Improving Responsibility modelling in Enterprise Architecture, Case Study in the Healthcare Sector

Feltus, Christophe and Petit, Michaël and Dubois, Eric

Luxembourg Institute of Science and Technology, University of Namur

1 March 2014

Improving Responsibility modelling in Enterprise Architecture, Case Study in the Healthcare Sector

Christophe Feltus[1], Michaël Petit[2], Eric Dubois[1]

[1]Luxembourg Institute of Science and Technology,
L-4362 Esch/Alzette, Luxembourg
[2]Computer Science Department, University of Namur,
B-5000 Namur, Belgium
Corresponding author*: christophe.feltus@list.lu*

ABSTRACT

Economy relies on companies evolving in an increasingly highly regulated environment, having their operations strongly formalised and controlled, and being often organised following a bureaucratic approach. In such a context, new and paramount governance requirements advocate for having the responsibility for business processes and tasks formally defined and assigned to the employees. Without efficient formalisation of the responsibility, these companies face the risk to prevent the satisfactorily delivery of business services and that their image is seriously altered and jeopardised. Hence, among the many challenges related to these new governance requirements is the modelling of the concept of responsibility in a unique and expressive model usable in concrete business situations. Unfortunately, in this domain, we have observed that no (meta)model exists and integrates these new needs yet. The second important requirement is to provide the appropriate rights to the employees following their responsibilities to perform specific tasks. Up to date, no solution, model or method addresses the rights provisioning following this perspective. In this context, the paper proposes firstly to define an expressive Responsibility metamodel in UML, named ReMMo, which allows representing the existing responsibilities at the business layer of the enterprise. Afterwards this Responsibility metamodel is integrated with ArchiMate to enhance its usability and benefits from the enterprise architecture formalism. This integration allows strengthening the semantic of the concepts and relations among concepts from the business layer of the enterprise, and more specially the assignment of rights on business objects to the employees.

Keyword: Responsibility metamodel, enterprise architecture, IT governance, ReMMo.

1. Introduction

Nowadays companies exploit information systems broadly deployed and largely interconnected with worldwide networks and solutions. Those systems most often use and deal with information of a crucial and confidential value and are accessed by people being responsible to perform different business tasks and to reach specific objective. Using these sophisticated information systems, companies possess powerful tools to manage all the dimensions of their business, from the management of customers, production activities, stocks, or human resources.

To support the modelling and the management of those companies, enterprise architecture modelling discipline has been designed and allows modelling all the concepts of the enterprise, from the behavioural concepts of task, process, application function, to the structural concepts of role, actor, application component, or to the informational concepts of business object, data object, contract, and so forth. All of these concepts are additionally structured in layer, to know the business, the application and the technical layers. Using these enterprise architecture models is very interesting in order to understand the connections among the concepts and to make decisions knowing the impact of the modification of one concept on others concepts such as for instance, the impact on the application server following the creation of a business service.

In such a complex and evolving environment we have observed that many governance standards and norms [1, 3, 45-48] have acknowledged the above challenges and have highlighted new needs to be fulfilled for the IT and corporate governance. Among these new challenges is the definition and the assignment of well-defined

responsibilities to employees. This assignment is critical for the companies who, without such an alignment, risk not to be able to deliver their services anymore and thus risk to be seriously discredited by their customers, thereby, jeopardising their image. Different aspects of this responsibility concern the definition of the obligations of the employee related to the responsibility and the accountability that those obligations are fulfilled, the assignment of the appropriate capabilities and accesses to the business objects [44, 49] required by the employees to perform those obligations, and the modelling of these responsibilities considering the applicable standard and norms.

Unfortunately, we have observed that the enterprise architecture modelling solutions do not yet fully consider and integrate these needs and obligations. As a result, we acknowledge that for the moment there exist no solution allowing thoroughly representing the responsibility at the business layer and, alike the concepts that allow defining it, such as the accountability, the capability or the right to use. This analysis has led us to the two following research objectives. Firstly, given that the definition and the modelling of this concept of responsibility remains insufficient and incomplete regarding to the many facets of governance, an *expressive* Responsibility metamodel is defined. This Responsibility metamodel is built around the accountabilities of an employee regarding a single business task and around the rights and capabilities required to fulfil these accountabilities. Although, these concepts of business task and right are common in the field of IT, there is no explicit relation between them and the rights and capabilities provided to the employees are not systematically aligned with their accountabilities.

The second objective aims to integrate the Responsibility metamodel with the business layer of ArchiMate [2], in order to enhance its *usability*. The latter is an enterprise architecture metamodel used to give business and IT static views of the corporate architecture as well as the links between these views. That integration allows moreover enhancing the semantic richness of the concepts that compose the enterprise architecture frameworks.

This paper is structured has following. In the next section, we present the main existing researches for representing the responsibility. In section 3, we elaborate the Responsibility metamodel. In section 4, we integrate it with ArchiMate and, in section 5 we evaluate the research using an extended case study from the healthcare sector and Section 6 concludes the paper.

2. Related work

This section aims at providing a view on the perception of the concept of responsibility in the IT domain. To date, this concept has been poorly addressed by the research concerned in the management of IT and authors having published on those topics are limited. Storer and Lock [4] define the responsibility as duties which are to be discharged by agents. Sommerville [5] completes this definition and precise that the duties exist in order to achieve, maintain or avoid some given state, subject to conformance with organisational, social and cultural norms and Stahl [6] introduces the notion of answerability: The responsibility is the ascription of an object to a subject rendering the subject answerable for the object. Martin [7] presents an interesting work to introduce the multi-facet of the responsibility in IT. In this paper, the authors collected a set of problems that could occur around the responsibility of a team of analysts and developers engaged in the development and the deployment of an electronic patient's records system (ERPs) in the National Health Service (NHS) in England. Through some well-defined examples, they illustrated the sharing of responsibilities between the designer and the users of the system. The paper presents, e.g., how a responsibility can be transferred from the system user to the designer when the information, necessary for the design, provided by the user on the processes is not accurate, and so forth.

Martin et al. analyse the work of the designers and their relationships with the users under the ethnographic perspectives and they try to bring solutions regarding the assignment of the responsibilities in the every-day design work. Sommerville [8] also lists six types of responsibility vulnerabilities to introduce his work: (1) unassigned, (2) duplicated, (3) uncommunicated, (4) misassigned,(5) responsibilities overload and (6) responsibility fragility.

Strens and Dobson [9] address the responsibility concept to consider the security of the information system and they advocate that the security must be perceived through a sociotechnical approach rather than only through a technical point of view. Without defining a formal model of responsibility, they explain that the responsibility is built around three types of needs: the need to know, the need to do, and the need to show how the responsibilities are fulfilled. Based on the responsibility, they explain that the designer of a system can better understand what the user needs. For the author, the obligations are linked to the agent that performs activities and by doing these activities; the agent fulfils his/her obligation.

Strens and Dobson [9] define the responsibility in the perspective of a relationship between agents. One of the agents gives the responsibility whereas the other agent receives it. In the frame of a delegation, based on this perspective, when a responsibility is transferred from one agent to another, a new responsibility is created and both agents are assigned to a type of responsibility and to a type of obligation. The agent that transfers the responsibility is called the responsibility principal and the agent that receives it is called the responsibility holder.

Strens' point of view, about the delegation of the responsibility, is that even if the responsibility principal does not perform the task that he has delegated to the responsibility holder, he remains answerable for that task. Although we could agree with the fact that the principal keeps his/her responsibility when the delegation happens between a manager and one of its subordinate, it is disputable when the transfer of responsibility happens between two agents at the same hierarchical layer. For instance, if a nurse accepts to replace one of her colleague during the pause, this colleague is responsible for the patients during this pause and she is answerable for her acts.

Cholvy [10] is interested in formally modelling the concept of responsibility in the field of IT. For the authors, this formalisation is complex due to the different meanings of the responsibility. In the paper, Cholvy et al. review three meanings of the concept and explain how formalising the responsibility is fundamental to improve the behaviours of the systems and the organisations. She depicts the three following definitions:

1. Something bad happened and you could have prevented it,
2. Obligation or moral duty to report or explain your actions or someone else's action to a given authority (answerability),
3. Position, which enables you to make decisions in a given organisation but implies that you must be prepared to justify your actions (accountability).

Those three definitions will be reviewed later in this section for the elaboration of the Responsibility model.

Ian Sommerville in [8] introduces the Responsibility Assignment Models that have for objective to facilitate the distribution of responsibilities in a system. For Sommerville, there is flexibility in the assignment of responsibility that may always be subject to negotiation. This negotiation takes place during the design of the socio-technical system and permits to set up hyphens between automated and manual tasks.

Sommerville defines two types of responsibilities: causal and consequential. Consequential responsibility reflects who gets the blame or credit for the occurrence of some state of affairs (this perception of the responsibility corresponds to the liability responsibility in Vincent [11]) whereas causal responsibility reflects who or what is responsible for making something happen or avoiding some undesirable state (this perception of the responsibility corresponds to the role responsibility in [11]). Sommerville expresses that the Authority is also a significant concept linked to the responsibility and that, once a responsibility is assigned to someone, there should be an authority which decides whether or not the responsibility has been properly discharged. According to Sommerville, in some cases, the responsibility can be shared between multiple agents. Three types of responsibilities have been depicted: the joint responsibility, the derived responsibility, and the delegated responsibility.

In [12], Sommerville proposes a model of the causal responsibility. As introduction, he depicts the advantages of modelling the responsibility without considering the agent that will be assigned to this responsibility. The four advantages are: (i) it focusses on the responsibility itself and on the intention of the organisation, (ii) it permits to analyse the relationship between responsibilities, (iii) it provides a basis for the assignment of responsibilities and (iv) it provides a basis for vulnerability analysis (i.e., do the agents have the requested capabilities, competencies, resources, and so forth).

Sommerville introduces a granularity in the responsibility that he considers to be simple or composite. A composite responsibility is made up of simpler responsibilities that are coherent and mutually dependent. The simple responsibility corresponds to our understanding of the responsibility. Sommerville distinguishes three classes of causal responsibility Doing, Monitoring or Avoiding. Its proposed pattern for responsibility description includes the following concepts: Name, Context, Type, Classification, Pre-conditions, Post-conditions, Normal process, Variations, Exceptions, Advice and Requirements. Moreover, the author considers that one role may be composed of responsibilities. In [31], the authors exploit the definitions of causal and consequential responsibilities and focus on this concept of responsibility for designing information systems. The authors argue

that using the responsibility permits to better analyse the problems that could arise during the design since it permits to depict organisational failure.

In [42], the author refines the requirement engineering process by enhancing the question *What should the system do*? by *What do the stakeholders need and produce*? He advocates that the modelling of the responsibility can contribute to that enhancement and, therefore, proposes an approach based on three levels:

1. The analysis of the documents.
2. The interviews of the stakeholders.
3. The field observations.

The finality of it is the discovery of information requirements in order for the stakeholder to discharge the responsibility and the translation of these information requirements into the system requirements.

The major difference between Sommerville's work and our work is the relation between the concept of accountability and the concept of responsibility. For Sommerville, only the consequential responsibility implies accountabilities and refers toward an authority that is another agent (person or organisation). The causal responsibility does not imply accountable. In our point of view, each agent is accountable, should it be the agent that does the procedure of a task or the one that is accountable for the achievement of its goal. To represent the responsibility, a graphical model has been introduced by Blyth et al. in ORDIT [13]. The ORDIT methodology aims at representing human resources and the technological system to achieve the organisational goals. Sommerville has proposed a complementary graphical notation that he illustrates in [5, 42].

3. The Responsibility metamodel

In this section, we analyse in details what the concept of responsibility means and we defined ReMMo for modelling a rich concept of responsibility, the accountabilities that are part of it and its links with the employees, the business roles, the tasks, and the rights and capabilities. Fig. 1 represents the main concepts of the Responsibility metamodel. In practice, we note that tasks related concepts (represented in yellow) are assigned to a business role, or sometimes directly to employees, through the accountabilities that are part of their responsibilities (all four in green). Although different actors (roles or employees) may be assigned responsibilities concerning the same tasks, these responsibilities have different meanings most of the time. For instance, a manager may be responsible for achieving the goal of a task although he is responsible for performing the procedure to achieve that task. The accountability may also exist and be impacted respectively by conditions (in green) and governance rules (in grey). Finally, rights and capabilities (both in orange) are required in order to perform these responsibilities and accountabilities.

Concretely, the elaboration of the Responsibility metamodel has been conducted in the field of design research method proposed by [24], named Action Design Research, which considers that the practitioners and end-users possess a rich knowledge regarding the research domain and that it is necessary to have them involved all along the artefact building activity. This method has consequently for objective to strengthen the connections between these practitioners and the researchers by combining the building, intervention and evaluation activities. Accordingly, it advocates for a continual evaluation of the problem and the built artefacts (ReMMo) in order to ceaseless adjust its elaboration with real usage settings. Therefore, a first version of ReMMo has been created by the researcher and alpha versions have been iteratively generated in a limited organisational context. This evaluation by the practitioner was performed at the European Court of Auditors [15]. Afterwards, in a second step, the more mature artefact is evaluated in a wider organisational setting and beta versions are shaped with the end-users. This second iteration of ReMMo has been performed at the Centre Hospitalier de Luxembourg [16].

3.1 Task and business object

Paterno [17] explains that, in the field of software engineering, tasks are activities that have to be performed to reach a goal and he describes a goal as either a desired modification of the state of an application or an attempt to retrieve some information from an application. For Paterno, tasks can also be divided into sub-tasks of lower complexity and the relationship between the tasks can be modelled in various ways. [36] argues that tasks represent solutions to the realisation of goals or soft goals. In order to be achieved or completed, soft goals, goals, and tasks may require resources to be available and, in BPMN [18], a task represents a single unit of work that is

not or cannot be broken down to a further level of business process detail without diagramming the steps in a procedure.

A business object is either a piece of information, a document, or a material object [19] which is, for Bruno and Torchiano [50], produced by a business process. It enables better and more consistent binding of a real-world concept, representing a product or service which is the goal of a business activity, with the actual business process for its realisation [21].

In ReMMo, we have considered the Task a behavioural element and we have modelled it as a Task class. As explained in i*[22], actors depend on each other to achieve a goal or to perform a task. In order to be compliant with these dependencies, while keeping the task as the unique concept concerned by the responsibility, we consider that both types of i* dependencies are Task types. To model this, we consider two types of attributes for the Task: the Goal and the Procedure and we express that one Goal always exists to define a Task although one Procedure may or may not exist. The business object is an object representing some concepts relevant to the organisation which is used by the Task. Accordingly, we define the Task and the Business Object by:

> **DEFINITION 1**: The task is a complete and identifiable piece of work necessary to achieve a goal and that may or may not be defined through a procedure. The task may be either a business task if it aims at achieving a business goal or a structural which if it aims at achieving a structural goal.
>
> **DEFINITION 2**: The business object is a passive element (information or document) which has relevance from a business perspective and which may be used by one or many task(s).

3.2 Responsibility and Accountability, Actor, Sanction and Condition

Globally, most of the authors reviewed in Section 2acknowledge that defining the responsibility aims at conferring one or more obligation(s) to an actor (the responsibility owner) [4, 8, 9]. As a consequence, that obligation provokes a moral or formal duty, in the mind of the responsibility owner, to justify the performance of the obligation to someone else [6, 8-10, 23, 43]. Vincent [11] proposes a structured taxonomy of the responsibility (STRC) and introduces six different perceptions of it according to different science disciplines and points out the role responsibility. The latter means that to improve the management of the responsibility assignment to employee, a set of responsibilities may be assigned to a business role. This statement is also supported by [8].

Beside the literature related to the responsibility, the review of the literature related to the accountability highlights the following: Spinello [24], Mulgan [54], Laudon and Laudon [26] and Stahl [6] express that the responsibility is associated to accountabilities regarding a business task. An accountability is broadly defined as: the obligation to give account to someone else [27-30, 40] under the threat of sanction(s). Ackerman [53] explains that the accountability is a process of justifying [...], the behaviour and the results and to sanction accordingly. Bovens [32] and Mulgan [25] explained that the sanction may be positive or negative. As the concept of accountably appears to be narrowly associated to the concept of obligation regarding a task [40, 27-30], we have also analysed the meaning of this obligation and we acknowledge that it represents what must be done to achieve an expected result [31]. Bovens [32] introduces the sanction in his definition of accountability acknowledging that an actor may face consequences resulting to the appreciation of the achievement of its accountabilities. Mulgan [25] also considers that the sanction is a component of the accountability although Fox [41] considers the hard accountability when there exists sanction(s) and soft accountability when there is not sanction. In that case, an accountability is equivalent to an answerability as introduced by Stahl [6]. For Dubnick [34], the sanction [...] can act as a background reminder for the actor about its moral engagement. Bovens [32] and Mulgan [25] consider that the sanction may be positive or negative: Positive sanctions are for instance a reward, recognition, the receipt of an amount of money although the negative sanction can be a disciplinary measure, a civil remedy. Sommerville [8] also used the twofold of sanctions for its works.

Globally, most of the authors acknowledge that defining the responsibility aims at conferring one or more obligation(s) to an actor (the responsibility owner). As a consequence, that obligation provokes a moral or formal duty, in the mind of this responsibility owner, to justify the performance of the obligation to someone else. Beside the literature related to the responsibility, the review of the literature related to the accountability highlights that the responsibility is associated to accountabilities regarding a business task. Accountability is broadly defined as

the obligation to give account to someone else under the threat of sanction(s) and is part of the responsibility. Accordingly, we propose the following definitions:

DEFINITION 3: The responsibility is a charge assigned to a unique actor to signify its accountabilities concerning a unique business task.

DEFINITION 4: The accountability is an element which is part of a unique responsibility and which represents an obligation of an actor to achieve the goal, or to perform the procedure of a task, and the justification that it is done to someone else, under threat of sanction.

The responsibility is defined for a unique actor to which it is assigned. The concept of actor has already been largely defined in the literature [36] and it will not be reviewed in detail in this work. This concept of actor has been defined in i* as an active entity which carries out actions to achieve goals by exercising its know-how. This actor may be either an employee or a business role.

DEFINITION 5: The Actor is an active entity which is assigned a set of responsibilities and that may check accountabilities.

DEFINITION 6: The BusinessRole is a type of actor which represents a set of employees who share common characteristics.

DEFINITION 7: The Employee is a type of actor which represents a human entity which may or may not play one or more business roles.

Additionally, we have introduced the concepts of sanction and condition which we define as following:

DEFINITION 8: The Sanction is an element associated with an accountability and which corresponds to the consequence resulting from the justification of the realisation (or not) of this accountability.

DEFINITION 9: The Condition defines a context which must be verified for the accountability to exist.

3.3 Capability and Right

To realise his accountability, an actor must possess a set of capabilities and rights to use. The capabilities are abilities that exist intrinsically to the actor and correspond to the knowledge, the know-how, or the attitude he possesses. The capabilities have been analysed by Vernadat [55]. Among the most used capabilities, according to our observation of real situations, we retrieve, eg.: the education, the experience and the knowledge about the enterprise, the authority, the ability to perform a business task, the ability to use software, or even physical characteristics. The concept of right is common but is not systematically embedded in all IT frameworks, Vernadat [55] and IT Governance Institute [2]. It encompasses facilities required by an employee to fulfil his accountability(ies). These facilities are described in terms of access to a business object and may, for instance, represent a right to access information. This access may be to read, to create information or to modify information. They may also correspond to a right to access other business object such as a company car, a meeting room, the support of a team and so forth.

The concept of right is common but is not systematically embedded in all IT frameworks. It encompasses facilities required by an actor to fulfil his accountability(ies). These rights to use are described in terms of access to a business object.

Capability and rights are components that have already been defined in the field of IT [55]. They have been introduced in ReMMo as well. These concepts are defined as following:

DEFINITION 10: The Capability represents the qualities, the skills or the resources intrinsic to the actor and which are required to perform one or several accountability(ies).

DEFINITION 11: The RightToUse represents an authorisation to perform an operation on a business object which is required to perform one or several accountability(ies).

3.4 Governance Rules and Source

In practice, we have observed that the governance rules originated from governance source. These governance sources provide high level rules that impact the elaboration of the responsibilities by expressing conditions over the accountabilities. Both the governance rule and the source are defined as following:

DEFINITION 12: The GovernanceRule is a high level prescript originating from dedicated sources and which constraints the definition of the accountabilities.

DEFINITION 13: The Source is a formal piece of information which creates responsibilities and which contains, amongst other, required or desired governance rules.

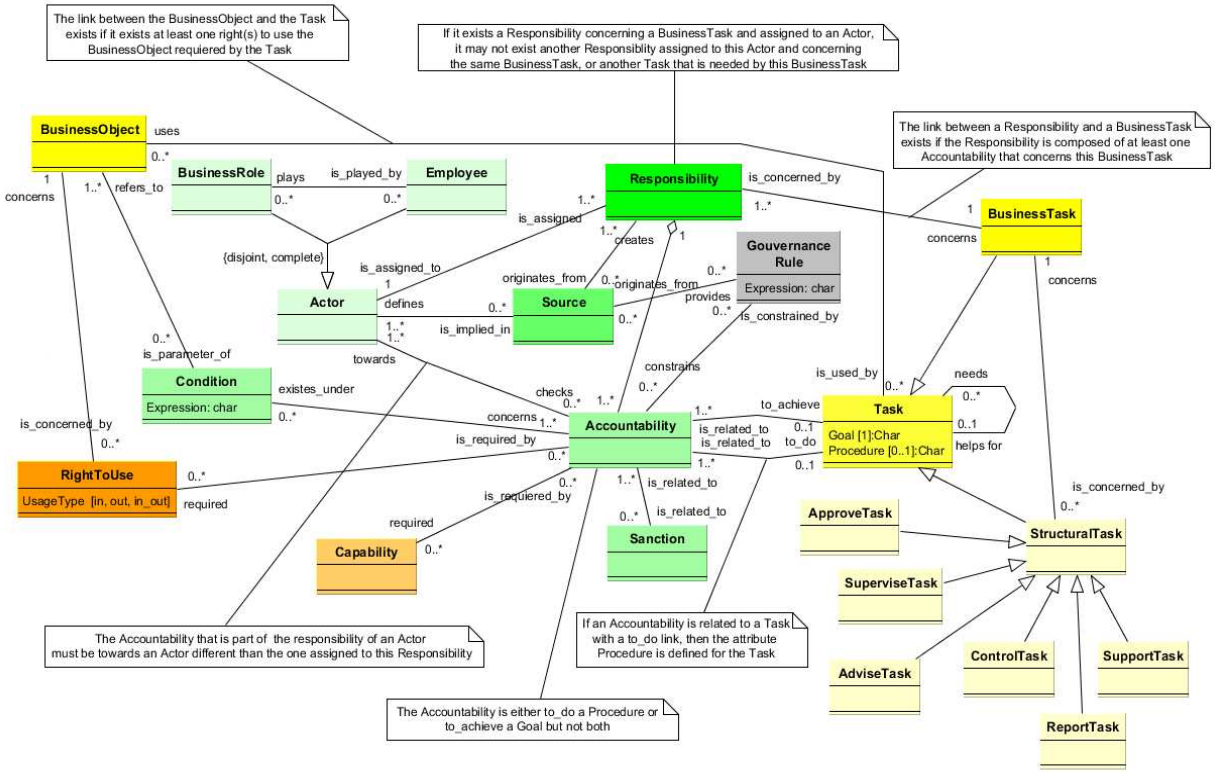

Figure 1: The Responsibility MetaModel, ReMMo.

4. ArchiMate extension with ReMMo

In previous section we have defined an expressive Responsibility metamodel aiming at supporting the modelling and the formalisation of the responsibilities of the employees. This Responsibility has been represented by means of an UML metamodel that allows accurately tracing the connection between i.e. the responsibilities and the accountabilities that compose it, the rights and capabilities necessary to achieve the latter, the condition of existence of accountabilities and the governance rules that define it. By the same time, we have been obliged to admit that the metamodel is UML is far to be use friendly and thereby, difficult to read and to exploit in practice. As a reason, we have decided to integrate this Responsibility metamodel with ArchiMate in order to benefit from the modelling language it provides. The ArchiMate language for modelling enterprise architecture has been preferred provided that it is a free and open standard sustain by The Open Group.

4.1 Method to extend ArchiMate

According to Parent and Spaccapietra [38], the integration of two metamodels requires resolving three types of heterogeneities: syntactic, semantic and structural. For our integration, only the semantic and the structural heterogeneities have been addressed. Indeed, the syntactic heterogeneity aims at analysing the difference between the serialisation of metamodel and, as explained by Zivkovic [39], addresses technical heterogeneity like hardware platforms and operating systems, or access methods, or it addresses the interface heterogeneity like the one which exists if different components are accessible through different access languages. The structural heterogeneity exists when the same metamodel concepts are modelled differently by each metamodel primitives. This structural heterogeneity has been addressed together with the analysis of the conceptual mapping and the definition of the integration rules. Finally, the semantic heterogeneity represents differences in the meaning of the considered metamodel' elements and must be addressed through elements mapping and integration rules. Regarding the mappings, three situations are possible: no mapping, a mapping of a type 1:1, and a mapping of a type n:m (n

concepts from one metamodel are mapped with m concepts from the other). Practically, no case of n:m mapping has been encountered during the mapping between ReMMo and ArchiMate.
After having defined the mapping, the concepts have been integrated in an integrated metamodel using both ArchiMate' extensions mechanisms: the addition of attribute and the specialisation [2]. Concretely, if no mapping was detected, the concept from ReMMo was integrated in the ArchiMate using the first extension mechanism which consists in adding attribute to an existing concept. If a mapping 1:1 exists without conflict between two concepts, both concepts are merged in a unique one, this concept is integrated in the integrated metamodel, and this concept kept the name of the ArchiMate concept. If a mapping of a type 1:1 with conflict exists between two concepts, this means that one concept from one metamodel is richer or poorer than a concept from the other metamodel and in this case, both concepts are integrated in the integrated metamodel using the second extension mechanism of ArchiMate which is the stereotype.

4.2 ArchiMate Responsibility extensions

Concretely, in order to perform the mapping, it has been necessary to remodel the concepts and the associations between concepts from ArchiMate in UML. ReMMo has been reworked has well in order to illustrate the associations classes and thereby model the mappings with the classes and relation classes from ArchiMate. Fig. 2 provides a view on both of those metamodels modelled in UML and of the mapping realised among them. The mapping realised have been summarised in Table 1. Two of them are illustrated in the following. As a first example of mapping, we have observed that the definition of the business role form ArchiMate which is "the responsibility for performing specific behaviour, to which an actor can be assigned" [2] and the definition of the responsibility from ReMMo are semantically close but that the definition of the responsibility is more precise as the one from the business role. Therefore, we have consider a 1:1 mapping with conflict between both concepts that have been added in the integrated metamodel and associated using a specialisation link such as the Responsibility is a stereotype of the business role written «Responsibility». A second example concerns the analysis of the definition of the concepts of business object from the ReMMo and from ArchiMate. Both definitions were semantically equivalent and both concepts have been merged in a unique one name Business Object.

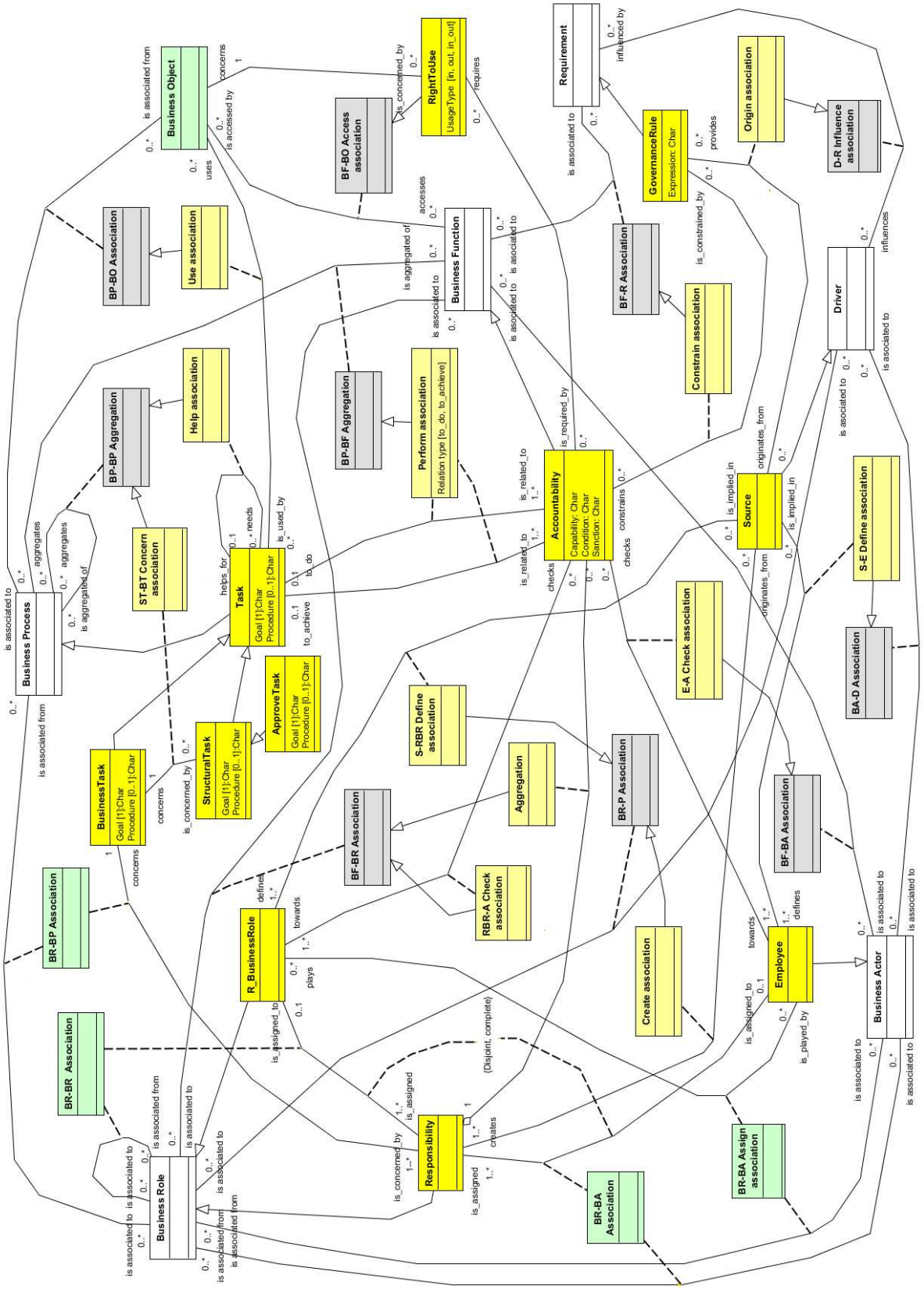

Figure 2. Integration of ReMMo and ArchiMate

Table 1: Concepts mapping and integration rules between ReMMo and ArchiMate

Responsibility element	**ArchiMate element**	**Mapping**	**Integration rule**	**Integrated element**
BusinessObject concept	Business object concept	1:1	Merge	Business Object
Task concept	Business process concept	1:1	Business process specialisation	«Task»
Task Help Task	Business process aggregation of Business process	1:1	BP-BP aggregation specialisation	«Help»
StructuralTask Concern BusinessTask	Business process aggregation of Business process	1:1	BP-BP aggregation specialisation	«Concern»
Task Use Business object	Business process association to Business object	1:1	BP-BO association specialisation	«Use»
R BusinessRole concept	Business role concept	1:1	ArchiMate Business role specialisation	«R_BusinessRole»
Responsibility concept	Business role concept	1:1	Business role specialisation	«Responsibility»
Employee concept	Business actor concept	1:1	Business actor specialisation	«Employee»
Accountability concept	Business function concept	1:1	Business function specialisation	«Accountability»
Responsibility Aggregation of Accountability	Business role association to Business function	1:1	Association specialisation	«Aggregation»
Responsibility Concern BusinessTask	Business role association to Business process	1:1	Merge	BR-BP Association
Employee Play R_BusinessRole	Business actor assignment to Business role	1:1	Merge	BR-BA Assign association
Employee Check Accountability	Business actor association to business function	1:1	BF-BA association specialisation	«E-A Check»
R_BusinessRole Check Accountability	Business role association to business function	1:1	BF-BR association specialisation	«RBR-A Check»
Task Perform by Accountability	Business process aggregation of Business function	1:1	BP-BF aggregation specialisation and addition of attribute	«Perform», Relation type: to_do or to_achieve
Employee Assign to Responsibility	Business actor association to Business role	1:1	Merge	BR-BA Association
R BusinessRole Assign to Responsibility	Business role association to Business role	1:1	Merge	BR-BR Association
RightToUse concept	access association	1:1	Access specialisation	«RightToUse»
Sanction concept	-	-	Addition of attribute	«Accountability», Sanction: Sanction description
Condition concept	-	-	Addition of attribute	«Accountability», Condition: Condition description
Capability concept	-	-	Addition of attribute	«Accountability», Capability: Capability description
Source concept	Driver concept	1:1	Driver specialisation	«Source»

GovernanceRule concept	Requirement concept	1:1	Requirement specialisation	«Governance Rule»
GovernanceRule Origin Source	Driver influence Requirement	1:1	D-R influence specialisation	«Origin»
GovernanceRule Constrain Accountability	Requirement association to Business function	1:1	BF-R association specialisation	«Constrain»
Source Define R_BusinessRole	Driver association to Business role	1:1	BR-D association specialisation	«S-BRB Define»
Source Define Employee	Driver association to Business actor	1:1	BA-D association specialisation	«S-E Define»
Source Create Responsibility	Driver association to Business function	1:1	BR-D association specialisation	«Create»

5. Case study in healthcare sector

To elaborate the ReMMo, we analysed how the concepts that compose the responsibility are presented in the scientific literature. Practically, for each concept, we reviewed and introduced its origin, we provided our own definition for it, and we integrated it in the Responsibility metamodel. In this section, we provide a short case study in order to illustrate from an example firstly how the responsibilities of employees from the healthcare domain may be modelled using ReMMo and how ReMMo is *expressive* enough to model all the aspect of the case study, this means how the concepts of ReMMo may be instantiated using information the healthcare domain (Fig 3), and secondly how ArchiMate extend with the Responsibility metamodel may improve the *usability* of the models (Fig. 4). The case that we have modelled is the following:

> *In the healthcare domain, the goal is to ensure patient care. To achieve this goal, it is necessary to hire employees who mainly treat patients. In a hospital, hiring employees is a goal under the responsibility of the CEO but taken in charge by the doctor general. The treatment of the patients is a task achieved by analysing the patient's pathology and giving him drugs. To analyse the pathology, it is necessary to seek information in a knowledge base and to make X-Ray analyses. During the treatment, a report about the pathology must be prepared and the team that provides care must be supervised by one doctor. Finally, to be able to seek information in the knowledge base, employees must be instructed on how to seek this information and must therefore receive the appropriate training.*

5.1 Responsibility modelling with ReMMo

In this healthcare case study, as modelled in Fig. 3 the `GuaranteeHealthCare` is BusinessTask that corresponds to a Goal in the healthcare domain. This BusinessTask has only a Goal attribute (but no Procedure attribute) which is to "`have all the patients treated`". That means that it may be achieved by different ways. This is also the case of the BusinessTask `TreatPatient` that helps to achieve the BusinessTask `GuaranteeHealthcare` and of the BusinessTask `DoX-RayAnalysis` that helps to achieve the BusinessTask `AnalysePatology`. All other BusinessTasks include a Procedure attribute that indicates how to achieve the Goal. This is the case of the `HireEmployee`, `ProvideDrugs`, `AnalysePathology` or `SeekInformationInPathologyKnoweledgeBase`. One isolated BusinessTask, `Upgrade-PatologyKnowledgeBase`, is not attached to the graph. This BusinessTask has, i.e., a Goal attribute which is "`knowledge base always up to date`" and a Procedure attribute which is "`verify and install last version`".

The BusinessTask `SeekInformationInPathologyKnowledgeBase` uses the `Pathology-KnowledgeBase` BusinessObject. This means that the employee who seeks information about the pathology in the knowledge base has to use this pathology knowledge base to do research on the existing pathology and to retrieve the pathology that corresponds to the patient symptoms.

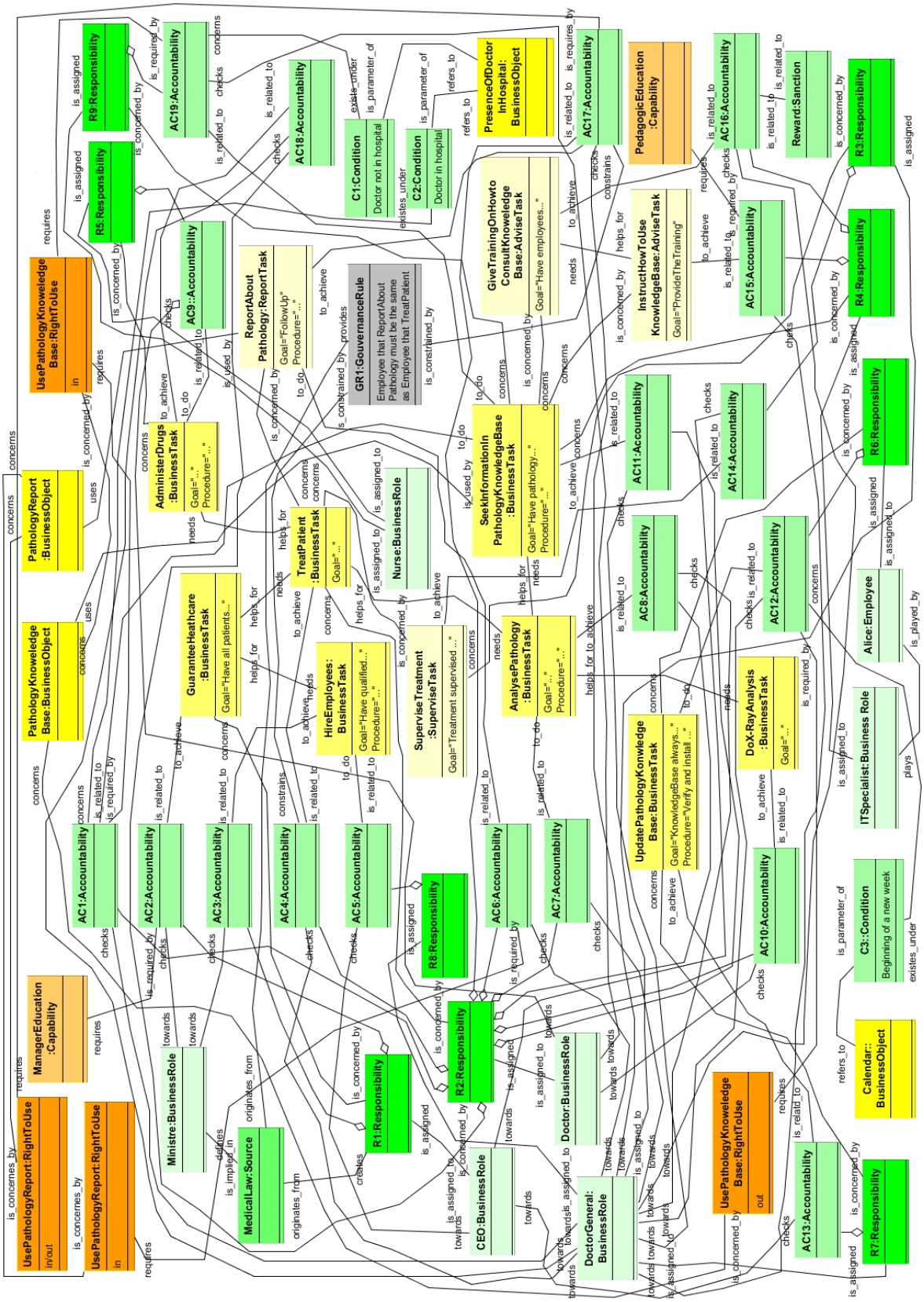

Figure 3: Responsibility model for the healthcare domain.

TreatPatient is concerned by the StructuralTasks `ReportAboutPathology` which is of type "`Report`" and to `SuperviseTreatment` which is of type "`Supervise`". `ReportAboutPathology` is a task which is defined by a goal to have a `PathologyReport` and with a procedure that expresses the different steps of the

reporting procedure. SuperviseTreatment is a task with a goal to supervise the treatment but the way to supervise is not defined. SeekInformationInPathologyKnowledgeBase is concerned by the Structural Task GiveTrainingOnHowToConsultKnowledgeBase which is of type "Advise" and that needs, to be achieved, the achievement of the Structural Task InstructhowToUseKnowledgeBase. The latter also concerns the ConsultPatologyKnowledgeBase Business Task.

One responsibility that we address, in the example of Fig. 3, are the ones of the Doctor which are concerned by the task SeekInformationInThePathologyKnowledgeBase. This Task is described by a Goal that is related to the Accountability AC11. This AC11 is part of the Responsibility R2. The Task is also described by a Procedure that is related to two Accountabilities which exist under the Condition C2 Doctor in Hospital or the Condition C1 Doctor not in Hospital, and which both refer to the BusinessObject PresenceOfDoctorInHospital. The AC1 is part of R2 assigned to the Doctor and exists if Condition C2 is true. The AC19 is part of R19 assigned to the Nurse and exists if Condition C1 is true. One difference between both accountabilities is that AC1 is towards the DoctorGeneral although AC19 is towards the Doctor. In this second case, the Doctor is no longer accountable to do SeekInformationInThePathologyKnowledgeBase but he remains accountable to achieve SeekInformationInThePathologyKnowledgeBase (AC1) towards the DoctorGeneral. To upgrade the PathologyKnowledgeBase is a very critical task and, to avoid too many employees to do it, a dedicated and well identified employee has therefore been designated: Alice.

Fig. 3 also represents the modelling of a delegation of the task TreatPatient from the Doctor to his Assistant. When a Doctor delegates this task, the delegation is recorded in the BusinessObject DelegationState. In the function of this BusinessObject, if the condition DelegationCondition1 The doctor has delegated the treatment of the patient to his assistants is true, AC2 and AC3 exist. Inversely, if the condition DelegationCondition2 The doctor has not delegated the treatment of the patient to his assistants is true, AC1 exists.

An example of RightToUse is the right to use the PathologyKnowledgeBase to seek information about the patient pathology or to update it. We therefore define two types of rights to use the pathology knowledge base. This first is the right of the type In for the Accountability AC1 and AC19. This means that the PathologyKnowledgeBase isused as an input for the Accountability. The second is the right of the type Out for the Accountability AC12 that means that the accountability requires the right to update the PathologyKnowledgeBase. The Accountability AC2 to achieve the Goal of Guaranteein-Healthcare requires the Capability which is to have a ManagerEducation.

The governance rule illustrated is dictated by the MedicalLawSource and imposes that the Actor who ReportsAboutPathology must be the same as the Actor that performs TreatPatient. This GovernanceRule constrains AC4 and AC5 to be part of the same Responsibility R2, which is assigned, in our case to the Doctor.

5.2 Responsibility modelling with ArchiMate extended with the responsibility

Fig 4 presents the Responsibility metamodel, instantiated according to the healthcare domain case study presented in Fig. 4, modelled following the ArchiMate metamodel integrated with the Responsibility metamodel. To keep the figure straightforward, it does not include the Sanction neither the relations "needs/helps for" and "concerns/is concerned by" between tasks, "is related to/is related to" between the sanction and the accountability and "checks/towards" between the business role and the accountability.

Overall, the Fig. 4 is more easily readable and comprehensible than Fig. 3, and is conforming to the ArchiMate formalism. The left side of the figure shows the source, the governance rule, the business roles and the employees. The middle of the figure shows the tasks and accountabilities. The right side of the figure shows the capabilities and the business objects.

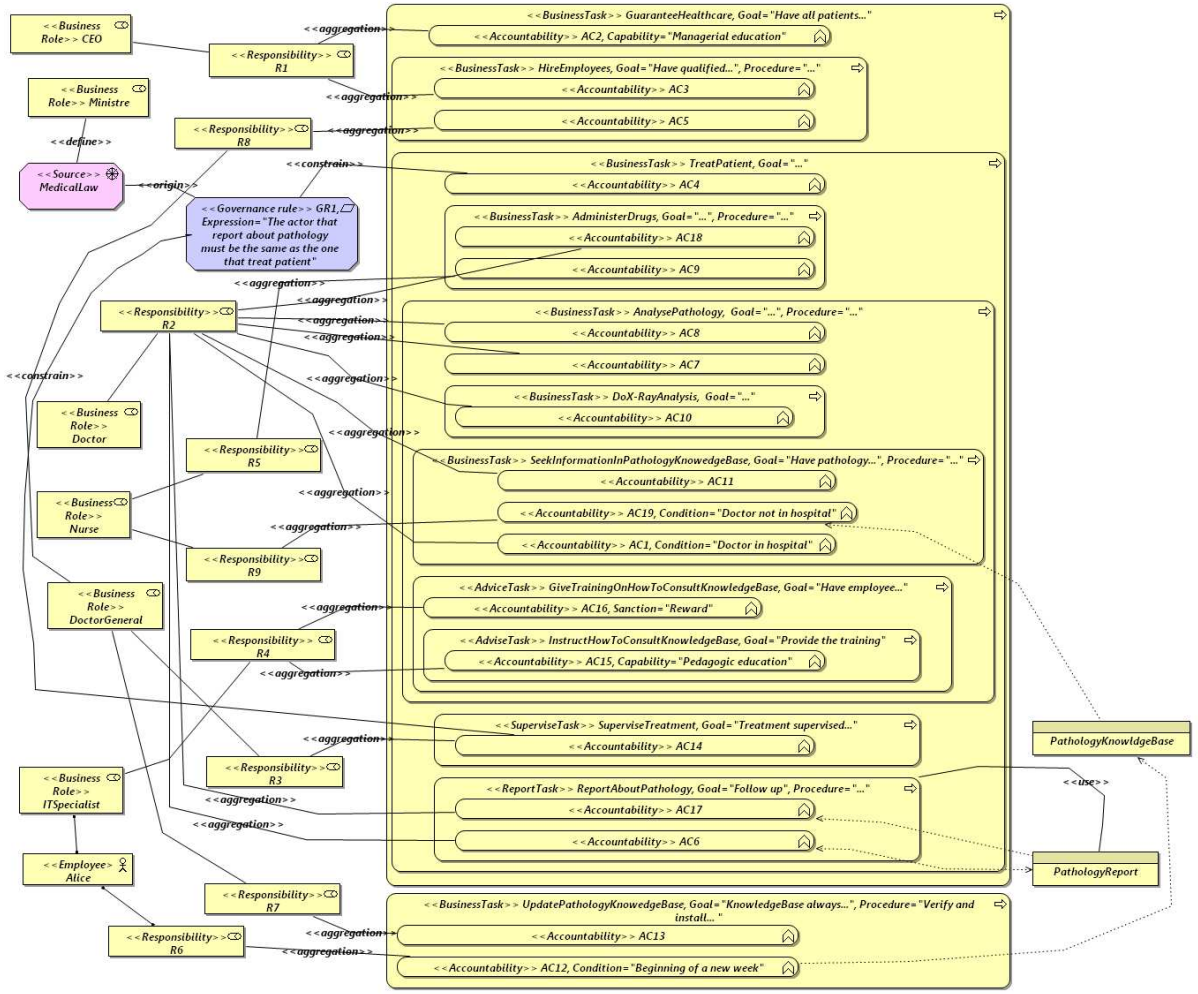

Figure 4: Responsibility model for the healthcare domain.

6. Conclusions

This research has been introduced by the statement that, nowadays the economy relies on companies operating sometimes with an information system shared by thousands of employees, continuously evolving, and gaining more and more flexibility and openness. Using these sophisticated information systems, companies possess powerful tools to manage all the dimensions of their business, from the management of customers, production activities, stocks, or human resources. In such a complex and evolving environment, aligning the business down to the appropriate IT infrastructure is a challenging activity that needs to be carefully handled. The alignment of the application layer with the business layer [37] is critical for the companies who, without such an alignment, risk not to be able to deliver their services anymore and thus risk to be seriously discredited by their customers, thereby, jeopardising their image. One aspect of this business/IT alignment concerns the alignment of the access [20, 28, 51] to business objects required by the employees according to the information they need to perform business activities. Many governance standards and norms have acknowledged the above alignment challenges and have highlighted new needs to be fulfilled in terms of business/IT alignment. Among them are the needs to have the responsibilities of the employees accurately defined and the access to the business objects provided according to these responsibilities. Our review of the related works related to the need for considering the concept of responsibility, as well as a set of concepts that allow defining it, such as the accountability, the capability or the right to use [52, 35], has shown that yet no existing model allows representing this responsibility, and the concepts composing it. Hence, this observation has led us to the analysis of the literature from the field of IS/IT and from the field of the human sciences. The literature analysed has allowed us to discover the amplitude of the notion of responsibility that gathers, at the same time, information related to (1) the accountabilities of the employees, which are mainly defined at the business layer. (2) the rights and capabilities that these employees

require to perform their accountabilities. These rights and capabilities are issued from the business layer but impact the application layer, with e.g. the definition of the access rights. And (3) the assignment of responsibilities to employees, directly or through the roles they play. Knowing the meaning and acknowledging the importance of these concepts led us to the definition of the Responsibility metamodel.

Enterprise architecture modelling is based on approaches which enable illustrating the inter relations between the different layers of a company and between the different aspects that it addresses such as the behaviour, the information, or the people. Enterprise architecture metamodels provide views which are understandable by all the stakeholders and which allow making decisions knowing the impact over the company. However, the problem with the enterprise architecture metamodels is that, in general, the concepts which compose them lack precision and, therefore, are hardly usable to perform, verify or justify concrete alignments. In parallel, we also considered that the enterprise architecture metamodels provide a good basis for this since they model the most significant concepts related to the information system of a company. To reap the benefits of the enterprise architecture metamodel for the engineering and the management of the access to business objects, we have decided to focus our research on integrating ReMMo with ArchiMate, as an example of enterprise architecture modelling language. After this integration, the associations between the business concepts of business actor, business role, business process, business function and business object have been semantically enriched.

References

[1] ISO38500. 2008. ISO/IEC 38500:2008, International Standard for Corporate Governance of IT.

[2] The Open Group. ArchiMate® 2.0 Specification. Van Haren Publishing, The Netherlands, 2012.

[3] IT Governance Institute, editor. CobiT 4.1: Framework, Control Objectives, Management Guidelines, Maturity Models. IT Governance Institute, Rolling Meadows, 2007.

[4] Storer T, Lock R (2008) Modelling responsibility. project working paper 7, indeed project.

[5] Sommerville I, Storer T, Lock R (2009). Responsibility modelling for civil emergency planning. URL http://www.ncjrs.gov/App/Publications/abstract.aspx? ID=250623.

[6] Carsten Stahl B (2006) Accountability and reective responsibility in information systems. 195:51-68. URL http://dx.doi.org/10.1007/0-387-31168-8_4.

[7] Martin D, Rouncefield M, O'Neill J, Hartswood M, Randall D (2005) Timing in the art of integration: 'that's how the bastille got stormed'. In Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work, GROUP '05, pages 313-322, New York, NY, USA.

[8] Sommerville I (2007) Models for responsibility assignment. In Responsibility and Dependable Systems. Springer.

[9] Strens R, Dobson J (1993). How responsibility modelling leads to security requirements. In Proceedings on the 1992-1993 workshop on New security paradigms, NSPW '92-93, pages 143-149, New York, NY, USA.

[10] Cholvy L, Cuppens F, Saurel C (1997) Towards a logical formalization of responsibility. In ICAIL '97: Proceedings of the 6th international conference on Artificial intelligence and law, pages 233-242, New York, NY, USA.

[11] Vincent N A (2011) A structured taxonomy of responsibility concepts. Moral Responsibility: Beyond free will and determinism, Nicole A. Vincent, Van de Poel and Van den Hoven, pages 15-35.

[12] Sommerville I (2007). Causal responsibility model. In Responsibility and Dependable Systems. Springer.

[13] Blyth J C, Chudge J, Dobson J E, and Strens M R (1993) Ordit: a new methodology to assist in the process of eliciting and modelling organizational requirements. In Proceedings of the conference on Organizational computing systems, COCS '93, pages 216-227, New York, NY, USA.

[14] Sein M K, Henfridsson O, Purao S, Rossi M, Lindgren R (2011) Action design research. MIS Q., 35(1):37-56, ISSN 0276-7783.

[15] Petit M, Feltus C, Vernadat F (2012) Enterprise Architecture Enhanced with Responsibility to Manage Access Rights - Case Study in an EU Institution, in Proceedings of The Practice of Enterprise Modeling - 5th IFIP WG 8.1 Working Conference (PoEM), Rostock, Germany.

[16] Feltus C, Dubois E, Proper E, Band, Petit M (2012) Enhancing the ArchiMate® Standard with a Responsibility Modeling Language for Access Rights Management, in Proceedings of the 5th International Conference on Security of Information and Networks (SIN), Jaipur, Rajastan, India.
[17] Paterno F (2001) Task models in interactive software systems. In S. K. Chang, editor, Handbook of Software Engineering and Knowledge. World Scientific Publishing Co.
[18] White S A (2004) Business process modeling notation (bpmn) version 1.0. Technical report, BPMI.org.
[19] Becker J, Bergener P, Delfmann P, Eggert M, Weiß B (2011) Supporting business process compliance in _nancial institutions - a model-driven approach. In Wirtschaftsinformatik, page 75.
[20] Feltus C, Dubois E, Petit M. (2015). Alignment of ReMMo with RBAC to manage access rights in the frame of enterprise architecture. In 9th International Conference on Research Challenges in Information Science (RCIS). Pages 262-273. IEEE.
[21] Katranuschkov P, Gehre A, Scherer R J (2007) Reusable process patterns for collaborative work environments in AEC. In Proceedings of the 13th International Conference on Concurrent Enterprising.
[22] Yu E (1997) Towards modeling and reasoning support for early-phase requirements engineering. In Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, RE '97, pages 226, Washington, DC, USA. IEEE Computer Society.
[23] Sliwka D (2006) On the notion of responsibility in organizations. Volume 22(2).
[24] Spinello R A (1996) Case Studies in Information and Computer Ethics. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition. ISBN 013533845X.
[25] Mulgan R (2000) Accountability: An ever-expanding concept? Public Administration, 78(3): 555-573. URL http://dspace.anu.edu.au/handle/1885/41945.
[26] Laudon C K, Laudon P J (2001) Essentials of management information systems: Organization & technology in the networked enterprise.
[27] Sinclair A (1995) The chameleon of accountability: Forms and discourses. Accounting, Organizations and Society, 20(2-3):219-237, February 1995. ISSN 03613682.
[28] Feltus C, Petit M, Dubois E (2011). ReMoLa: Responsibility model language to align access rights with business process requirements. In Fifth International Conference on Research Challenges in Information Science (RCIS). Pages 1-6. IEEE.
[29] Bovens M (2010) Two concepts of accountability: Accountability as a virtue and as a mechanism. West European Politics, 33(5):946-967.
[30] Blind P K (2011) Accountability in public service delivery: A multidisciplinary review of the concept. Prepared for the Expert Group Meeting Engaging Citizens to Enhance Public Sector Accountability and Prevent Corruption in the Delivery of Public Services,.
[31] Dobson J, Martin D (2006) Enterprise modeling based on responsibility. In Trust in Technology: A Socio-Technical Perspective, volume 36, pages 39-67.
[32] Bovens M (2007) Analysing and Assessing Accountability: A Conceptual Framework. European Law Journal, 13(4):447-468.
[33] Jonathan A. F. (2007) The uncertain relationship between transparency and accountability. URL http://ideas.repec.org/p/cdl/glinre/410729.html.
[34] Dubnick M J (2007) Situating accountability: Seeking salvation for the core concept of modern governance (manuscript, university of New Hampshire).
[35] Feltus C, Dubois E, Petit M (2010). Conceptualizing a responsibility based approach for elaborating and verifying RBAC policies conforming with cobit framework requirements. In Third International Workshop on Requirements Engineering and Law (RELAW). Pages 34-43. IEEE.
[36] Amyot D, Horkoff J, Gross D, Mussbacher G (200) A lightweight GRL profile for i* modeling. In Proceedings of the ER 2009 Workshops (CoMoL, ETheCoM, FP-UML, MOST-ONISW, QoIS, RIGiM, SeCoGIS- on Advances in Conceptual Modeling - Challenging Perspectives, ER '09, pages 254-264, Berlin, Heidelberg, Springer-Verlag.
[37] Feltus C, Petit M, Sloman M (2016). Enhancement of business it alignment by including responsibility components in RBAC. Business/IT Alignment and Interoperability BUSITAL 2010. Page 61.
[38] Parent C, Spaccapietra S (2000) Database integration: The key to data interoperability. In Advances in Object-Oriented Data Modeling, pages 221-253.

[39] Zivkovic S, Kühn H, Karagiannis D (2007) Facilitate modelling using methodintegration: An approach using mappings and integration rules. In Hubert Osterle, Joachim Schelp, and Robert Winter, editors, ECIS, pages 2038-2049. University of St. Gallen,
[40] Day P, Klein R (1987) Accountabilities: five public services. Tavistock, URL http://www.worldcat.org/oclc/17412398.
[41] Fox J A (2007). The uncertain relationship between transparency and accountability. (410729), URL http://ideas.repec.org/p/cdl/glinre/410729.html.
[42] Sommerville I, Lock R, Storer T, Dobson J (2009) Deriving information requirements from responsibility models. In CAiSE '09: Proceedings of the 21st International Conference on Advanced Information Systems Engineering, pages 515-529, Berlin, Heidelberg, Springer-Verlag.
[43] Prendergast C G (1995) A theory of responsibility in organizations. Volume 13(3), pages 387-400.
[44] Feltus C, Petit M (2009). Building a responsibility model using modal logic-towards Accountability, Aapability and Commitment concepts. In International Conference on Computer Systems and Applications, 2009. Pages 386-391. IEEE.
[45] ISO27000 (2012) ISO/IEC 27000:2012 family of standards - Information Security Management System (included are. ISO/IEC 27000, 27002, 27006, 27005, 27004, 27003).
[46] Basel2 (2004) Basel Committee on Banking Supervision. (June 2004). International convergence of capital measurement and capital standards; a revised framework. Bank for International Settlements.
[47] SOX. Sarbanes-oxley act of 2002, united states code, pl 107-204, 116 stat 745. July 2002.
[48] Feltus C, Petit M., Dubois E (2009). Strengthening employee's responsibility to enhance governance of IT: COBIT RACI chart case study. In Proceedings of the first ACM workshop on Information security governance, page 23-32. ACM.
[49] Feltus C, Petit M (2009). Building a responsibility model including accountability, capability and commitment. In International Conference on Availability, Reliability and Security. Pages 412-419. IEEE.
[50] Bruno G, Torchiano M (2000) Process enabled information systems. In ICEIS, pages 32-37.
[51] Feltus C (2014). Aligning access rights to governance needs with the responsibility metamodel (ReMMo) in the frame of enterprise architecture (Doctoral dissertation, University of Namur - Public Research Centre Henri Tudor, Belgium - Luxembourg).
[52] Feltus C (2008). Preliminary Literature Review of Policy Engineering Methods; Toward Responsibility Concept. In 3rd International Conference on Information and Communication Technologies: From Theory to Applications. Pages 1-6. IEEE.
[53] Ackerman J M (2005) Social accountability in the public sector: A conceptual discussion.
[54] Bovens M (2007) Analysing and Assessing Accountability: A Conceptual Framework. European Law Journal, 13(4):447-468.
[55] Vernadat F (2002) Enterprise modelling and integration. In ICEIMT, pages 25-33.