Refining the Notion of Responsibility in Enterprise Engineering to Support Corporate Governance of IT

Feltus, Christophe; Petit, Michaël; VERNADAT, François

Published in:
Proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'09), Moscow, Russia

DOI:
10.3182/20090603-3-RU-2001.00152

Publication date:
2009

Document Version
Early version, also known as pre-print

Refining the Notion of Responsibility in Enterprise Engineering to Support Corporate Governance of IT

Christophe Feltus[*,**], **Michaël Petit**[**], **François Vernadat**[***]

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

[**]*PReCISE Research Centre, Faculty of Computer Science, University of Namur, B-5000 Namur, Belgium, (e-mail: mpe@info.fundp.ac.be)*

[***]*Directorate for Information Technology and Telecommunications, European Court of Auditors, 12, rue Alcide de Gasperi, L-1615 Luxembourg, Luxembourg, (e-mail: francois.vernadat@eca.europa.eu)*

Abstract: Current insecure economic context and ongoing needs for more insurance between business partners advocate for a better alignment of the company with newly arising principles of corporate governance of IT. To contribute to that alignment, this paper first of all presents our generic responsibility model built on the concepts of Accountability, Capability and Commitment and combines that model with the CIMOSA framework. This CIMOSA enhancement enables the modeler to define easily usable and deployable enterprise policies throughout the company as well as throughout extended enterprises. Secondly, the paper permits to validate our responsibility model by analyzing and confronting it against the CIMOSA framework. Its advantages are illustrated with a model of the Supplier Tendering Process, a procurement process from the automotive industry.

Keywords: Process Model, Enterprise Engineering, Responsibility Model, IT Governance

1. INTRODUCTION

Corporate governance is becoming more and more necessary, in the current insecure economic context, to give the assurance to shareholders that the company will make profit and that its accounts are valid. The requirements of corporate governance define what is necessary to provide the assurance that the company will make the necessary investments, that its performance is aligned with its objectives, and that the organization is in conformity with standards and laws. The ISO/IEC 38500:2008 standard for corporate governance of information technology proposes a framework of principles for managers to use when evaluating, directing and monitoring the use of information technology in their organizations. This framework provides six guiding principles: Establish responsibilities, Plan to best support the organization, Acquire appropriately, Ensure performance when required, Ensure conformity with rules and Ensure respect for human factors. The first of those principles, "Establish responsibilities", aims at ensuring that individuals and groups within the organization understand and accept their responsibilities. Responsibility in the field of IT has already been largely investigated: first, because of IT security constraints and requirements, and second, in the field of software requirement engineering. IT security depicts responsibility mainly when it addresses access control. Indeed, to provide agents with rights and obligations to perform actions within an application or a component, main access control models use the concept of role to group agents based on their responsibilities, functions, geographic location, domain of work, etc. Some examples of those models are RBAC (Ferraiolo et al., 2001), UCON (Park et al., 2002) or OrBAC (Cuppens et al., 2003). However, the inconvenience already observed in large companies is that the engineering of these roles sometime leads to situations where their number is bigger than the number of agents (Cao et al., 2006).

Responsibility has also been the subject of research in the field of software requirement engineering. Indeed, this concept exists in a number of methods, e.g. GBRAM (Antón, 1996), KAOS (Fontaine, 2001) or i* (Yu et al., 2001). i* makes goal-oriented strategic modeling and analysis of requirements by using three mains concepts: actors, intentional elements, and links. Actors are described in their organizational setting and have attributes such as goals, abilities, beliefs, and commitments. Actors can be agents, roles, and positions. The disadvantage of those methods is that they are limited to concepts directly linked to the software requirement like a right or an obligation, without offering the possibility to be extended to wider concepts like the agent commitment. Other responsibility models exist, but are often linked to social or psychological areas, or are limited to very specific domains like (Sommerville et al., 2007) (Wright et al.,2004)

The formalization of the responsibility in enterprise architecture models (EAM) cannot avoid the new needs dictated by the governance requirements. To face that, we propose in this paper an enhancement of the Computer Integrated Manufacturing Open System Architecture - CIMOSA (Vernadat, 1995) (CEN/ISO, 2005) framework with three responsibility ele-

ments, and we depict how it improves the definition of rules and policies that govern the enterprise information system.

The first step of our methodology is the elaboration of a responsibility model. The next section introduces this generic responsibility model, its concepts and its advantages. The second step depicts the existing CIMOSA responsibility concepts and enhances the current version according to our generic responsibility model. This second step is presented in section 3. Finally, the last step is the illustration of the enhanced CIMOSA language in a case study based on a process reference model issued from the automotive manufacturing domain, in section 4.

2. RESPONSIBILITY MODEL

The new governance's constraints have led us, in our previous works, to elaborating a generic responsibility model. This synthetic and very pragmatic model has been designed mainly based on a review of the responsibility concept in the scientific literature (Feltus, 2008) and addresses the following three responsibility elements: capability, accountability and commitment (Fig. 1).

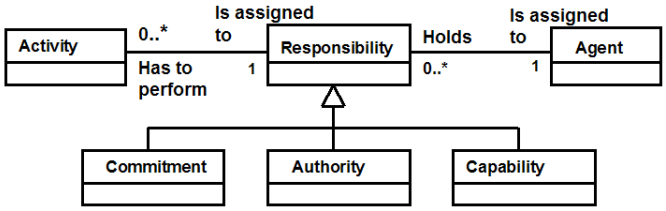

Fig. 1. Synthetic Responsibility Model in UML diagram

Each element of the model is defined based upon the review as follows:

Agent: (or employee) is a person external or internal to an organization, a system or a software component. An agent has to perform the activities he is responsible for. In other models, this concept is also called subject, actor or user. For facilitating their management, those agents are often grouped together based on their common properties and attributes. As previously explained in the literature overview, the most famous type of classification is based on the concept of role, but variations exist, such as for example the team, the hierarchy, or some geographical constraints or domain of work, etc. Some examples of those models are RBAC, UCON or OrBAC.

Responsibility: a lot of definitions of responsibility exist. However, commonly accepted responsibility definitions encompass the idea of "*having the obligation to ensure that something happens*". Moreover, the literature review highlights that being responsible implies that it is necessary to have one or many capacities, accountabilities and commitments. But at the opposite, one commitment and one accountability always relate to one responsibility, whereas one capability may be attached to many responsibilities.

Activity: is an operation performed by an agent responsible for it. This concept does not exist in the realm of access control models describing right or/and obligation needed to perform an operation. For example, the right to read a document or the obligation to satisfy conditions before executing an operation do not make the activity that requires them explicit. By contrast, "activity" is a main concept in requirement engineering. For example, in Tropos (Fuxman et al., 2001), a goal may be achieved by fulfilling an activity. The relation between agent, responsibility and activity can be stated as: "there is one and only one agent responsible for one activity, one agent may have many responsibilities and one responsibility may apply on many activities".

Accountability: is a concept that exists mainly in engineering methods and that appears through the obligation to perform an activity or an action. This concept describes the state of being accountable for the achievement of the results of an activity. Recent laws, like the Public Company Reform and Investor Protection Act of 2002, known under Sarbanes-Oxley and the Basel II requirements for the financial institutions, have put forward the need of more accountabilities in the hands of agents and more precisely the CEO and CFO. This accountability represents an obligation to be kept informed of whether or not accounts of the enterprise are valid.

Commitment: is the moral engagement of an agent to fulfill an activity and the assurance that he will do it in respect of an ethical code. Commitment is the most infrequent concept. In the field of access control, traditional policy model such as RBAC do not address this concept. In requirement engineering ***i**** partly introduces it (e.g. when defining dependency as an "agreement" between two actors).

Capability: describes the required qualities, skills or resources to perform an activity. Capability is a element that is part of all security models and methods, and is most frequently declined through definitions of access rights, authorizations or permissions.

The advantages of such a model are important for four reasons:

1. It leads information to be aligned with the principle 1 of the ISO/IEC 38500:2008 standard: Establish clearly understood responsibilities for IT.
2. The accountability is bound to the agent rather than to a group of agents (like in others models). The agent is more involved personally, and more concerned by the activity he has to perform, because the result is not shared anymore.
3. It addresses the commitment of agents that are responsible for performing activities, and consequently increases the ethics of the business in general.
4. It allows checking that the right capability is assigned to the right agent. This advantage guarantees firstly that the agents receive the minimum privileges necessary for achieving their activities and consequently, it decreases the vulnerability of the system and secondly, that no one has capabilities that are not required (confidentiality and security requirements and conformance to laws such as privacy regulations).

Moreover, due to its simplicity, this model has the ability to be adapted to extended enterprise (Bolseth, 2005) and in a large number of fields or models.

3. CIMOSA AND THE RESPONSIBILITY MODEL

The main models of EAM are CIMOSA, the Zachman framework (Sowa et al., 1992) or TOGAF (Togaf, 2007). GERAM is an integrated model at the conceptual layer that integrates CIMOSA, GRAI/GIM, and PERA. Those models are often structured according to abstraction layers (conceptual, organizational or technical) and according to views (functional, economic, resource and/or organizational). As a consequence they are, like CIMOSA, often a concentration of several models (see Fig. 2, borrowed from (Vernadat, 1995)).

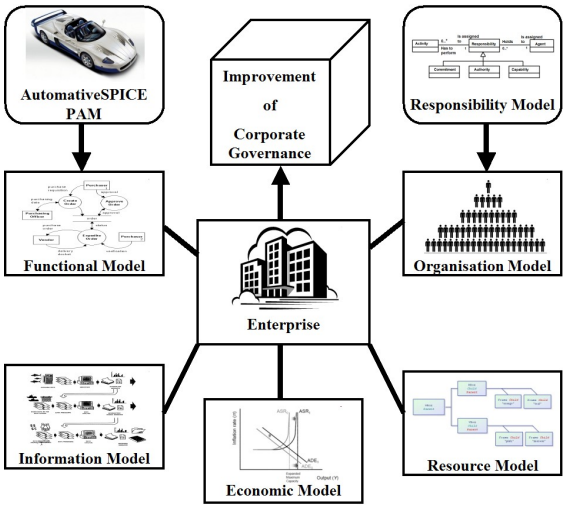

Fig. 2. EAM models are federations of others models

3.1 Analysis of CIMOSA basic responsibility concepts

CIMOSA encompasses (Vernadat, 1995):

1. A *Modeling Framework* that provides semantic unification of the concepts. It contains 3 axes (CIMOSA Cube):

- the GENERATION (with 4 views : Function, Information, Resources and Organization),
- the INSTANTATION,
- the DERIVATION.

2. An *Integrating Infrastructure* that supports model execution and acts as a common IT execution platform.

3. The *System Life Cycle* that describes the major phases in the engineering of a CIMOSA system.

The responsibility concepts of our model (section 2) are mainly addressed in the Modeling Framework. By analyzing it, we see that an Agent is a Functional Entity (i.e. an active resource), is represented in the Resource View, and appears when resources are derived from the requirements definition to the implementation description. The responsibility is represented in the Organizational View. Indeed, this view is composed with Organization Units that are low level decision centers or work positions assigned with responsibilities and authorities, and Organization Cells that are higher level decision centers with a manager, responsibilities and authorities. Those cells are consequently structuring the organizational units into larger entities at different responsibility levels. This information is completed in (Mauchan, 2007) that presents a class diagram of CIMOSA model and highlights how the Organizational Unit is responsible for the process and how this process is composed of activities (or task) that need capability. In addition to the responsibility element, the CIMOSA Modeling Framework introduces the concept of Authority.

Capability in the current CIMOSA framework is defined as a resource element of the Resource View. This element is linked to and needed in the activity concept of the Function View (required capabilities/competencies) and is linked and provided by the agent concept of the Resource View (provided capabilities/competencies). In (Vernadat, 2004), Capability set is defined as a set of capabilities (i.e. technical characteristics) for technical agents or a set of competencies (i.e. skills) for human agents.

The Commitment is not explicitly taken into account in CIMOSA.

The Accountability of an agent regarding an activity is the obligation to perform that activity and to obtain the expected results. Although that activity is defined by both: the results (control outputs, function outputs and resources outputs) and the agents that perform it (input resource), no explicit link exists between the accountability of that agent and the activity.

Fig. 3 summarizes the CIMOSA's responsibility concepts at a requirement level.

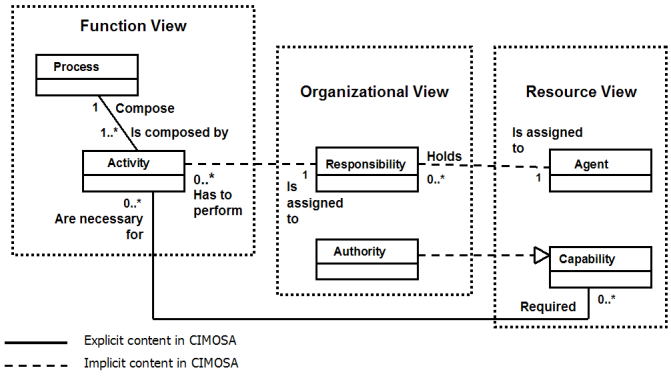

Fig. 3. Basic CIMOSA responsibility model UML Diagram

3.2 Enhancement of the CIMOSA framework and of our responsibility model

The current representation of the responsibility in the CIMOSA model explained in section 3.1 can be improved by incorporating it into our responsibility model presented in section 2. Fig. 4. illustrates that and represents the integration of that concept at a requirement level:

The *responsibility* concept is explicitly introduced in the Organization view. It is linked to the activity to be performed and to the agent responsible for it. By doing so, we provide the possibility to distinguish the agent that has the required capabilities/competencies to perform the task and the agent that will be accountable of it. This modification will provide facilities to manage the delegation of activities or the possibility to easi-

er replace an agent by another. It introduces as consequence the notion of role (Ferraiolo et al., 2001) in the CIMOSA Framework.

The *capability,* while remaining an element from the Resource View, is no more linked to the activity but it is linked to responsibility. With that modification and in the perspective of being at the requirement level, the agent is responsible if and only if he has the capabilities to perform the activity.

The *commitment* concept will be introduced in the organizational view as one of the elements that compose the responsibility.

The *accountability* will formally exist as a element that composes the responsibility. With that concept, it is possible to identify which agent is accountable of which activity.

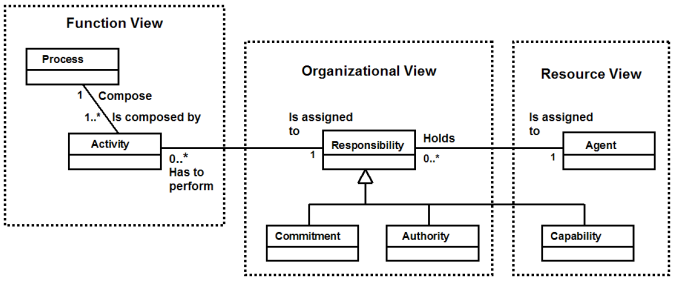

Fig. 4. Improved Responsibility Model UML Diagram

The combination of the CIMOSA model with the Responsibility model is integrated in the CIMOSA language with a new responsibility component defining the responsibility's elements of the ResourceInput (agent) that perform the activity (Fig. 5)

ResourceInput: *Name of ressource*
Responsibility:
Accoutable : *list of accountabilities*
Capability : *list of capabilities*
Commitment :*list of commitments*

Fig. 5. CIMOSA updated language

In parallel to the enhancement of the CIMOSA model, the analysis permits to understand a new concept: the *Authority.* The Authority will be introduced in the responsibility model as an instance of the Capability. Indeed, the definition of this concept is "the power to command and control others agents". That means, according to our definition of section 2, a well precise type of right.

4. CASE STUDY

The advantages of the enhanced CIMOSA can be illustrated in various areas. We demonstrate, in that case study, the advantages for the management of access rights applied for a process from the automotive industry: the Supplier tendering process (Fig. 7) borrowed from the AutomativeSpice Process Assessment Model (PAM) (www.automotivespice.com). This PAM has been developed by consent of the car manufacturers within the Automotive Special Interest Group of the joint procurement Forum / SPICE User Group. This process is structured according to the ISO/IEC 15504, "Information Technology – Process assessment" standard structuring a process with the following elements: *name, purpose, outcomes* (issues of the process implementation) and *base practices* (needed to achieve outcomes). Additionally, the process description following ISO/IEC 15504 encompasses *output* and *input work products*. The current version of ISO/IEC 15504 does not address the responsibility. However, in the associated maturity model, the Process Capability Indicator required to be at level 2 describes that "*interfaces between the involved parties are managed to ensure […] clear assignment of responsibility*." By using the CIMOSA formalism, it is possible to represent the Supplier tendering process as a flow of base practices (equivalent to activities) as shown in Fig. 8.

Output Work Products
02-01 Commitment / agreement [Outcome 6]
08-12 Project plan [Outcome 4]
12-04 Supplier proposal response [Outcome 5]
13-04 Communication record [Outcome 1, 6]
13-15 Proposal review record [Outcome 3, 4]
13-19 Review record [Outcome 2]

Fig. 6. Supplier tendering process Output Work Product

Process ID	SPL.1
Process Name	Supplier tendering
Process Purpose	The purpose of Supplier tendering process is to establish an interface to respond to customer inquiries and requests for proposal, prepare and submit proposals, and confirm assignments through the establishment of a relevant agreement / contract.
Process Outcomes	As a result of successful implementation of this process: 1) a communication interface is established and maintained in order to respond to customer inquiries and requests for proposal; 2) request for proposal are evaluated according to defined criteria to determine whether or not to submit a proposal; 3) the need to undertake preliminary surveys or feasibility studies is determined; 4) suitable staff are identified to performed the proposed work; 5) a supplier proposal is prepared in response to the customer request; and 6) formal confirmation of agreement is obtained.
Base Practices	**SPL.1.BP1: Establish communication interface.** A communication interface is established and maintained in order to respond to customer inquiries or requests for proposal. [Outcome 1] **SPL.1.BP2: Perform customer enquiry screening.** Perform customer enquiry screening to ensure validity of contract, ensuring the right person is quickly identified to process the lead. [Outcome 1] **SPL.1.BP3: Establish customer proposal evaluation criteria.** Establish evaluation criteria to determine whether or not to submit a proposal based on appropriate criteria. [Outcome 2] **SPL.1.BP4: Evaluate customer request for proposal.** Requests for proposal are evaluated according to appropriate criteria. [Outcome 2] **SPL.1.BP5: Determine need for preliminary pre-studies.** Determine need for preliminary pre-studies to ensure that a firm quotation can be made based on available requirements. [Outcome 3] **SPL.1.BP6: Identify and nominate staff.** Identify and nominate staff with appropriate competency for the assignment. [Outcome 4] **SPL.1.BP7: Prepare supplier proposal response.** A supplier proposal response is prepared in response to the customer request. [Outcome 5] **SPL.1.BP8: Establish confirmation of agreement.** Formally confirm the agreement to protect the interests of customer and supplier. [Outcome 6] NOTE.1: The nature of the commitment should be agreed and evidenced in writing. Only authorized signatories should be able to commit to a contract.

Fig. 7. Supplier tendering process

In that representation of Fig. 8., some elements can be extracted from the AutomativeSpice PAM, whereas others need to be completed based on the domain knowledge.

Element from AutomativeSpice PAM :

1.CIMOSA's *Activities* are associated to ISO/IEC 15504's *base practices* and are represented using rectangles and the link between base practices by arrows.

2.In previous work (Gâteau et al., 2008), we have made a mapping between ISO/IEC 15504 and the responsibility model. This mapping has led to the definition of links between

- first, the capability concept of the responsibility model and the *Input Work Product* of the PAM;
- and second, the accountability concept of the responsibility model and the *Output Work Product*.

Only *Output Work Products* exist in the AutomativeSpice PAM. They are listed in Fig. 6.

Elements not contained in AutomativeSpice PAM:

1.We propose an implementation sequence for the base practices (activity)

2.One particularity of the AutomativeSpice PAM process is that there are no *Input Work Products* defined. We introduce an example of input work product for a better understanding of the enhanced model. We also add some examples of commitments.

3.We focus our example on a particular *base practice: SPL1 BP7 Prepare Supplier Proposal Response* achieved through the agent *SuppPropAgent*. According to CIMOSA, the agent represented by a rounded rectangle owns the responsibility.

4.Finally, this *base practice* is subdivided into 3 operations that are *ReceiveCfP, Write Proposal* and *Check Proposal*.

Based on that flow of activities of Fig. 8, it is possible to describe the *SupplierTendering* process (Fig 9) and the *SPL1 BP7: Prepare supplier proposal response* activity (Fig. 10) in the CIMOSA Language through the use of construct templates. In that description, the elements of AutomativeSpice are written in boldface and the additional examples are written in italic.

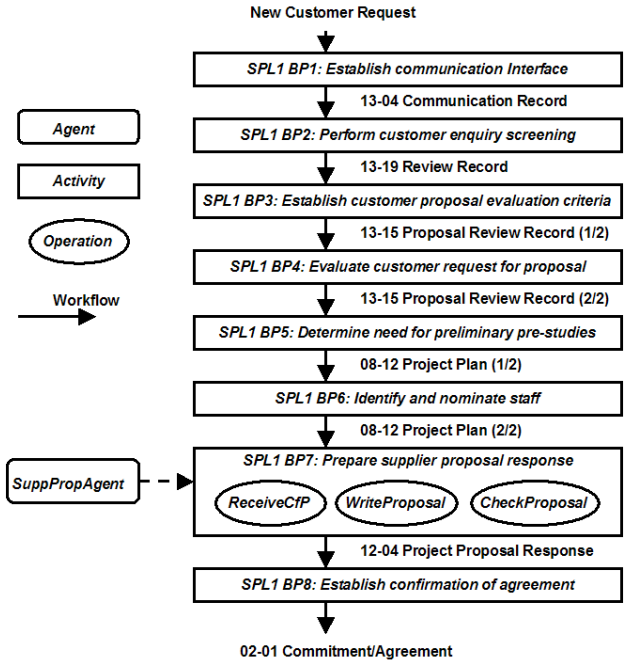

Fig. 8. *Supplier Tendering* process flow of activity

PROCESS **Supplier Tendering**
TriggeringEvents: **NewCustomerRequest**
ProcessBehaviour:
WHEN (START WITH **NewCustomerRequest**) DO **SPL1 BP1**
WHEN (ES (**SPL1BP1**) = **13-04 CommunicationRecord**) DO **SPL1 BP2**
WHEN (ES (**SPL1 BP2**) = **13-19 Review Record**) DO **SPL1 BP3**
WHEN (ES (**SPL1 BP3**) = **13-15 Proposal Review Record (1/2)**) DO **SPL1 BP4**
WHEN (ES (**SPL1 BP4**) = **13-15 Proposal Review Record (2/2)**) DO **SPL1 BP5**
WHEN (ES (**SPL1 BP5**) = **08-12 Project Plan (1/2)**) DO **SPL1 BP6**
WHEN (ES (**SPL1 BP6**) = **08-12 Project Plan (2/2)**) DO **SPL1 BP7**
WHEN (ES (**SPL1 BP7**) = **12-04 Project Proposal Response**) DO **SPL1 BP8**
WHEN (ES (**SPL1 BP8**) = **Commitment/Agreement**) DO FINISH

Fig. 9. PROCESS *Supplier Tendering*

ACTIVITY **SPL1 BP7: Prepare Supplier Proposal Response**
FunctionInput : Supp*PropFile*
ControlInput: **ProjectPlan**
ResourceInput: *SuppPropAgent*
FunctionOutput: **SuppPropResp**
ControlOutput: {*new-SuppProp*}
MeanDuration: *3 weeks*
ActivityBehaviour:
{*SuppPropAgent.ReceiveCfP(CfP)*;
SuppPropAgent.WriteProposal (CfP, *CustomerID*);
SuppPropAgent.CheckProposal (*Customer*, *ProposalRespons*)}

Fig. 10. ACTIVITY *SPL1 BP7: Prepare Supplier Proposal Response*

Fig. 10 shows that at the activity level, *ResourcesInput* clarifies the resource that is "responsible" for the achievement of the activity. This resource is defined as FUNCTIONAL ENTITY *SuppPropAgent* in Fig. 11.

FUNCTIONAL ENTITY *SuppPropAgent*
Type: *Human*
InRealLife: *M.S. Johnson*
ObjectView: *OV-08* (* resource description*)
Location: *Bldg#3-Office#120*
Operations:
ReceiveCfP (IN **08-12 Project Plan**, OUT: *Acknowledgement*);
WriteProposal (IN *CustomerID* and *CfP*, OUT *Unchecked proposal*);
CheckProposal (IN *Unchecked proposal*, OUT **12-04 Project Proposal Response**);

Fig.11.: FUNCTIONAL ENTITY *SuppPropAgent*

Fig. 9, 10 and 11 highlight that the current CIMOSA language does not permit a formalization of the three responsibility elements that we have attached to the CIMOSA model (Fig 4.). To enhance the language accordingly to the enhanced model, we propose to add to the ACTIVITY description those three responsibility elements in the *ResourceInput* (or Agent) definition according to the format illustrated in Fig. 5.

To generate that enhanced language, we use an open source tool (eGroupware) whose particularity is to support business activity based on a process approach. Two new functions have been developed for that tool in order to illustrate our research: the formalization of responsibility, and the provision of access rights to network and software components according to the responsibilities. For our case study and in line with ISO15504, each component of the *Supplier Tendering* process is recorded in the tool: *name*, *purpose*, *outcomes*, *base practices (activity)* and *operations,* and according to our responsibility model: *capability, accountability* and *commitment* (Fig. 5.).

Based on the information recorded, an enhanced description of the *PrepSuppResp* activity containing the responsibility elements is generated using the CIMOSA language (Fig. 11.), and is used to provision access right over the network and the software components. For that activity, the requirements issued from the process assert that *SuppPropAgent* needs access to the *CustomerID* and the *CfP* to be able to achieve the task he is responsible for. This functionality is already possible with a specific eGroupware module (Gâteau et al., 2008) that provision access rights to network components.

The following case study also highlights others advantages like the possibility to have process sharing for extended enterprise activities like the procurement or the eBusiness. Indeed, it is possible to have a process shared between two companies and

to have responsibility accordingly defined. In that case, both companies firstly agree on capabilities strictly need to achieve the process and on the expected commitment and accountabilities. Secondly, they exploit the responsibility common definition to engineer their own access rights.

ACTIVITY **SPL1 BP7: Prepare Supplier Proposal Response**
FunctionInput : Supp*PropFile*
ControlInput: **ProjectPlan**
ResourceInput: *SuppPropAgent*
Responsibility:
Accountable : *ReceiveCfP Acknowledgment, EvaluateCfP Status, Write SuppPropResp and CheckProposal Message*
Capability : Access to *CustomerID, Access to CfP, Training, Time and Necessary tools*
Commitment : *to respect ethical code*
FunctionOutput: **SuppPropResp**
ControlOutput: {*new-SuppProp*}
MeanDuration: *3 weeks*
ActivityBehaviour:
{*SuppPropAgent.ReceiveCfP(CfP)*;
SuppPropAgent.WriteProposal (CfP, *CustomerID*);
SuppPropAgent.CheckProposal (*Customer*, *ProposalRespons*)}

Fig. 11. Enhanced ACTIVITY *SPL1 BP7: Prepare Supplier Proposal Response*

5. CONCLUSIONS

Corporate IT governance requires having the responsibilities clearly defined and aligned with the business process. The literature review shows that responsibility is a concept modelled using accountability, capability and commitment elements. CIMOSA does not systematically integrate all facets of those elements. As consequence, this paper enhances the CIMOSA framework and language with a generic and pragmatic responsibility model. The main advantage of this enhancement is that it proposes a solution to exploit the performance and the facilities of CIMOSA framework, while in parallel covering governance requirements that are: better visibility of each agents' responsibility, better security, and better business/IT alignment.

The enhanced CIMOSA framework is illustrated in the field of the management of access right applied for the Supplier tendering process borrowed from the automotive industry. The advantages of the CIMOSA enhancement are an expression of the access right strictly needed for an agent to achieve an activity according to the requirements issued from the business process description.

Finally, the analysis and the confrontation with CIMOSA have also contributed to validate and expand the responsibility model with the CIMOSA's perception of the responsibility.

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 Analysis. *ICRE 96*,Colorado Springs, USA.

Bolseth, S. (2005), A Process Model for the Extended Enterprise, *EurOMA 2005*, Hungary.

Cao, X., Iverson, L. (2006). Intentional access management: making access control usable for end-users. *Second Symposium on Usable Privacy and Security*, vol. 149. ACM, New York, 20-31.

CEN/ISO (2005). Enterprise Integration Constructs for enterprise modelling, prEN 19440, CEN/TC 310 and ISO/TC 184, BSI secretariat, London, UK.

Cuppens, F., Miège, A. (2003), Modelling contexts in the OrBAC model. 19th Annual Computer Security Application Conference, Las Vegas, USA.

Feltus, C. (2008), Preliminary Literature Review of Policy Engineering Methods - Toward Responsibility Concept, *ICTTA 2008*, Damascus, Syria.

Ferraiolo, D. F., Sandhu, R., Gavrila, S., D. Kuhn, R., Chandramouli, R. (2001), Proposed NIST Standard for Role-Based Access Control, *ACM Transactions on Information and System Security*, 4 (3), 224-274.

Fuxman, A., Pistore, M., Mylopoulos, J., Traverso, P. (2001), Model Checking Early Requirements Specifications in Tropos, *In Proceedings of RE 01*, 174-181.

Fontaine, P.J. (2001), Goal-Oriented Elaboration of Security Requirements. M.S. Thesis, Belgium.

Gateau, B., Feltus, C., Aubert J., Incoul, C. (2008), An Agent-based Framework for Identity Management: The Unsuspected Relation with ISO/IEC 15504, *RCIS 2008*, Morocco.

Mauchan, M. (2007), thèse «Modélisation pour la simulation de chaines de production de valeur en entreprise industrielle comme outil d aide à la décision en phase de conception / Industrialisation»

Sommerville, I., Storer, T., Lock, R. (2007) Responsibility modelling for contingency planning. In Workshop on Understanding Why Systems Fail, Contingency Planning and Longer Term Perspectives on Learning from Failure in Safety Critical Systems.

Sowa, J.F., Zachman, J. A. (1992), Extending and Formalizing the Framework for Information Systems Architecture, *IBM Systems Journal*, 31/3.

Togaf (2007), The Open Group Architecture Framework (TOGAF 8.1.1 'The Book'), *Van Haren Publishing*.

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

Vernadat, F. B. (2004), Enterprise Modelling: Objectives, constructs & ontologies, *Tutorail EMOI-CaiSE Workshop*, Latvia.

Yu, E. S., Liu, L. (2001). Modelling Trust for System Design Using the i* Strategic Actors Framework. Eds. Lecture Notes In Computer Science, vol. 2246. Springer-Verlag, London, 175-194.

Wright, P.M, White, K., Gaebler-Spira, D. (2004) Exploring the relevance of the personal and social responsibility model in adapted physical activity: A collective case study. Journal of Teaching in Physical Education, 23(1), 71-87.