Who Govern my Responsibilities? SIM: a Methodology to Align Business and IT Policies in the Industrial Field

Feltus, Christophe; INCOUL, Christophe; AUBERT, Jocelyn; GATEAU, Benjamin

Published in:
Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 09), Volume 13 | Part 1,Moscow, Russia.

DOI:
10.3182/20090603-3-RU-2001.00041

Publication date:
2009

Document Version
Early version, also known as pre-print

Who Govern my Responsibilities? SIM: a Methodology to Align Business and IT Policies in the Industrial Field.

Christophe Feltus, Christophe Incoul, Jocelyn Aubert, Benjamin Gâteau

Public Research Centre Henri Tudor, 29, Avenue John F.Kennedy,
L-1855 Luxembourg-Kirchberg, Luxembourg
(e-mail: christophe.feltus@tudor.lu)

Abstract: Governance of IT is becoming more and more necessary in the current financial economic situation. This trend does not avoid the definition of corporate and IT policies. To improve that matter, the paper has for objective to propose a methodology for defining policies that are closer to the business process, and based on the strict definition of the actors' responsibility. This responsibility model is mainly defined based on the three concepts of capability, accountability and commitment. The methodology is illustrated based on a case study that highlights how it is possible to implement access control mechanism through agent-based infrastructure by extracting requirements from company practices and process formalizations.

Keywords: Models, Responsibility, Process Model, IT Policies, Agent Framework, ICT Governance.

1. INTRODUCTION

The importance of the *Governance of ICT* is becoming more and more important in companies, particularly since the accounting scandals of 2002, and more currently through the ongoing market crisis. The case of Enron is one famous and well-known example. Following those scandals, a lot of laws and standards were published in order firstly to guarantee the stability of the financial sector and, by extension, all sectors of the industrial economy, and secondly, to enhance the governance of all these public and private companies. Sarbanes-Oxley is one of these laws that aim at providing guarantees over the company's accountability. The new ISO/IEC 38500:2008 *Corporate governance of information technology* standard is one standard that provides a framework for effective governance of IT. One of the main constraints imposed by these laws and standards is to have responsibilities clearly established and accepted internally by the collaborators and externally by the stakeholders as well. The importance of that statement has oriented our works and we propose, in the scope of that research, to make a contribution to one of the most significant pillars of the governance, which is responsibility. This concept has a major meaning in the managerial pyramid of the company in that it composes the structural artefact of the decisional framework. Consequently, it sounds justified to guarantee accurate decisions to ensure that responsibilities of enterprise activities are rightly defined and enforced. It implies that having responsibility appropriately affected involves a prior clear definition of its components (capability, accountability and commitment), and to develop an efficient methodology to analyse and fix them through all the decisional layers of the company, from the top layer down to the technical one. The first part of that twofold objective has already been basically investigated in our previous work (Gâteau et al., 2008). Indeed, by depicting how the responsibility is introduced and interpreted in a large number of industrial frameworks, we have been able firstly to extract main components of it and, secondly, to validate them through a responsibility ontology. This ontology has been built by analyzing how responsibility components are declined in the realm of IT security, from access control models such as RBAC (Sandhu et al., 2000) up to framework for ICT governance like Cobit [4], in the realm of requirement engineering, as well as through EAM frameworks like CIMOSA (http://www.cimosa.de/) or PERA (http://www.pera.net). The issue of that wide review of structuring framework has lead to the conclusion that mainly three components build the responsibility concept. Those components are: capability, accountability and commitment. The second part of the research aims at elaborating a methodology for defining, structuring and deploying the responsibility in industries' information systems. In the literature, each area of information activity already has methods for elaborating its information frameworks. In the field of requirement engineering: (Yu et al., 2001), (Antón, 1996) and (Fontaine et al., 2001), in the field of access control: (Fernandez, 1997) (Roeckle et al., 2000) (Crook et al., 2002) (Qingfeng et al., 2003) (Neumann et al., 2002), and in the field of EAM: (Vernadat, 1995) (Meir, 1999). However, none of them integrate the notion of responsibility in their methodology. Our work aims at filling that gap with an innovative five-step method focusing on the responsibility ontology. The advantage of this method will be:

- It is focused on the concept of responsibility rather than on a concept of role, user, or activity,
- It is adapted to manage responsibility exceptions.

- It could be tailored to a large part or to a small part of the activity,
- It provides a generic responsibility diagram that may be rapidly used and adapted for a specific framework,
- It is based on a responsibility model that has been validated against different domains like access control or requirement engineering.

2. RESPONSIBILITY MODEL

A plethora of definitions of responsibility exists. We may however state that the commonly accepted responsibility definition encompasses the idea of having the obligation to ensure that something happens.

Previous work (Aubert et al., 2008) shows that responsibility can be described as a set of three additional elements that are capability, accountability and commitment. Figure 1 depicts our responsibility model. The relation between responsibility and the three other concepts is of the form 0..* to 1. That means that being responsible involves the possibility to dispose of many capacities, accountabilities and commitment.

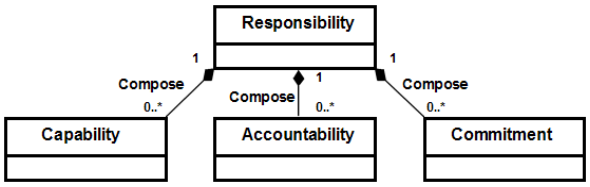

Fig. 1. Responsibility model

Capability describes the quality of having the required qualities or resources to achieve a task. For instance, a strategic capability for a given responsibility could be: "A resource must know the strategic objectives of the organisation". An operational capability could be: "The coach of the resources must have write access to the HR software".

Accountability describes the state of being accountable on the achievement of a task. For instance, a strategic accountability for a given responsibility could be: "A project leader must achieve the financial Key Performance Indicators defined for the project". An operational accountability could be: "The project manager must define the project plan".

Finally, *Commitment* is the engagement of a stakeholder to fulfil a task and the assurance that he will do it. For instance, a strategic commitment for a responsibility could be:" The Chief Financial Officer accepts to manage the accounting department and not commit insider dealing". An operational commitment could be: "An employee of the procurement staff accepts not to use the system for his personal use".

The consistency between concepts may also be examined based upon the assumption that the capability needed for assuming a responsibility corresponds to the accountability of another user's responsibility. Both responsibilities' components, capability and accountability, are strongly linked to each other in that accountability of a role or a person permits to deduce capability of another role or person, and conversely a capability stems from accountability (e.g.: The capability "An engineer has access to a specific file" stems from the accountability "An engineer has to share a specific file with another engineer").

3. METHODOLOGY

These complementarities of the responsibility's concepts are an important element for the definition of policies according to our responsibility model. The methodology described in this section has for objective to explain how to define the enterprise IT policies according to the responsibility model. This methodology is a five-step approach. To facilitate the understanding, we illustrate each step with a case study.

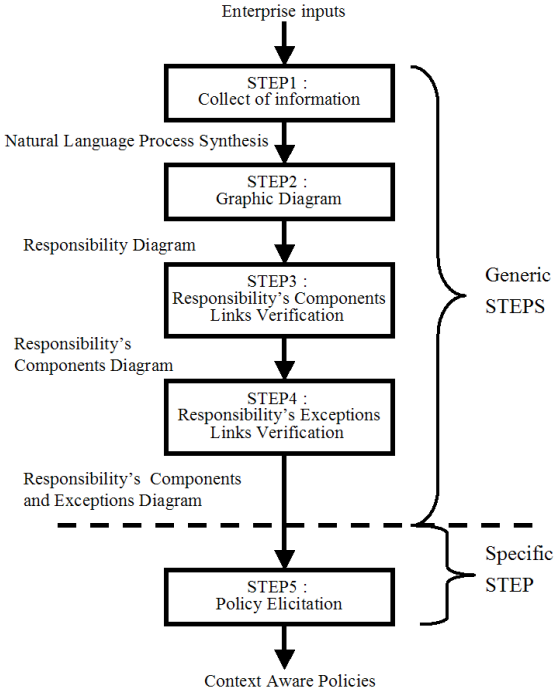

Fig. 2. SIM Methodology

3.1 STEP 1. Collection of information

The first step has for objective to define the context and to collect each component that will be formalized in the policy.

STEP 1 input: Inputs of step 1 are elements collected from business case studies, business processes, business procedures, and effective practices in the enterprise.

STEP 1 output: Output of step 1 is a formalized and structured synthesis of the process in natural language.

STEP 1 actions: The actions performed at this step encompass a number of activities to collect information about the process and the responsibility components. These activities are interviews of the key members of the personnel, analyses

of existing process descriptions, analysis of enterprise referentials like the ISO 9000 quality book.

By these activities, we can summarize:

- Process responsibilities, as well as their composing elements, like accountabilities, capabilities, and commitments;
- The existing relations between responsibilities and responsibility components.

To illustrate that methodology, we describe this first step based on the "Enterprise Christmas Gifts Process". After having performed interviews and analyses of the enterprise activity, the following synthesis has been written:

Mr. Johnson is the manager of the IT Company named "HighTech". Every year, Mr. Johnson organizes, during the Christmas period, a large mailing of postcard and gifts in order to thank (best) customers for their loyalty.

This process permits to develop customer loyalty and update the yearly customer list.

Due to an overload of work at this period (such as closing the annual report) and because this task is less business sensitive as some other production tasks, Mr. Johnson prefers delegating this task to his employees.

Generally, he assigns work to:

A responsible for updating the customer list, selecting mail targets, printing envelopes and sending cards and bottles of champagne to the best customers;

A responsible for receiving customers' feedback (such as wishes, complaints or orders), analysing it and providing him with the results ;

A responsible for giving sufficient access rights to all process stakeholders.

However, for some very important customers, Mr. Johnson prefers sending bottles of champagne himself.

Each year, after receiving the feedback analysis, Mr. Johnson realizes that the process encounters some problems, such as (best) customers that haven't received cards or gifts, or customers having received them twice.

3.2 STEP 2 Graphic diagramme

The second step translates the process from natural language toward a graphical representation.

STEP 2 input: Input of step 2 is the synthesis achieved in step 1.

STEP 2 output: Output of step 2 is a graphical representation of the responsibility framework of the analysed process. It encompasses a representation of the responsibility and its components, and the links between components.

STEP 2 actions: The actions performed at that step are composed of three sub-tasks.

Sub-task 1: Definition of each responsibility and transcription of it using boxes. Each box stands for a responsibility; it encompasses its accountabilities and its capabilities.

According to our case study, four responsibilities are extracted: *Postcard Sending*, *Drive Customers Relationship*, *Feedback Analysis*, and *Give Access Rights*. This is illustrated in figure 3.

Sub-task 2: For each responsibility, an analysis of the required capabilities is made, and is translated through a folded corner in the corresponding responsibility box. The same operation is made for the accountabilities.

Sub-task 3: This last sub-task consists in the definition of links between responsibilities components. Four kinds of links exist:

- *Delegation link* constituting the delegation of a responsibility's accountability toward another responsibility. This is illustrated by the accountability 5 of responsibility "*Drive Customer Relationship*",that delegates its accountability to the responsibility "*Feedback Analysis*".
- *Implication link* constituting for a responsibility's accountability the existence of another responsibility's capability. This is illustrated by the capability 2 "*Access Customer file*" of responsibility "*Feedback Analysis*" that implies the accountability 1 of the responsibility "*Give Access Rights*".
- *Contribution link* highlighting that one responsibility's accountability contributes to another accountability of the same responsibility. This is illustrated by the accountability 2 "*Print Envelopes*" which contributes to achieving the accountability 4 "*Send postcards*" of the responsibility "*Postcard Sending*".
- *Execution link* formalizing that a capability of a responsibility is necessary to execute an accountability of the same responsibility. This is illustrated by the capability 1 "*Customer Feedback*" of the "*Drive customer relationship*" responsibility that is necessary to achieve the accountability 4 "*Monitor the satisfaction of the clients*".

3.3 STEP 3 Responsibility's components links verification

This third step of the methodology is the first refining step. It aims at analysing the graphical representation of the process issued from step 2, depicting inconsistencies, and correcting the diagram to eliminate them.

STEP 3 input: Input of step 3 is the process graphical representation issued from step 2.

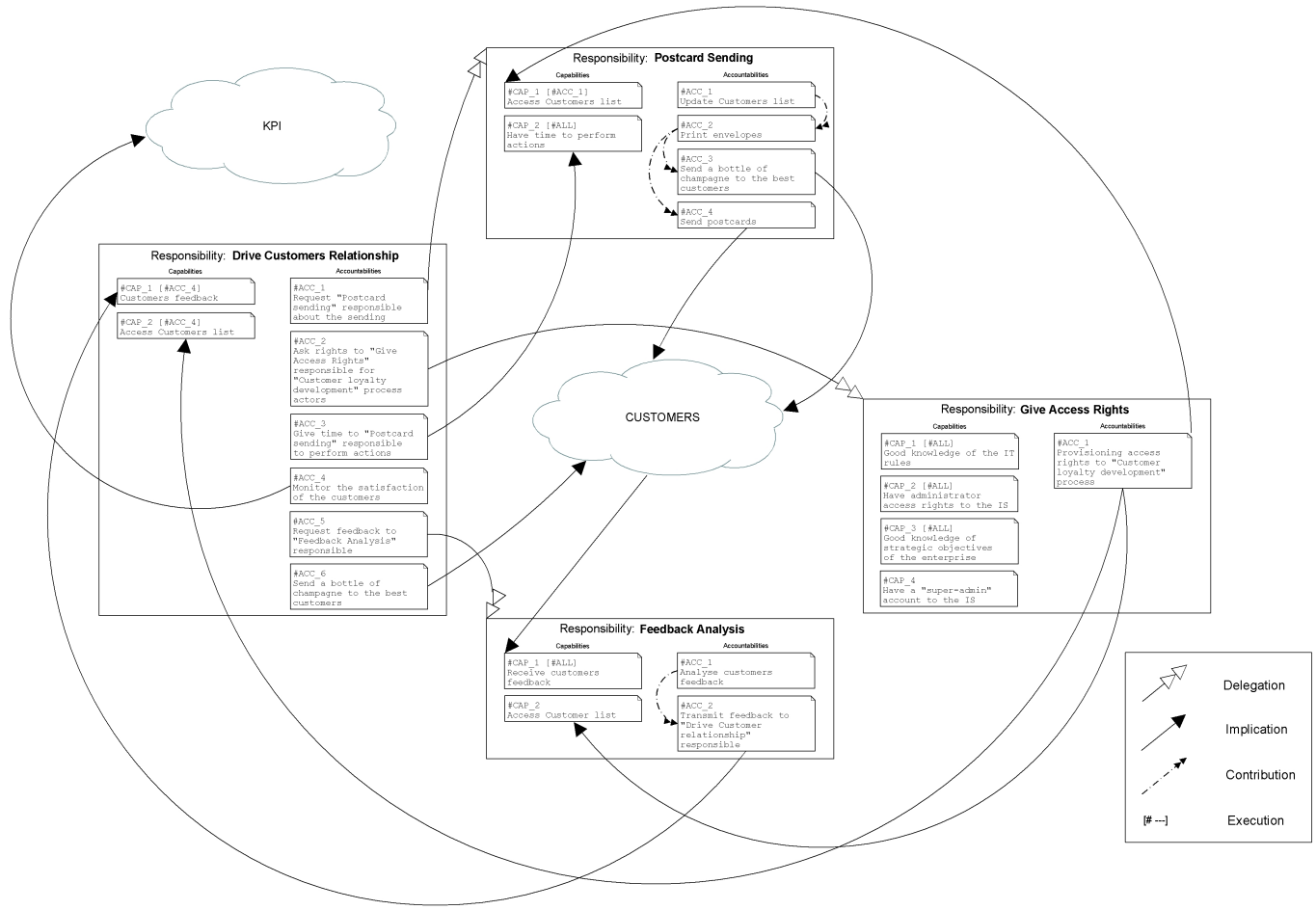

Fig. 3. Responsibility model for "Customer loyalty development process"

STEP 3 output: Output of step 3 is a graphical representation of the responsibility framework of the analysed process refined according to the components relationships.

STEP 3 actions: The actions performed at that step are composed of three sub-tasks.

Sub-task 1: Deep analysis of the capability components for each responsibility. The main objectives of this analysis are to detect and solve the problem of unnecessary capabilities. Capabilities may be unnecessary in the case of useless capabilities for the achievement of accountabilities of the same responsibility. This means that they do not have execution links. It is illustrated by the capability 4 "*Have a super admin account*" of the responsibility "*Give Access Rights*" that is not necessary for the accountabilities of that responsibility. To face this inconsistency, it is necessary to suppress the capability.

Sub-task 2: Deep analysis of the accountability components for each responsibility. The main objective of this analysis is to make sure of that all accountabilities are provided and exist in the model, and to assure that all accountabilities are necessary. Some accountabilities are not fully justified if:

- no link exist between the accountability with one or more capabilities in the process,
- no links exist between the accountability and another responsibility (no delegation),
- the accountability is not an outcome of the process.

It is illustrated by accountability 1 of "*Give Access Rights*" responsibility that "*Providing access rights to Customer loyalty development process*" is too large because it is not necessary for the "*Feedback Analysis*" to have that right.

Sub-task 3: Once accountabilities are verified, it is possible to check that all capabilities necessary for their achievement exist. This is illustrated by the accountability 2 "*Print envelopes*" of the responsibility "*Postcard Sending*" that may not be achieved because one capability misses that accountability: capability 3 "*Ability to print*".

3.4 STEP 4 Responsibility exceptions links verification

This fourth step of the methodology is the second refining step. It aims at analysing the graphical representation of the relation within the process, depicting inconsistencies, and correcting the graph to eliminate them if necessary.

STEP 4 input: Input of step 4 is the process graphical representation issued from step 3.

STEP 4 output: Output of step 4 is a graphical representation of the responsibility framework of the analysed process refined according the relationship between components.

STEP 4 actions: The activity of that step aims at detecting and correcting conflicts and incoherencies with regard to responsibility rules dictated by the enterprises, for example:

Delegation rules. If a responsibility is delegated from one actor to another, the enterprise should have rules to manage the delegation. These rules must be satisfied in the responsibility graph. Example of these delegation rules are: if a responsibility is delegated, all the capabilities necessary for it are also delegated and the accountability may be kept in the hand of the person that delegates, or in the hands of the person that is delegated to, but not with both persons at the same time. Some conflict may exist regarding that rule. E.g.: The responsibility "*Drive Customer relationship*" delegates the "*Sending Activities*" (sending postcards and bottles of champagne) but keeps, at the same time, its accountability of sending bottles of champagne. As a consequence, two responsibilities have the same accountability. If both persons achieve their accountability; it is possible that the action is performed twice.

Separation of duties. Some corporate rules may impose the separation of duty for some responsibilities. It is traditionally the case of the responsibility to order products, and the responsibility to validate the invoice of the product order.

Cardinality constraints. The responsibility graph also needs to be checked at this step for alignment with cardinality requirements. E.g.: the number of accountabilities handled by an equal responsibility is sometimes limited in order to avoid an unjustified increasing overload of work for a single person. This constraint must be balanced according to the work effort necessary for achieving each accountability.

3.5 STEP 5 Policy eliciting

This last step of our methodology aims at deriving policies from the responsibility model.

STEP 5 input: Input of step 5 is the process graphical representation issued from step 4.

STEP 5 output: Output of step 5 is a set of context dependant policies.

STEP 5 actions: The activity of that step aims at translating the responsibility graph into a given policy format.

Several derivations are possible. We can transform the responsibility model into an Process Framework like the one specified by the standard ISO/IEC 15504, *Information Technology – Process assessment*, (parts 1-5), 2003-2006 or an organisational model for multi-agent systems. In the following, we give a rapid overview of the possible transformation from an organisation model point of view.

MOISE^{Inst}, an Institution Specification Model for multi-agent systems, is defined in (Gâteau, 2007). This model is used to define normative organisations that are composed of:

- A structural specification (SS) defining the roles that agents will play, the links between these roles and the groups which agents playing roles should participate in and where interactions take place;
- A functional specification (FS) defining goals that have to be reached in the system;
- A normative specification (NS) clearly defining rights and duties of roles and groups on a mission (set of goals).

These specifications constitie the Organisational Specification (OS), i.e. the representation of the organisation independent of agents evolving in the system and becoming organised with regard to this OS. An Organisational Entity (OE) is an instance of the OS and is built from the set of agents that have adopted roles according to the SS of the OS, interacting within groups, and activating missions according to the current FS and norms (NS).

From the Responsibility Model point of view, we can consider Capabilities and Accountabilities of a Responsibility as right and duties that the person who will have the Responsibility will have to respect. A Capability is permission for the person to do something and an Accountability is an obligation to do something (in order to execute the whole process in which the Responsibility is defined).

In MOISE^{Inst}, missions (from the functional specification) define the achievement responsibility of collective goals gathered in a functional scheme. The NS defines permission and obligation (rights and duties) by specifying norms: a role is obliged to achieve a mission (a set of coherent goals). In this model, we define a role responsibility on a specific task (several goals indeed) by defining norms.

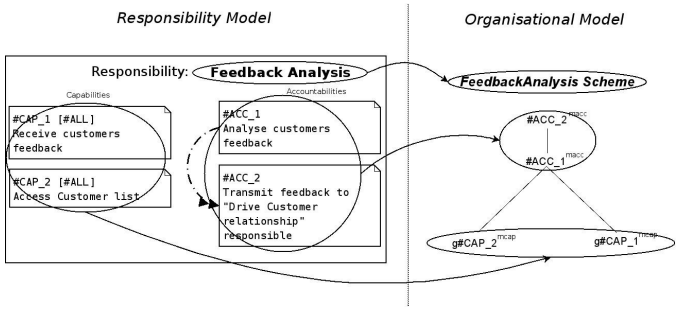

Fig. 4. Transformation of responsibility model into organisational model

A responsibility model resulting of the methodology depicted in this paper can be transformed into an organisational model and then be instantiated by a set of agents. As we can see on Fig. 4, each responsibility is represented by a functional scheme. Capabilities are leaf goals and belong to the *mcap* mission. Accountabilities are root goals (the last accountabil-

ity to achieve becomes the root goal in fact) and belong to *macc* mission. Links between capabilities and accountabilities become plan and structure the functional scheme: the goals #CAP_2 and #CAP_1 must be reached in order to reach the goal #ACC_1 which permit to reach the root goal #ACC_2 and then finish the scheme. At last, rights and duties are distinguished by associating missions with a deontic operator (permission or obligation) and a role in the organisational model. For instance, the role which will be responsible of the feedback analysis is obliged to achieve mission *macc* and permitted to achieve the mission *mcap*. We obtain two norms composing the NS and represented as follows:

- N01: Obl($Role_{feedback}$, macc)
- N02: Perm($Role_{feedback}$, mcap)

In the responsibility model, roles will be introduced in future work in order to instantiate the model by making a person play a role. In the organisational model, agents play roles and commit to missions (the commitment concept is also a part of the responsibility).

Once the OS is obtained, it could be instantiated by a set of agents playing roles and reaching goals regarding to the norms defined in the NS in order to make them execute the functional scheme, and to respect their responsibilities.

5. CONCLUSION AND FUTURE WORKS

Improving ICT governance is an important matter, especially in the current ongoing economic context. We propose in this paper to improve that field firstly by introducing our formalization of the responsibility of stakeholders involved in industrial activities and by elaborating an innovative responsibility model and, secondly, having the paper describe our methodology to define, structure and deploy the responsibility all around the company. This methodology, based on the concepts of our responsibility model (accountability, capability and commitment) is illustrated in this paper by the transformation of the responsibility model into an organisational model for multi-agent system.

In the proposed methodology, the definition of the responsibility component is performed in a sequential way. Future works will consist of improving the methodology with an iterative approach for refining the responsibility engineering (step 3 and step 4).

To illustrate that paper, the methodology has used an academic case study. The methodology is also under validation and improvement in a large international *Financial Services Provider* company using a real process case study.

This research was funded by the National Research Fund of Luxemburg in the context of SIM (Secure Identity Management - FNR/04/01/03) and TITAN (Trust-Assurance for Critical Infrastructures in Multi-Agents Environments, FNR CO/08/IS/21) projects.

REFERENCES

Antón, A., (1996), Goal-Based Requirements Analysi,. *Second ICRE 96*, Colorado Springs, USA.

Aubert, J., Gâteau, B., Incoul, C., Feltus, C., (2008) SIM: An Innovative Business-Oriented Approach for a Distributed Access Management, *ICTTA2008*, Damascus, Syria.

Crook, R., Ince, D., Nuseibeh, B., (2002) Towards an Analytical Role Modelling Framework for Security Requirements, Proc. of REFSQ'02, Essen, Germany.

Fernandez, E. B., Hawkins, J. C., (1997) Determining Role Rights from Use Cases, *ACM Workshop on RBAC*.

Fontaine, P. J., (2001) Goal-Oriented Elaboration of Security Requirements. M.S. Thesis, Dept. Computing Science, University of Louvain.

Gâteau, B., F eltus, C., Aubert, J., Incoul, C. (2008), An Agent based Framework for Identity Management: The Unsuspected Relation with ISO/IEC 15504, *IEEE International Conference on Research Challenges in Information Science (RCIS)*, Marrakech, Morocco.

Gâteau, B., (2008,) 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, defended in Luxembourg the 26th of June 2008.

Meir, H. L., Marios P. K. (1999), FirstSTEP process modeler a CIMOSA-compliant modeling tool, *computers in Industry*, Volume 40, Issues 2-3, Pages 267-277.

Neumann, G., Strembeck, M., (2002) A Scenario-driven Role Engineering Process for Functional RBAC Roles, *SACMAT 02*, Monterey, California, USA.

Qingfeng, H., Antón, A. I., (2003), A Framework for Privacy-Enhanced Access Control Analysis in Requirements Engineering, *REFSQ'03*, Austria.

Sandhu, R., Ferraiolo, D.,Kuhn, R., (2000) The NIST Model for Role Based Access, Control: Towards A Unified Standard , *Proceedings RBAC-00*, Berlin Germany, July 26-27 2000, pages 47-64.

Roeckle, H., Schimpf, G., Weidinger, R. (2000). Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization. *RBAC '00. ACM*, New York, NY, 103-110.

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

Yu, E. S., Liu, L. (2001). Modelling Trust for System Design Using the i* Strategic Actors Framework. *Workshop on Deception, Fraud, and Trust in Agent Societies Held During the Autonomous, Eds. Lecture Notes In Computer Science*, vol. 2246. Springer-Verlag, London, 175-194.