Building a Responsibility Model Including Accountability, Capability and Commitment

Feltus, Christophe; Petit, Michaël

Published in:
Proceedings of the The Fourth International Conference on Availability, Reliability and Security ("ARES 2009 - The International Dependability Conference"), Fukuoka, Japan

DOI:
10.1109/ARES.2009.45

Publication date:
2009

Document Version
Early version, also known as pre-print

Building a Responsibility Model Including Accountability, Capability and Commitment

Christophe Feltus and Michaël Petit

***Abstract*—This paper aims at building a responsibility model based on the concepts of Accountability, Capability and Commitment. The model's objectives are firstly to help organizations for verifying the organizational structure and detecting policy problems and inconsistency. Secondly, the paper brings up a conceptual framework to support organization for defining their corporate, security and access control policies. Our work provides a preliminary review of the researches performed in that field and proposes, based on the observations, an UML responsibility model and a definition of all its concepts. Thereafter, to propose a formal representation of the model, we have selected the suitable language and logic system. The analyze highlights that an important variable is whether the responsibility is perceived at a user or at a company level**

***Index Terms*— Responsibility, Capability, Commitment, Accountability, Access control, Right management, Formal system, Security management.**

I. Introduction

It is notable that nowadays, the responsibility committed from a person to perform a task is an aspect that has for a long time remained overshadowed and that nevertheless appears to be from a major interest. The perception of responsibility has often been limited to a combination of rights and obligations. However current business (for example in the financial sector) demonstrates that the moral aspect is improvable and that taking care of that matter would avoid in some cases malfunctions of the system. In practice, responsibility is most often translated through policies. It exists much definition of policy. For our work, we prefer the definition of policies from [31] that is *Policies are rules that governe the choice in behaviour of a system. Security policies define what actions are permitted or not permitted for what or for whom, and under what condition (...)* This definition is interesting it that, even if it is coming from a low level context, it sounds applicable to the high level one such as the management.

Based upon the above observations, the first objective of that paper is to perform a literature review of policy models and engineering methods to identify the main policy's concepts. From that literature review, a model of responsibility is elaborated and integrates main responsibility's concepts and main relationships between those concepts. The specificity of that model is its genericity that permits in the first hand to integrate policies from all abstraction layers of the company, e.g.: IT policy are declined from Corporate policy, and in the second hand, to be compatible to policy from different domains of the company. E.g.: IT policy, organizational policy, or security policy. Finally, we introduce a formalization of the concepts using logic system. The formalization main objectives are to propose a basic logic framework for defining all concepts and, by using that framework, verifying organizational structure and detecting policy problems and inconsistency.

Our work will be based on the hypothesis that this responsibility is composed by the tuple (Capability, Accountability, Commitment). Our previous work [2] has introduced principal semantic characteristics about those three concepts and has brought formalizing elements using standard logics.

The work is introduced by Camerer's observations over research in the field of policy. These observations presented in the next section provide a precious warning we have to take care for our research. Section 3 reviews the concepts of responsibility in access control models and in engineering methods. Section 4 formalizes the responsibility and its concepts with an UML model and presents the selection of a formal system. Section 5 introduces future works around the formalization and section 6 concludes.

II. From Business to Security Policy

Before going ahead in the literature review, let make a hook to understand the analysis made by Camerer [9] on researches in business policy and strategy. An important observation in his work is that: *« There are at least three symptoms of the disease causing the queasy dissatisfaction with policy research:*

1. *Concepts are often ambiguous and their definitions are not agreed upon;*
2. *Checklists or theories are rarely tested, and never tested directly against competing theories and*
3. *Theories do not 'cumulate' or built upon previous theories as they should.*

These three deficiencies are a result of the way policy research is typically done."

Camerer explains that policy research should evolve from

Manuscript received September 30, 2008.

Christophe Feltus is with the Center for IT Innovation, Public Research Centre Henri Tudor, 29, Avenue John F. Kennedy, L-1855 Luxembourg-Kirchberg, Luxembourg (e-mail: christophe.feltus@tudor.lu)

Michaël Petit is with the Computer Science Department, University of Namur, B-5000Namur, Belgium (e-mail: mpe@info.fundp.ac.be)

an inductive to a deductive approach. He argues that induction contribute to an unproductive debate about variable definitions and to a lack of testability and failure of theory. Unlikely, his conviction is that deductive models can express hypotheses in a language that is more amenable to progressive debate. This point of view is a precious warning we have to take into account before beginning our researcher in that it may prevent us to perpetrate the same mistakes. This warning is moreover substantial because of the subjective character of the moral aspect under focus in our research. In his work, Camerer only addresses business policy. Therefore, this consideration needs to be adapted according to our research's context and it is consequently necessary to clarify the relation that exists between business policies and IT policies. Wies [10] shows the links between high and low-level policies. He depicts the variation of importance of the technology and the business aspects when translating high-level onto low-level policies. High-level policies tend to focus on business aspects whereas low-level policies focus on technology aspects. Although they are spread on different abstraction layers of the policy hierarchy, business policies and IT policies should be consistent because both should be derived from (management and/or IT) goals and hence embody (management and/or IT) strategy's aspects. Rifaut et al. [11] propose to use Goal-Oriented Requirements Engineering (GORE) methods to define goals, strategies and policies. Rifaut explains that these methods can be used to analyze and model systems at all organizational level, from business models down to architectures, see Fig. 1. He argues that the four artifacts that are objectives, policies, strategies and indicators may be globally considered as objectives and that consequently, low level objectives contributes to achieve higher level one. E.g.: Having access control management contributes to have a performance IT security and having a performance IT security contribute to have a good corporate governance.

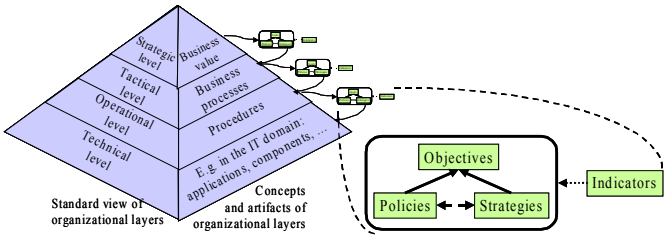

Fig. 1. GORE model for Policy refinement

Based on the previous assumption that there exist links between policies from different layers, further analysis of the literature has been conducted to depict the principal elements that compose the policy concept.

III. Responsibility literature review

It is rapidly observable when analyzing policy literature that a very large amount of authors show interest in that concern. Consequently, a number of surveys have already been produced in that domain [15][17][18] and [19] but none has targeted the responsibility through the tuple (Capability, Accountability, Commitment).

Despite that proliferation of works, it is noteworthy that up to now there does not really exist a distinction between works addressing access control model, policy model, role engineering and permission/policy engineering. Based on that assumption, it appears meaningful for apprehending that topic to clarify this point and to highlight the existing dichotomy between model and method. To perform our review, we will base our analysis on a commonly accepted idea that a model or conceptual model is a representation designed to show the structure of a system or concept and that (at least in our case), a method is a body of techniques for collecting data necessary to instantiate the conceptual model. Consequently and as illustration, the Role-Based Access Control (RBAC) model [1] proposes a structure for providing access based on role whereas role engineering [3] and [4] is a method aiming to define roles to instantiate the conceptual model. Identically, policy may also be modeled and there exists a proliferation of methods to instantiate it. These methods may be classified according to the technique they use. We propose to start with methods based on Requirements Engineering (RE) and to continue with a list of others. Moreover, it is more frequent to read paper targeting policy language than policy model. Those policy languages are innumerable and spread over the entire organizational model layers. Most famous of them are Ponder [5], Policy Description Language [6], Security Policy Language [7], and Rei [8]. Amazingly, the policy model used to support the policy expression by the policy language remains rarely specified. This review presents successively the responsibility through access control models and engineering methods. The components of the responsibility s tuple are :

- *Capability*: which describes the quality of having the requisite qualities or accesses to resources to achieve a task;
- *Accountability*: which describes the state of being answerable about the achievement of a task;
- *Commitment*: which is the engagement of a stakeholder to fulfil a task and the assurance he will do it.

These definitions are refined through the description of these concepts in section 4.

Responsibility in the field of IT has already been investigated because of IT security constraints and requirements firstly, and of software requirement engineering secondly. IT security depicts responsibility mainly when it addresses access control. Indeed, to provision employees with right and obligation to operate over an application or a component, main access control model use the concept of role to group employee based on their responsibility, function, geographic location, domain of work, etc. Some examples of those models are the Mandatory Access Control, RBAC [10], UCON [11], OrBAC [12], etc. However, the inconvenient already observed in large company is that the engineering of that roles leads sometime to situations where the amount of roles is bigger than the amount of employees.

Responsibility has also been subject of research in the field of software requirement engineering. Indeed, this concept is

centric for a large amount of methods like I*[13]. I* makes goal-oriented strategic modeling and analysis of requirements by using three mains concepts that are: 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. Agents are concrete actors, systems or humans, with specific capabilities. The inconvenient of those methods is that they are limited to concepts directly linked to the software requirement like the right or the obligation without offering the possibility to be extended to wider concepts like the Commitment.

The state of the art of policy concepts introduces a review offour main recognized access control models: Mandatory Access Control (MAC), Discretionary Access Control (DAC), Role-based Access Control (RBAC) and Usage Control Model (UCON).

Our survey has also covered others approaches that due to the size of the paper are not presented here. In summary we may observe that firstly, some concepts are commonly accepted, such as right, role and obligation. Definition of the two firsts concepts are scarce. Only one definition has been found for the concept of "right": the right (or permission) is explicitly granted to a subject to access an object in a specific mode, such as read or write [1]. For the concept of "role", only one definition has also been found in [13]. The concept of obligation is subject to more debate. For Bettini et al. [14], obligations are conditions or actions that must be fulfilled either by the users or the system after a decision. In [1], Sandhu et al. define obligations as requirements that have to be fulfilled by the subject for allowing access. Crook et al. [15] extend the notion of obligation to obligation policy that relate to actions that must be carried out on targets by subjects when a predefined event occurs and Haley et al. in [16] define it as what actions must be taken before access can be granted.

TABLE 1.
AC MODEL AND RESPONSIBILITY'S CONCEPTS

	MAC	**DAC**	**RBAC**	**UCON**
Subject	Yes	Yes	Yes	Yes
Object	Yes	Yes	Yes	Yes
Group	No	User Group	Role	Defined by objects and subject's attributes
Capability	Access Right	Access Right	Access Right	Access Right
Accountability (Obligation, Constraint)	No	No	Yes, static and dynamic separation of duty	Defined by objects and subject's attributes
Commitment	No	No	No	No

TABLE 2.
ENGINEERING METHODS AND RESPONSIBILITY'S CONCEPTS.

	KAOS	**I***	**GBRAM**	**ARMF**	**RACAF**	**Scenario Driven**	**Uses Cases**
Subject	Agent	Actors	Agent	Users	Actors	Subject	Actors
Object	Yes	Yes	-	Asset	Data	-	Object
Group	-	Yes	-	Yes	Yes	Yes	Yes
Capability (Right, Authorzation)	Authorization rules	Abilities and beliefs	-	Permission	Permission	Permission	Access right
Accountability (Obligation, Constraint)	Achieve requirements and expectations	Goal	Achieve a goal	Perform a task	Perform a task	Perform a scenario	Pre-conditions, post-conditions
Commitment	No	Yes	No	No	No	No	No

Table 2 is a summary and a comparison of the reviewed engineering methods. We may observe that, because the most frequently addressed concern of Capability is the access right, existing models and methods most of the time remain targeting low-level layers of abstraction of the organization. Moreover, if we consider responsibility as a tuple (Capability, Accountability, Commitment), we observe that nowadays there exists no model and method that entirely take into account all these responsibility components. Other responsibility models exist but are often links to social or psychological areas, or in very specific domains like [41, 42].

IV. FORMALIZATION OF THE RESPONSIBILITY

This chapter aims at defining a responsibility model to clarify and better understand concepts that compose responsibility notion. To achieve that, we firstly use the Unified Modeling Language (UML) to represent the components of the model and their relations and then, we propose to introduce the formalization of its components with formal language and logic system. With the desire to keep this paper didactic and to grant a common understanding of responsibility concepts, the work will be grounded based on the following case study:

Mister Boss is the manager of the marketing company named "SelltheWorld". Each year, Mister Boss organizes during the Christmas period a large sending of postcards to all its customers. This year, Mr Boss has too much work for closing the annual report and consequently decides to delegate this task to one of its employees. Because the task is less business sensitive as some other production task, Mr Boss decides to delegate it to a part-time secretary named Sophie. Sophie has just get married and consequently, she accepts this additional work without Commitment. Mr Boss asks to the IT service manager to give Sophie the necessary access right to the customers address list. The IT service manager asks an employee from the IT service named John to realize the necessary operation for providing this right. On January the 30th, Mister Boss receives over 100 complains of customers that didn't receive Christmas card.

Mr Boss has duly formalized Sophie's Accountability by asking her to realize the sending activity. It was consequently clear about what she was accountable to do. To achieve that sending, she got the necessary Capability that was the access to the customers file. However, due to the fact that her thought went to her new husband rather that to the work to accomplish, she didn't really want to achieve the work and failed to assure her responsibility due to a miss of Commitment.

John's responsibility can also be analyzed by that case study. John is a well paid IT staff that is very happy with his function. He has received clear Accountability to give access right to Sophie and he has the needed capabilities due to its position as network administrator. He has consequently been responsible to fulfill Mr Boss' request.

A.Responsibility model

This section presents our model of responsibility. The major interest of it is its genericity. Indeed, the model aims to be generic enough to be applied to all kind of organizations, at each abstraction layers of it, and also for all domains of the company like for example the IT security (and the management of access right), the management, or the production.

Some components of the model are generic in that they are present at all instantiations of it. Others components have been added with the objective to illustrate the application of the model in the context of the management of access rights. Those components are *Access Right* and *Resource* . Our model reuses some commonly accepted components presented in the literature survey in sections 3 and 4, whereas others are new. The model encompasses the following concept:

- *Organization*: At the top of the UML model (see Fig. 2) is the organization. Organization represents a structure that pursues collective goals and that is limited by a defined border. This structure encompasses employees (users) that are responsible to perform tasks (or processes) that implicitly generate profit. Organization also encompasses resources that could be whether produced by the task or used by a user to perform a task.
- *User*: User appears as a person external or internal to an organization, a system or a software component. User has to achieve a task he is responsible for. Number of synonyms of it exists like subject, actor or agent. For administration facilities, those users are often grouped together based on their profile. As previously explained in the literature overview, the most famous type of classification is the role but variations exist such as for example the team, the hierarchy, or some geographical constraints.

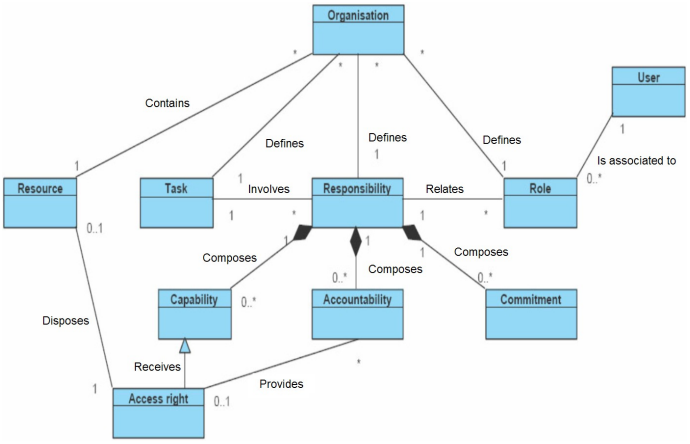

Fig. 2. UML model of responsibility applied to the right management

- *Role*: Role describes the position of a person in the organisation. This position may be related to a hierarchical status, a geographic position, the membership to an organisation unit or department, or whatever. This component is largely present in the literature that provides some definitions of it.
- *Responsibility*: It also exists a plethora of definitions of responsibility and this paper has not for duty to propose a new one. We may however state that commonly accepted responsible definition encompasses the idea of having the obligation to ensure that something happens. Moreover, the above literature review shows that it makes sense to hang on to it the three additional elements that are Capability, Accountability and Commitment. One basic relation existing in the model is consequently the relationship between Responsibility and Capability, Accountability and Commitment. This relation is of the form 0..* to 1. That means that being responsible involves that it is possible to dispose of many Capacities, Accountabilities and Commitment. But at the opposite, on Commitments is only bound to one responsibility, and adequately for Accountability and Capability.
- *Task*: is the operation performed by the role (or the user), which is responsible for it. This concept doesn't exist in the realm of access control model that tends rather to speak about right or/and obligation needed to perform an operation. E.g.: The right to read a document or the obligation to satisfy conditions before executing an operation. By contrast, task is a centric concept in requirement engineering. E.g.: in Tropos, a goal may by achieve by fulfilling a task. The relation between role, responsibility and task is to be underlined. This relation is to be read: "there is one and only one role responsible for one task, and one role may have many responsibilities and one responsible may perform many tasks".
- *Accountability*: is a concept that exists mainly in

engineering methods and that appears through the obligation to achieve a task or to perform an action. This concept describes the state of being answerable about the achievement of a task. The case study above illustrates that Sophie is accountable toward Mr Boss regarding the task she has been assigned responsible for. In the same way, John is accountable toward the IT manager for providing the access right.

- *Commitment*: is the moral engagement of a stakeholder to fulfil a task and the assurance that he will do it. Commitment is the most infrequent concept. Traditional policy model such as RBAC do not address it, however i* partly introduces it (e.g. when defining dependency as an "agreement" between two actors). However, to distinguish if it is a moral concept or an obligation remains interpretable. This component is illustrated through the cases study as follow: Firstly, we may state that because Sophie has other duty in mind, she has not the willingness to achieve the task. We may state that she is not committed to do it. At the opposite, John is a well paid IT staff that is very happy with his function. He is fully committed to perform the task.
- *Capability*: which describes the quality of having the requisite qualities, skills or resources to perform a task. Capability is a component that is part of all models and methods, and is most frequently declined through definition of access rights, authorizations or permissions. Based upon the above case study, the Capability is illustrate through the Sophie's Capability to access the customer's file. This Capability exists because John was responsible to provide that access right. The case study illustrates also John's Capability to be responsible for providing access right. Indeed, due to his position of network administrator, he has the right to manage all employees' access right.

Additionally, the UML model of responsibility (Fig. 2) includes two added elements to the basic responsibility model: "*access right*" and "*resources*". These elements permit to illustrate the case of a particular type of Capability that is the access to resources. We define a resource as something needed for or produced by performing a task and that can takes a large scale of representation such like information, manpower or money. The access right is defined as a statement over the type of action that could be performed by a user over that resource. This access right is a Capability for a responsible while being at the same time Accountability for another. This relationship between Capability, access right and Accountability has been more deeply explained in [2] and [36]. In our model, Capability is a broader concept than the mere one of access right.

The advantages of such a model (Fig. 1) are important for 4 reasons:

1. It permits to improve the business/IT alignment and brings material to answer to the principle 1 of the ISO/IEC 38500:2008 standard [40]: 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 [39]). This makes the agent personally more involved and more concerned by the activity to achieve because he does not shared the result anymore.
3. It addresses the commitment aspect of the responsibility and consequently increases the ethics of the business in general.
4. It guarantees that the right capability is affected to the right agent. This advantage guarantees that the agents receive the minimum privileges necessary for achieving their activities and consequently, it decreases the vulnerability of the system.

B.Selection of a formal system

Even if this model brings up a first contribution for verifying the organization structure and detecting policy problems and inconsistency, it appears impossible to exploit it without the help of a formal language. This section introduces a preliminary reflection over the selection of that language.

To select a language, we may state that the model of responsibility formalizes information that represents responsibility elements in force in the company. That information composes a system that is part of the real world called the universe of discourse and that encompasses a number of properties (constraints) that the system must satisfy. In [38], Meyer at al. explains that some of the constraints may not be violated and could be formalized using predicate logic, temporal logic or dynamic logic whereas others are violable and formalized using deontic logic. The constraint that before to have access to a file, it is necessary that the right for accessing the file has been dully set on the fileserver is inviolable. Indeed, according to our case study, it is impossible that Sophie get access to the customers list if she doesn t have the right to read the concerned file. If we consider that read the file is a proposition, we can deduce that having the access right is a Capability or a modal operator of read the file . Some others constraints are considered as ideal but violable. This could be illustrate by the responsibility of John that as to set the necessary access right for Sophie that but due to an overload of work did not have enough time to achieve it. Time is considered in that example as the Capability necessary to fulfill the task. John has not assumed is responsibility because the statement that John is capable to do it as been violate.

In [33], Cholvy et al. propose a formalization of the concept of responsibility. In her work, she explains that responsibility is a concept that has several facets that correspond to very different meanings. She extracts three definitions of responsibility, which implicitly encompasses the three concepts from our model (Capability, Accountability, Commitment). The first definition links the responsibility concept to something bad that has happened to a person that could have caused or prevented it. This definition is mainly issued from the legal world. The second definition issued from Cholvy's paper claims that *responsibility is an obligation or a moral duty to report or explain the action or someone else's*

action to a given authority (answerability). This definition helps at defining the Commitment as a moral duty in parallel with an obligation that is considered as a legal duty. The third definition defines the responsibility according to a position in an organization and explains that someone responsible for something should be prepared to justify his action. This justification brings the content of the concept of Accountability and consequently nuances Accountability versus answerability. Based upon the three definitions, Cholvy proposes a logic framework and explains how the framework may be used to model different aspects of the responsibility. She used the deontic logic and the logic of actions to achieve that. Deontic logic is the field of logic that is concerned with obligation (O), permission (P) and prohibition (F), and that permits to reason about ideal versus actual states or behavior.

According to her approach, and based upon the Meyer's explanations over the necessity to prefer deontic logic for modeling system that encompasses ideal but violable properties, way may rightly agree that Cholvy's choose is suitably justified. If we consider the model of responsibility as a user based representation of one responsibility, what means in other words, that the concepts of responsibility introduced in the model represents the responsibility of a unique user to perform a unique task, the three components that compose the responsibility tuple are violable. For example, regarding the Capability, we may state that based upon the case study, Sophie must have the list of addresses. However, it may happen that due to undefined reasons, she doesn't have it.

If we expand the sphere of responsibility from a user based perception to an organization based perception, this statement is no more automatically true. Indeed, if the company is considered as a set of tasks, persons, and responsibilities, we may suppose that in an ideal situation, it must exist at least one Capability, one Accountability and one Commitment corresponding to each responsibility.

The existence of Capability and Accountability concepts is easily manageable and verifiable. Indeed, it is easy for an operation or a processes manager to determine the dully capabilities necessary to perform a task or to clearly fix the expected accountabilities. Moreover, such concepts are easily traceable in a database for example or with a software tool. This exercise as already been achieved in previous works [36].

The consistency between both concepts may also be examined based upon the supposition that the Capability needed for assuming a responsibility corresponds to Accountability of another user's responsibility. Fig.3 illustrates that links. Based upon our cases study, we may consider that :

a) *Sophie's Capability (having access right) is the Accountability of John (provides access right).*
b) *John's Capability (having time for performing the right management) is the Accountability of the IT service manager (provide time the IT service staff)*
c) *IT service manager's Capability (having budget to hire IT employees) is the Accountability of Mr Boss (provide IT service manager budget)*

If we base our work on that reasoning that in an ideal situation, responsibilities in a company are dully fixed and that capabilities and Accountability exist for each responsibility, we may conclude that those two concepts are inviolable and may be formalized using predicate logic.

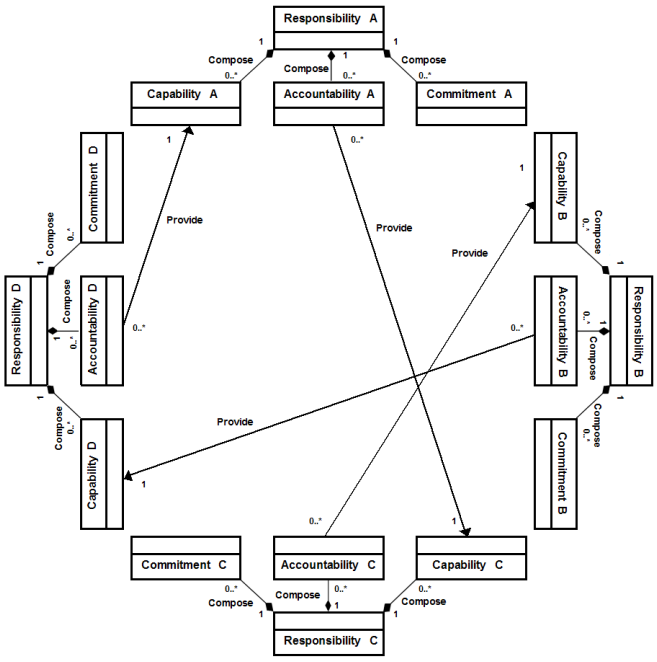

Fig. 3. UML model of multiple responsibilities interactions

While based on our hypotheses that the existence of Capability and Accountability is inviolable, the concept of Commitment is more likely to discussion. Because this concept is strongly depending of the moral willingness, we may argue that no real elements may absolutely guarantee its inviolability. This affirmation may however be nuanced if we look toward social, psychology, or managerial sciences. The salary, the relationship with colleagues, or the concordance of the job with the interest of the employee are some elements that probably influence it. However, in this paper we consider that those elements are not objectively manageable and do not provide a guarantee of inviolability. We will consequently prefer the usage of deontic logic for formalizing that element.

We may consequently suppose that some elements of responsibility may be formalized using predicate logic and others with deontic logic.

V. Future works regarding the formalization of the responsibility

Additionally to the Cholvy s proposition to formalize responsibility with deontic logic and action logic, our future works extend the formalization of the responsibility with the components of the responsibility tuple (Capability, Accountability and Commitment). The responsibility (R) assigned to a user (u) to perform a task (t) is written R([t]u). Based upon our previous observations, we state that this formalization has one specificity that resides in that the components of the responsibility s tuple are at the same time conceptual components and modal operators: Capability (CA), Accountability (AC) and Commitment (CO). We have

consequently to develop a formalization based on the deontic logic to formalize the user based formalization of the responsibility and extend this formalization to predicate logic to represent the responsibility at an organization level. An envisaged possibility to define responsibility s modal operators is to develop the user based representation of the responsibility based on the adaptation of the *Traditional Threefold Classification* (TTC) [37]. To achieve that, we transpose Obligatory to Accountable in that both modal operators bring up the notion of a constraint that is indispensable and makes obligatory by a legal issue (e.g.: a policy), we transpose Permissible by Capable in that both defend the idea that this constraint permits an action to be performed. And we keep the Optional (OP) modal operator of the standard deontic logic unchanged. To achieve that transposition, we need to define the Incapability (IN) and the Unaccountability (UN) (see 2 And 3). Moreover, equally to the deontic standard schema, the Fig. 4 highlights that the three rectangular cells are jointly exhaustive and mutually exclusive. Indeed, each proposition is accountable, optional or incapable. Moreover, Capable modal operators are those that are either Accountable or Optional and Unaccountable modal operators are those that are either Optional or Incapable.

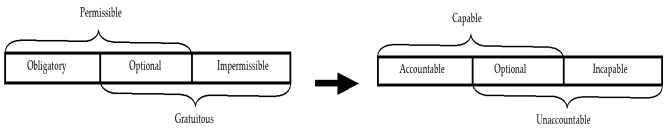

Fig. 4. From Traditional toward a Responsibility based Threefold Classification

Based upon the TTC, the *Traditional Definitional Scheme* (TDS) [37] states by the set of definitions from 1 to 4 that something is permissible if and only if its negation is not obligatory, impermissible if and only if its negation is obligatory, gratuitous if and only if it is not obligatory, and optional if and only if neither it nor its negation is obligatory. If we consider that the proposition (p) is the performance of a task (t) by a user (u) and is noted [t]u, the set of definitions from 1' to 4' may defines the concepts of the responsibility according to the Responsibility based Threefold Classification.

$$PEp \leftrightarrow \sim OB\sim p . \quad (0)$$
$$IMp \leftrightarrow OB\sim p . \quad (2)$$
$$GRp \leftrightarrow \sim OBp . \quad (3)$$
$$OPp \leftrightarrow (\sim OBp \,\&\, \sim OB\sim p) . \quad (4)$$

$$CA[t]u \leftrightarrow \sim AC\sim [t]u . \quad (1\ )$$
$$IN[t]u \leftrightarrow AC\sim [t]u . \quad (2\ )$$
$$UN[t]u \leftrightarrow \sim AC[t]u . \quad (3')$$
$$OP[t]u \leftrightarrow (\sim AC[t]u \,\&\, \sim AC\sim [t]u) . \quad (4')$$

For achieving a task, u must have the necessary capabilities and be committed to perform it. Whether or not he is accountable do not presents any impact on the realization. Whatever, not achieving a task for which the user is accountable may lead to some kind of blame. This aspect is not discussed in that paper.

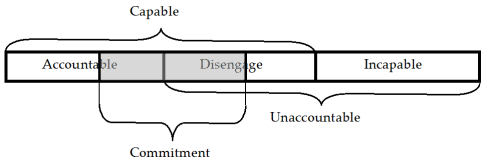

Fig. 5. Commitment on Responsibility Threefold Classification

Future formalization works will also aims at defining the Commitment. We already suppose that it will be necessary to also define it based on the TTC. Fig. 5 shows how it seems logic to represent it.

VI. Conclusions

We have analyzed the literature to understand the semantics of AC policy conceptual models and engineering methods. We have observed that some elements are commonly accepted components whereas others remain debated or not addressed. Commonly accepted concepts are user (and related ones such as group or role), resource and Capability. Capability is most frequently declined under access right, authorizations or permissions. Accountability is a concept that exists mainly in engineering methods and that is declined as the obligation to achieve a task or to perform an action. Commitment is the most infrequent concept. Based upon that observation, we have developed a conceptual model of responsibility using an UML class diagram and have defined all the conceptual components and clarified some important relationships between those. Thereafter, to propose a formal representation of the model, we have selected the suitable language and logic system. The analyze highlights that an important variable is whether the responsibility is perceived at a user or at a company level.

In this paper, the responsibility concept has mainly been addressed based on an IT approach. However, the "operational" and "management" fields are also rich of responsibility's theories [34] and [35]. This area will be the focus of our future researches and will permit to refine our first findings. Consequently, our future works will focus on continuing the development of the model of responsibility, and most specially the concept of Commitment that is important to consider in high-level layer of the organizational model. Moreover, defining policy that allows taking into account the Commitment opens doors to new approaches that have right now poorly be taken into account in traditional and renowned risk management solutions

As a conclusion regarding the Camerer's warning of section II, we have done this analysis to clarify the semantic of all components that encompass the responsibility and we may consequently state that symptom 1 and 3 identified by Camerer has been addressed. Firstly the symptom 1 that is "Concepts are often ambiguous and their definitions are not agreed upon" has been partially tackled with clear literature-based enlightenment of the concepts. Secondly symptom 3 that is "Theories do not 'cumulate' or built upon previous theories as they should." has been addresses with a tentative definition of "responsibility" considering the way its conceptual component are addresses by others authors.

Another part of our work aims at defining a new approach to derive the responsibility from the high-level down to the

lower one. Our first researches demonstrate that potentials solutions are to link responsibility's concepts with organization's processes. To support the progress of that approach, a software prototype has been developed based on "egroupware open framework". Those researches and the prototype have been presented in [36].

References

[1] R. Sandhu, J. Park, Usage Control: A Vision for Next Generation Access Control, The Second International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security, 2003.

[2] C. Feltus, A. Rifaut, An Ontology for Requirements Analysis of Managers' Policies in Financial Institutions, I-ESA2007, Madeira, Portugal.

[3] Gustaf Neumann, Mark Strembeck, A Scenario-driven Role Engineering Process for Functional RBAC Roles, SACMAT'02, June 34, 2002, Monterey, California, USA.

[4] Coyne, E. J. 1996. Role engineering. First ACM Workshop on Role-Based Access Control, Gaithersburg, Maryland, United States.

[5] N. Damianou, N. Dulay, E. Lupu, M. Sloman , The Ponder Policy Specification Language Workshop on Policies for Distributed Systems and Networks (Policy2001), HP Labs Bristol, 29-31. Springer-Verlag.

[6] Bertino, E., Mileo, A., and Provetti, A. 2005. PDL with Preferences. IEEE international Workshop on Policies For Distributed Systems and Networks, Policy 2005 – Vol. 00, IEEE Computer Society, Washington, DC, 213-222.

[7] Basile, C.; Lioy, A.; Perez, G. Martinez; C., F. J. Garcia; Skarmeta, A. F. Gomez, POSITIF: A Policy-Based Security Management SystemPolicies for Distributed Systems and Networks, 2007. POLICY'07, pp. 280 – 280.

[8] Lalana Kagal, Rei : A Policy Language for the Me-Centric Project, TechReport, HP Labs, September 2002.

[9] Colin Camerer, Redirecting Research in Business Policy and Strategy, Strategic Management Journal, Vol.6, No. 1. (Jan. – Mar., 1985), pp. 1-15.

[10] René Wies, Using a Classification of Management Policies for Policy Specification and Policy Transformation. In Proc. ISINM '95, Santa Barbara, California, May 1995.

[11] André Rifaut, Christophe Feltus, Improving Operational Risk Management Systems by Formalizing the Basel II Regulation with Goal Models and the ISO/IEC 15504 Approach, REMO2V'2006, Luxembourg.

[12] Davrondhon Gafurov, Kirsi Helkala, Nils Kalstad Svendsen, Security models for electronic medical record, Telektronikk 1.2005.

[13] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn and Ramaswamy Chandramouli, Proposed NIST Standard for Role-Based Access Control, ACM Transactions on Information and System Security, Vol. 4, No. 3, August 2001, Pages 224-274.

[14] C. Bettini, S. Jajodia, X. S. Wang, and D. Wijesekera, Provisions and Obligations in Policy Management and Security Applications, 28th VLDB conference, China, 2002.

[15] Robert Crook, Darrel Ince, Bashar Nuseibeh, Modelling access policies using roles in requirements engineering, Information and Software Technology 45 (2003) 979-991.

[16] Charles B. Haley, Robin C. Laney, Jonathan D. Moffett, and Bashar Nuseibeh, Using Trust Assumptions with Security Requirements, Requirements Engineering Journal, vol. 11 no. 2 (April 2006) pp. 138-15.

[17] Robert Crook, Darrel Ince, Bashar Nuseibeh, On Modelling access policies: Relating Roles to their Organisational Context, RE 2005, Paris.

[18] Pete A. Epstein, Engineering of Role/Permission Assignement, PhD thesis.

[19] Crook, R., Ince, D., and Nuseibeh, B., "Using i* to Model Access Policies: Relating Roles to their Organisational Context", Social Modelling for Requirements Engineering, Giorgini, P., Maiden, N., Mylopoulos, J., and Yu, E., eds., MIT Press, 2006.

[20] P.J. Fontaine, Goal-Oriented Elaboration of Security Requirements. M.S. Thesis, Dept. Computing Science, University of Louvain, June 2001.

[21] Yu, E. S. and 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 35 194.

[22] L. Liu, E. Yu, J. Mylopoulos, Analyzing Security Requirements as Relationships Among Strategic Actors, SREIS'02, Raleigh, North Carolina, 2002.

[23] Antón, Goal-Based Requirements Analysi,. Second ICRE'96, Colorado Springs, USA, 1996.

[24] Robert Crook, Darrel Ince, Bashar Nuseibeh, Towards an Analytical Role Modelling Framework for Security Requirements, Security Requirements Group, Departement of Computing, The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK.

[25] Henry Mintzberg, Structure in Fives: Designing Effective Organisations, Englewood Cliffs, NJ: Prentice-Hall, 1983. pp. 312

[26] Qingfeng He, Annies I. Antón, "A Framework for Privacy-Enhanced Access Control Analysis in Requirements Engineering", Proc. of the 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03), pp. 137-146, Klagenfurt/Velden, Austria, June 16-17, 2003.

[27] E. B. Fernandez and J. C. Hawkins, "Determining Role Rights from Use Cases", Proc. of the ACM Workshop on Role-Based Access Control, 1997.

[28] Roeckle, H., Schimpf, G., and Weidinger, R. 2000. Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization. In Proceedings of the Fifth ACM Workshop on Role-Based Access Control (Berlin, Germany, July 26 - 28, 2000). Role-Based Access Control '00.

[29] Chandramouli, R. 2001. A Framework for Multiple Authorization Types in a Healthcare Application System. 17th Annual Computer Security Applications Conference, 2001. ACSAC. IEEE Computer Society, Washington, DC, 137.

[30] D. J. Thomsen, Richard C. O'Brien and C. Payne, Napoleon: Network Application Policy Environment, ACM Workshop on Role-Based Access Control, 1999, pp. 145-152.

[31] N. Dulay, E. Lupu, M. Solman, N. Damianou, A Policy Deployment Model for the Ponder Language , An extended version of paper in Proc. IEEE/IFIP International Symposium on Integrated Network Management, (IM'2001), Seattle, May 2001, IEEE Press.

[32] OASIS, "eXtensible Access Control Markup Language (XACM) Version 2.0" February 2005. www.oasis-open.org/committees/xacml/

[33] L. Cholvy, F. Cuppens, and C. Saurel. Towards a logical formalization of responsibility. In Proc. of the Sixth International Conference on Artificial Intelligence and Law, pages 233--242, 1997.

[34] Mintzberg H. Mintzberg on Management: Inside our strange world of organizations. The Free Press, New York, 1989.

[35] Gray, B. Collaborating. Jossey-Baas, San Francisco, 1991.

[36] J. Aubert, B. Gateau, C. Incoul, C. Feltus, SIM : An Innovative Business-Oriented Approach for a Distributed Access Management, International Conference on Information & Communication Technologies: from Theory to Applications (IEEE ICTTA2008), Damascus, Syria.

[37] http://plato.stanford.edu/entries/logic-deontic

[38] J.-J. Ch. Meyer, R.J. Wieringa, F.P.M. Dignum, The Role of Deontic Logic in the Specification of Information Systems, International Series In Engineering And Computer Science archive, Logics for databases and information systems book contents, Kluwer, pp. 71-115, 1998.

[39] C. Feltus, Preliminary Literature Review of Policy Engineering Methods - Toward Responsibility Concept, ICTTA2008, Damascus, Syria.

[40] International Standard for Corporate Governance of IT (IT Governance) - ISO/IEC 38500, 2008

[41] I. Sommerville, T. Storer, and R. Lock. 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, June 2007.

[42] P.M Wright, K. White, D. Gaebler-Spira (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.