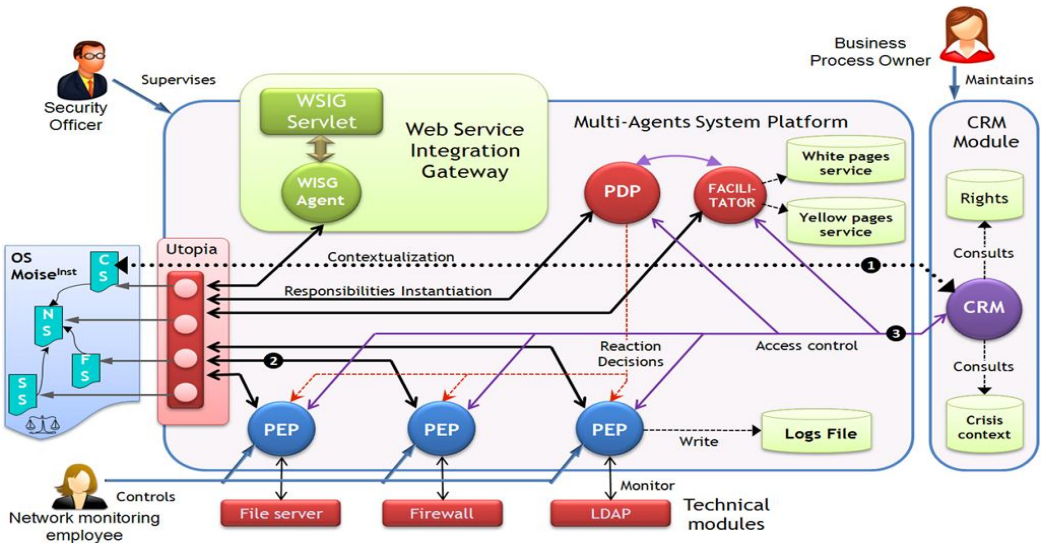

Figure 1. Synoptical ReD Architecture

- PEP (Policy Enforcement Point) enforces the security policies provided by the PDP.

Thanks to the JADE Multi-Agent System development framework [5, 6], communication between agents is made totally transparent. Fig. 1 highlights how the global architecture of ReD is implemented with UTOPIA (namely, ReDTopia) and combined with a Context Rights Management agent (CRM). This architecture will be used for the use case of the paper.

Firstly, UTOPIA [7] is an insTitution Oriented ProgrammIng frAmework. UTOPIA aims at simplifying agent based institutions programming. For this work the use of UTOPIA enables us to tackle some lacks of ReD [8] by taking into account the state of the organization. In our context, the organization is considered as a set of actors who perform tasks and who can be grouped within a regulated structure and designed to achieve objectives and personal interests [11]. The second point concerns the Context Rights Management (CRM) agent who is in charge of providing rights to software agents (e.g.: PDP, PEP and Facilitator). Rights are dependent on the Context Manager of UTOPIA. For this, the CRM uses the rights and the crisis context database. The role of these databases is discussed in more detail later in section III.

III. AGENT RESPONSIBILITY SPECIFICATION

A. *Contextual agent responsibility*

Mostly two agents are concerned by the access right function: the agents who provide the rights at the access right management layer and the agents who receive the rights at the operation layer. In a non-crisis context, agents are assigned to responsibilities like PEP, PDP, CRM, etc. Those assignments of responsibilities to agents require specific access rights. By analyzing for instance the activity of monitoring the network (see Fig. 1), we observe that this activity is handled by a human agent: the network monitoring employee that is helped by software agents (assigned PEP) who are intelligent probes which analyze the data flow on the file servers, the firewalls and the LDAP. In a non-crisis context, the intelligent probes have the responsibility to collect data, to make a basic correlation between the flows and the antecedent flow records and to report this analysis to the network monitoring employee.

When a crisis occurs, for instance an external attack on the network monitoring activity, the normal monitoring rules and procedures do no longer work as usual and it is required to change the responsibility of the agents. For instance, if the attacker can perform DoS attack on a part of the network and those agents are out of order, other agents then have to assure the responsibilities of the attacked agents. Practically, if the probe that analyses the dataflow on a router is down, then the probe that analyses the flow on the IDS (Intrusion Detection System) can decide to analyze both, the flow on the router and the flow on the IDS. To meet this new responsibility during the attack, he requests additional access rights.

In the current architecture, the automatic assignment of rights to agent depending on their responsibilities is not immediate and this contradicts the need to react promptly to the attack.

To tackle the weakness, our approach is based on the refinement of the agent responsibility and on the granting of access rights, at the management layer, based on the responsibility. The right management agent exploits two databases accessed by the CRM agent. The first database contains the rights corresponding to the responsibilities and the second database contains a set of crisis contexts. Each crisis context defines, depending on the type of crisis, which agent is assigned which responsibility. Both databases are accessed by the CRM to provide rights at the operational layer and are managed by the Business Process Owner at the access right management layer.

B. *The responsibility meta-model*

The definition of the agent responsibility in ReD is mostly incomplete. Most of the architecture is only

considering the agent with respect to the outcome that an agent has to produce. Sometimes, advanced solutions integrate the inputs that those agents request for performing the outcome. We define the responsibilities as a state assigned to an agent to signify him its obligations concerning the task, its accountabilities regarding its obligations, and the rights and capabilities necessary to perform it. In order to integrate a dynamic re-assignment of the responsibility from one agent working in normal condition to one agent working in a crisis environment, we consider all the concepts which compose the responsibility. In [12] we have proposed a model that can be used to depict the agent responsibility containing three sets of concepts: (i) the obligation and accountability, (ii) the right and capability and (iii the delegation and assignment process that we explain in the next sub-sections.

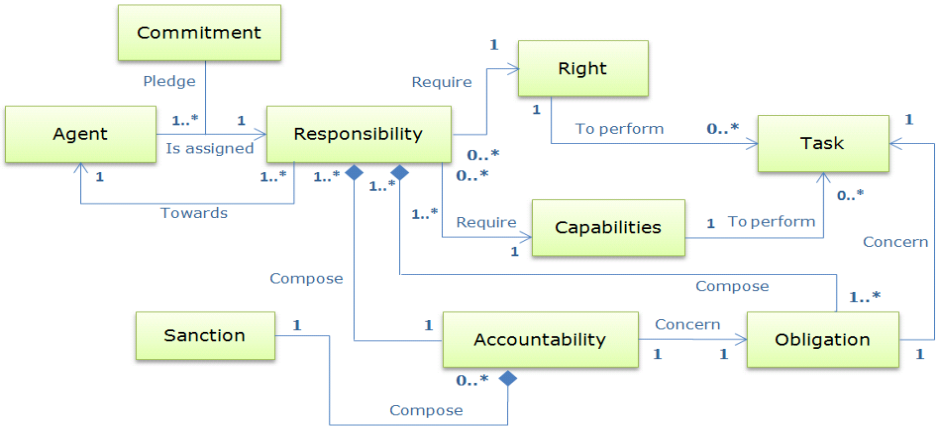

Figure 2. The Agent responsibility model

2) Concept of right/capability

The concept of capability describes the requisite qualities, skills or resources necessary to perform a task. Capability is a component which is part of most of the models and methods [14], and it may take the form of knowledge or know-how, possessed by the agent such as ability to make decision, its processing time, its faculty to analyze a problem, and its position on the network.

Right (Fig. 2) is a common component but is not systematically included in all responsibility frameworks. Right encompasses facilities required by an agent to fulfill his obligations. We make the distinction between the pre-assignment right which gathers rights that the agent needs to possess before he can be assigned a responsibility, and the post-assignment right (e.g. the access right) that the agent gets once he is assigned responsible.

Authority describes the power or right to give orders or to make decisions. This concept is introduced e.g. in CIMOSA [15] as the *power* to command and control other human agents and to assign responsibilities. CIMOSA argues that responsible agents have rights over resources in the first place and over processes, actions and tasks in the second place.

Delegation right describes the right to transfer a part of the responsibility to another agent who pledges commitment for it. Transferred responsibilities may be rights, obligations or both. The delegation of an obligation may or may not be accompanied by the delegation of the right to further delegate this same obligation [14]. This delegation of rights depends on the right's type (access to information, money, time…) and on the agent's status, function or position. This delegation also may or may not include the transfer of the related accountability [16].

1) Concept of obligation/accountability

Obligation is the most frequent concept appearing in literature [10] as well as in industrial and professional frameworks. Obligation is a duty which links a responsibility with a task that must be performed. We define a task as an action using or transforming an object.

Accountability is a duty to provide justifications on the performance of a task to someone else under threat of sanction [13]. Accountability is a type of obligation to report the achievement, maintenance or avoidance of some given state to an authority and, as consequence, is associated to an obligation.

3) Assignment/delegation process

Assignment is the action of binding an agent to a responsibility. Delegation is the transfer of an agent's responsibility assignment to another agent. The commitment pledged by the agent related to an assignment or delegation represents his engagement to meet the corresponding responsibility and the assurance that he does it in respect of good practices. This component, traditionally called *Commitment's antecedent* in literature, corresponds to more pragmatic variables [17].

Based on the commitment outcomes and antecedent definitions, we may assume that an agent being committed to the responsibility of a task implies on the one hand an increase of trust in the achievement of the obligation or in the accountability attached to it, and on the other hand more

efficiency (and consequently more capabilities) for this agent to perform the task.

C. Agents responsibilities specifications according to the responsibility model

Based on the responsibility meta-model defined in the above subsection B, we may instantiate the responsibility meta-model for each responsibilities of the network. Because of the size of the paper, only the four most important meta-concepts will be instantiated here: The obligation related to the task, the agent towards whom the agent is responsible the capability, and the right.

Table I presents the responsibilities of the ReD agents engineered from [2] and [4]. For the PEP, we observe that the responsibility include obligations such as the obligation to retrieve the logs from the component he monitors, to update the log file database, etc. To meet this obligation, he must have the capabilities to be on the same network as the component he controls and to communicate with the PDP and the facilitator agent. He also must have the right to read the log file on the concerned network component and to write the log in a central logs database. Finally, he is a towards the Head IT Operation.

Table I summarizes the responsibilities of all agents. Those responsibilities conceptual components will afterwards be used in practice for the dynamic assignment as explained in section IV.

Table I: Agents responsibilities

	Responsibility	Obligation related to the task	Accountable towards	Capability	Right
SOFTWARE	PEP	Retrieve the logs from the component he monitors Analyses the logs. Update the log file database Communicate with the facilitator the get the address of the other components (PDP, CRM)	Head IT Operation	PEP must be on the same network as the component to control PEP must be able to communicate with the PDP and the facilitator agent	Read log file on the concerned network component Write log in the central logs database
	PDP	Based on the incident report from the PEP, decide which reaction policy is appropriate to be deployed Communicate with the facilitator the get the address of the other components (PDP, CRM)	Security Officer	Fast bandwidth High CPU resources Central position on the network	No specific right
	Facilitator	Provide IT addresses of the requested component Make a mapping between the component name and the IP address.	Security Officer	Position in which he is always available Bandwidth depending on the network size	Read and write to the white pages services database Read and write to the yellow pages services database
	CRM	Provide access right on request Provide Crisis context information to the contextual specification element of the OS Moise^{Inst}	Security Operator	Bandwidth depending on the network size	Read and write to the white pages services database Read and write to the yellow pages services database
	WSIG	Transfer policies to the PDP Communicate with the facilitator the get the address of the other components (PDP, CRM)	Security Officer	Have a position on the network close to the PDP Be on the same network as the servlet to be an interface between the servlet and the PDP	No specific right
HUMAN	Security officer	Control the Business Process Owner activity Monitor that logs are up to date Define the crisis context level	Directors board	Good analyses skill Good security experience Ability to make decision in a crisis situation	Access to the log files Access to the servlet
	Network Monitoring Employee	Control PEP activities Report incident to the Business Process Owner	Security Officer	Good technical skill	Access to log files Access to network monitoring tools
	Business Process Owner	Decide the Crisis context level Update Crisis context database Update access right database	Directors board	Be able to understand the business impact of the incident to decide the crisis context level	Right to the crisis context Right to the right database

IV. DYNAMIC RESPONSIBILITY ASSIGNMENT

In this section, we illustrate, based on the ReD architecture, how enhancing the agent responsibilities could contribute to more efficiency and effectiveness in crisis management situations. We first introduce the ReDTopia component. Then, we explain how, at a technical layer, the responsibilities are assigned in the network based on specifications from the logical layer as described in Section III.C and how those responsibilities are dynamically assigned according to determined crisis levels. The transfer of the responsibility model to the technical layer is achieved in two steps: the concepts of responsibilities, capabilities, obligation and accountabilities are translated by the means of the RedTopia architecture and the concept of right is

instantiated and operationalized according to the context by the CRM agent.

A. The ReDTopia architecture

As presented in section II, UTOPIA simplifies the development of Multi-Agent Systems by dynamically assigning responsibilities. Its main function is to assure the intelligent distribution of responsibilities to agents in an evaluative organization. The model used by UTOPIA to specify the organization of an Electronic Institution is Moise^{Inst} [10, 18]. The supervision of the agents functioning in that institution is supervised and controlled with a set of institution services regrouped in a specific *normative middleware* called SYNAI [19] on which the software agents are executed.

Moise^{Inst} is an Organization Specification (OS) system composed by four dimensions. These four types of specifications are described in a XML file that creates a framework for specifying responsibilities and, consequently, for establishing the agent responsible in terms of tasks, obligations, rights, accountabilities and capabilities. The next subsections explain in details how to design those specifications.

1) The Structural Specification

The Structural Specification (Fig. 3) defines (i) the responsibilities which agents are assigned in the logical layer (ii) the relations between the responsibilities in terms of data exchange during the execution of the tasks. E.g. **<Link source="PEP-Fileserver" destination="PDP"/>** specifies that the PEP-Fileserver is allowed to send information to the PDP (iii) the Groups of agents assigned to the same responsibility. These groups are used by UTOPIA to regulate the responsibility depending on the context. E.g. In case of a crisis, if a PEP from a PEP group is corrupted, the responsibility of that failing agent is transferred to new agents from the same group. For example, **<Responsibility id="PEP" min="3" max="3"/>** creates a PEP group with a cardinality sets to 3.

```
<StructuralSpecification>
    <Group id="RED" min="5" max="5">
        <Responsibility id="CRM" min="1" max="1" />
        <Responsibility id="PDP" min="1" max="1" />
        <Responsibility id="PEP" min="3" max="3" />
        <Responsibility id="PEP-FileServer" min="1" max="1" />
        <Responsibility id="PEP-Firewall" min="1" max="1" />
        <Responsibility id="PEP-LDAP" min="1" max="1" />

        <Link source="PEP" destination="PDP" />
        <Link source="PEP-FileServer" destination="PDP" />
        <Link source="PEP-Firewall" destination="PDP" />
        <Link source="PEP-LDAP" destination="PDP" />
        <Link source="PEP-FileServer" destination="CRM" />
        <Link source="PEP-Firewall" destination="CRM" />
        <Link source="PEP-LDAP" destination="CRM" />
    </Group>
</StructuralSpecification>
```

Figure 3. Structural Specification XML schema

2) The Functional Specification

The Functional Specification (FS) on Fig. 4 defines global business processes that can be executed by the different agents participating to the Organization according to their responsibilities and Groups. In our case, the CRM has to provide access rights and is accountable to report that task performance to the Security Operator., e.g. **<TaskId obligation_task= "Transfer policies to appropriate PEP" accountability_to="SecurityOfficer"> PDPListen</TaskId>**.

```
<FunctionalSpecification>
    <TaskId obligation_task="Transfer policies to appropriate PEP"
            accountable_to="Security Officer">PDPListen</TaskId>

    <TaskId obligation_task="Retrieve and Update log file database"
            accountable_to="Head IT Operation">PEPListen</TaskId>

    <TaskId obligation_task="Provide access right on request"
            accountable_to="Security Operator">CRMListen</TaskId>
</FunctionalSpecification>
```

Figure 4. Functional Specification XML schema

3) The Contextual Specification

The Contextual Specification (CS) specifies the possible evolution of the organization in terms of a state/transition graph. E.g. for our needs we have defined two main contexts depending on the situation: normal or crisis and we can have a transition between a normal and a crisis context: **<Transition id="t1" source="normal" target="crisis eventId="crisis"/>** .

4) The Normative Specification

```
<NormativeSpecification>
    <Norm id="N1" bearer="CRM" context="normal"
          task="CRMListen" />
    <Norm id="N2" bearer="PDP" context="normal"
          task="PDPListen" />
    <Norm id="N3" bearer="PEP" context="normal"
          task="PEPListen" />
    <Norm id="N4" bearer="PEP-FileServer" context="normal"
          task="PEPListen" />
    <Norm id="N5" bearer="PEP-Firewall" context="normal"
          task="PEPListen" />
    <Norm id="N6" bearer="PEP-LDAP"  context="normal"
          task="PEPListen" />
</NormativeSpecification>
```

Figure 5. Normative Specification XML schema

The Normative Specification (NS) on Fig. 5 defines the deontic relations gluing the three independent Specifications (SS, FS, CS). This NS clearly states rights and duties of each responsibilities/Groups defined at the SS layer for tasks defined at of FS layer in the context of specific states from the CS layer. E.g. **<Norm id="N2" bearer="PDP" context"normal" action"="PDPListen"/>** specifies that PDP agent is responsible to perform **PDPListen** in a normal situation.

B. Example of transfer from a non-crisis to a crisis context

To illustrate the transfer from a non-crisis to a crisis context, we depict the case of the PEP responsibility assignment to agent that, in a normal situation, is the following: One agent is assigned a PEP responsibility, each agent is associated to one component that he must monitor and he reports to the PDP. Each time an agent wants to access a component of the network (e.g. the LDAP), he needs to previously contact the CRM. That CRM consults

the access rights database in order to retrieve the rights associated to the responsibility assigned to the agent.

In an abnormal situation, the context evolves and an appropriate crisis context is selected from the Crisis context database. This database, which is maintained by the Business Process Owner, contains a set of crisis contexts. When the Network Monitoring employee detects a crisis, he refers to the Business Process Owner that sets the new context in order to restore the situation. Each context is adapted to a specific case. For example, if the crisis concerns the corruption of a PEP agent, a transfer of responsibility from the corrupted agent to another agent from the same group is required. This responsibility transfer also implies a simultaneous transfer of rights.

The diagram of activities, as highlighted by white numbers in black circles on Fig. 1, is the following:

❶ The CRM retrieves the new context (crisis context) in the context database and send it to the CS component of Moise^{Inst};

❷ Based on the new context, Moise^{Inst} reorganizes the agents' responsibilities taking into consideration the agent commitment to be assigned to the additional responsibility [20]. In that case, the new context requests a new deployment of the PEP responsibilities, such that the responsibilities of the PEP which is down are transferred to agents from the same group;

❸ The agents who receive the additional PEP responsibilities request access rights corresponding to those responsibilities to the CRM.

V. CONCLUSIONS AND FUTURE WORKS

Critical infrastructures are more and more present and needs to be seriously managed and monitor regarding the increasing amount of threats. In order to achieve this and to react when an attack occurs, we have defined a dynamic Multi-Agent System, which supports the reaction after an incident. This system, initially developed for static assignments of responsibility to agents, has needs for more dynamism to stay aligned to the new arising risks. That paper gives an insight about the concept of responsibility and addresses that new challenge by providing a framework for assigning responsibilities to agents depending on the crisis context. That contextualized responsibility assignment permits to dynamically manage the agent access rights.

The paper is illustrated based on a use case that shows how the transfer of rights for a corrupted PEP is performed during an evolution from normal to crisis situation.

ACKNOWLEDGMENT

This research work was funded by the National Research Fund of Luxembourg in the context of TITAN (Trust-Assurance for Critical Infrastructures in Multi-Agents Environments, FNR CO/08/IS/21) project.

REFERENCES

[1] Council Directive 2008/114/EC of 8 December 2008 on the identification and designation of European critical infrastructures and the assessment of the need to improve their protection.

[2] C. Bonhomme, C. Feltus, and D. Khadraoui, "Multi-Agent based Decision Mechanism for Incident Reaction in Telecommunication Network" (poster), Eight ICCSA IEEE, Hammamet, Tunisia

[3] C. Feltus, D. Khadraoui, and C. Bonhomme, "Electric blackout prevention: Toward a computer-mediated weather alert broadcasting solution", in International Conference on Society and Information Technologies (ICSIT 2010), 2010, pp. 45-50.

[4] C. Feltus, D. Khadraoui and J. Aubert,: A Security Decision-Reaction Architecture for Heterogeneous Distributed Network., ARES, IEEE Computer Society, 1–8, 2010

[5] Bellifemine, A. Poggi, G. Rimassa. JADE - A FIPA-compliant agent framework. Part of this report has been also published in Proceedings of PAAM'99, London, April 1999, pp.97-108

[6] F. Bellifemine, G. Caire, A. Poggi, G. Rimassa, JADE - A White Paper. Sept. 2003

[7] P. Schmitt, C. Bonhomme, J. Aubert, and B. Gâteau, "Programming electronic institutions with UTOPIA", in Demo Session of the 22nd International Conference on Advanced Information Systems Engineering (CAiSE'10), 2010.

[8] P. Schmitt, C. Bonhomme, and B. Gâteau, "Easy programming of Agent based Electronic Institution with UTOPIA", in the 10th international conference on New Technologies of Distributed Systems (NOTERE 2010), Tozeur - Tunisia, May 31 - June 2, 2010.

[9] C. Feltus, D. Khadraoui, B. de Remont and A. Rifaut, Business governance based policy regulation for security incident response. In: Crisis'07, Marrakech, Morocco(2-5 July 2007)

[10] B. Gâteau, O. Boissier, D. Khadraoui and E. Dubois. MOISE^{Inst}: An Organizational Model for Specifying Rights and Duties of Autonomous Agents. Workshop of the 7th International Conference on Coordination Models and Languages, Namur – Belgium, 2005

[11] B. Gâteau. Modélisation et Supervision d'Institutions Multi-Agents. PhD Thesis held in cooperation with Ecole Nationale Superieure des Mines de Saint Etienne and CRP Henri Tudor, June 2007.

[12] C. Feltus, M. Petit, Building a Responsibility Model Including Accountability, Capability and Commitment, ARES 2009 – IEEE, 16-19/3/2009, Fukuoka, Japan.

[13] B. C. Stahl, "Accountability and reflective responsibility in information systems". In: C. Zielinski et al. The information society - emerging landscapes. Springer, 2006, pp. 51 -68.

[14] I. Sommerville, R. Lock, T. Storer, and J. Dobson, "Deriving Information Requirements from Responsibility Models", CAiSE 2009, Amsterdam, June 8-12.

[15] F. B. Vernadat, "Enterprise Modelling and Integration", Chapman & Hall, London (1995), ISBN 0-412-60550-3.

[16] R. S. Sandhu, and V. Bhamidipati, 1998. "An Oracle implementation of the PRA97 model for permission-role assignment". Third ACM Workshop on Role-Based Access Control.

[17] C. Vandenberghe, K. Bentein, and F. Stinglhamber, "Affective commitment to the organization, supervisor, and work group: Antecedents and outcomes", February 2004, pp. 47-71.

[18] B. Gâteau, O. Boissier, D. Khadraoui and E. Dubois. Controlling an Interactive Game With a Multi-agent Based Normative Organizational Model. Workshop of the 17th European Conference on Artificial Intelligence (ECAI), Riva del Garda – Italy, 2006.

[19] O. Boissier and B. Gâteau. Normative Multi-Agent Organizations: Modeling, Support and Control. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Germany, 2007.

[20] G. Boella, R. Damiano, J. Hulstijn, and L.v.d. Torre. Role-based semantics for agent communication: embedding of the 'mental attitudes' and 'social commitments' semantics, 2006. Japan: ACM