Enterprise Architecture Enhanced with Responsibility to Manage Access Right - Case Study in an EU Institution

Petit, Michaël; Feltus, Christophe; VERNADAT, François

DOI:
10.1007/978-3-642-34549-4_10

Publication date:
2012

Document Version
Early version, also known as pre-print

Enterprise Architecture Enhanced with Responsibility to Manage Access Rights - Case Study in an EU Institution

Michaël Petit[*], Christophe Feltus[*,†] and François Vernadat[‡]

[*] PReCISE Research Centre, Faculty of Computer Science, University of Namur, Belgium.
[†] Public Research Centre Henri Tudor, Luxembourg, Luxembourg.
EE-Team[1], Luxembourg, Luxembourg.
[‡] Directorate for Information & Technology, European Court of Auditors, Luxembourg.
mpe@info.fundp.ac.be, christophe.feltus@tudor.lu, francois.vernadat@eca.europa.eu

Abstract. An innovative approach is proposed for aligning the different layers of the enterprise architecture of a European institution. The main objective of the alignment targets the definition and the assignment of the access rights needed by the employees according to business specifications. This alignment is realized by considering the responsibility and the accountabilities (doing, deciding and advising) of these employees regarding business tasks. Therefore, the responsibility (modeled in a responsibility metamodel) is integrated with the enterprise architecture metamodel using a structured method. The approach is illustrated and validated with a dedicated case study dealing with the definition of access rights assigned to employees involved in the user account provisioning and management processes.

Keywords: Access rights management, Business/IT alignment, Enterprise architecture, Responsibility, Case study.

1 Introduction

Access rights management is the process encompassing the definition, deployment and maintenance of access rights required by the employees to get access to the resources they need to perform the activities assigned to them. This process is central to the field of information security because it impacts most of the functions of the information systems, such as the configuration of the firewalls, the access to the file servers or/and the authorization to perform software operations. Furthermore, the management of access rights is complex because it involves many employee profiles, from secretaries to top managers, and concerns all the company layers, from the business to the technical ones. On one hand, access rights to IT components must be defined based on functional requirements (defining who can or must use which functionality) and, on the other hand, based on governance needs (defining which

[1] The Enterprise Engineering Team (EE-Team) is a collaboration between Public Research Centre Henri Tudor, Radboud University Nijmegen and HAN University of Applied Sciences. www.ee-team.eu

responsibility exists at the business level). The functional requirements advocate that, to perform an activity, the employee must hold the proper access rights. The governance needs are those defined by governance standards and norms and those aiming at improving the quality and the accuracy of these access rights [1].

Practically, one can observe [2] that the existing access control models [3, 4, 5, 6, 7, 8] and rights engineering methods [9, 10, 11] do not permit to correctly fulfill these needs, mostly because they are handled at the technical layer by isolated processes, which are defined and deployed by the IT department or by an isolated company unit that, generally, does not consider their management according to the governance needs. To address this problem, the paper proposes an approach based on the employees' responsibilities that are identified and modeled by considering these governance needs. On one hand, the modeling of the responsibility concept permits to consider several dimensions of the links that associate an employee with the activities he/she has to perform. On the other hand, the integration of the responsibility in a business/IT alignment method, for the engineering of access rights, permits to engineer and deploy the rights strictly necessary for the employees, thereby avoiding too permissive (and possibly harmful) access rights.

Enterprise architecture frameworks (EAFs) can be used to model the interrelations between different abstraction layers of a company (e.g. the business, the application and the technical layers) and, according to different aspects such as behavior, the information or the static structure [12]. These models provide views that are understandable by all stakeholders and support decision making, highlighting potential impacts on the whole enterprise. For instance, the enterprise architecture models can be used to understand the impact of a new business service integrated in the business layer on the technical layer and, consequently, enable analysis of some required server capacity. Conversely, the failure of a server has an impact on one or more applications and therefore on business services. The enterprise architecture models support analysis of the impact of various events or decisions and as such the improvement of alignment. For supporting the alignment between the enterprise layers, the EAFs have undergone major improvements during the first decade of the 2000's and some significant frameworks have been developed such as ArchiMate [12], the Zachman framework [13] or TOGAF [14]. Even if the advantages of EAFs are not to be demonstrated anymore, the high abstraction level of the modeled concepts and of the links between these concepts makes it sometimes difficult to use the EAFs to perform, verify or justify concrete alignments. In particular, EAFs do not permit to engineer precisely the access rights provided to the employee at an application layer based on the specification from a business layer.

The paper proposes a contribution to help solving the problem of alignment of access rights with business responsibility originating from governance requirements. The solution extends a particular EAF promoted by the European Commission and used at the European Court of Auditors (ECA) with concepts for representing responsibility at a business level. This extension is obtained by integrating the ECA EA metamodel with the responsibility metamodel of our previously developed Responsibility Modeling Language [2, 15]. The foreseen advantage of integrating both is the enhancement of the alignment among the concepts from the business perspective, the concepts from the application perspective and the concepts from the technical perspective (see Sect. 3). Ultimately, this alignment will support the

definition of the access rights to be provisioned to employees, based on their responsibilities. The applicability of the improved metamodel is demonstrated through a case study performed in a real setting.

The paper is structured as follows. In the next section, the responsibility metamodel is introduced. In Section 3, the ECA EA metamodel is presented and, in Section 4, both are integrated. In section 5, a case study related to the User provisioning and User account management processes is presented. Finally, in Section 6, some conclusions are provided.

2 Modeling responsibility

The elaboration of the responsibility metamodel (Fig. 1) has been performed on the basis of a literature review. As explained in previous papers [2, 15], in the first place, it is analyzed how the responsibility is dealt with in information technology professional frameworks, in the field of requirements engineering and role engineering and in the field of access right and access control models [15]. Then, this literature review was completed with an analysis of a state of the art on responsibility in the field of Human Sciences.

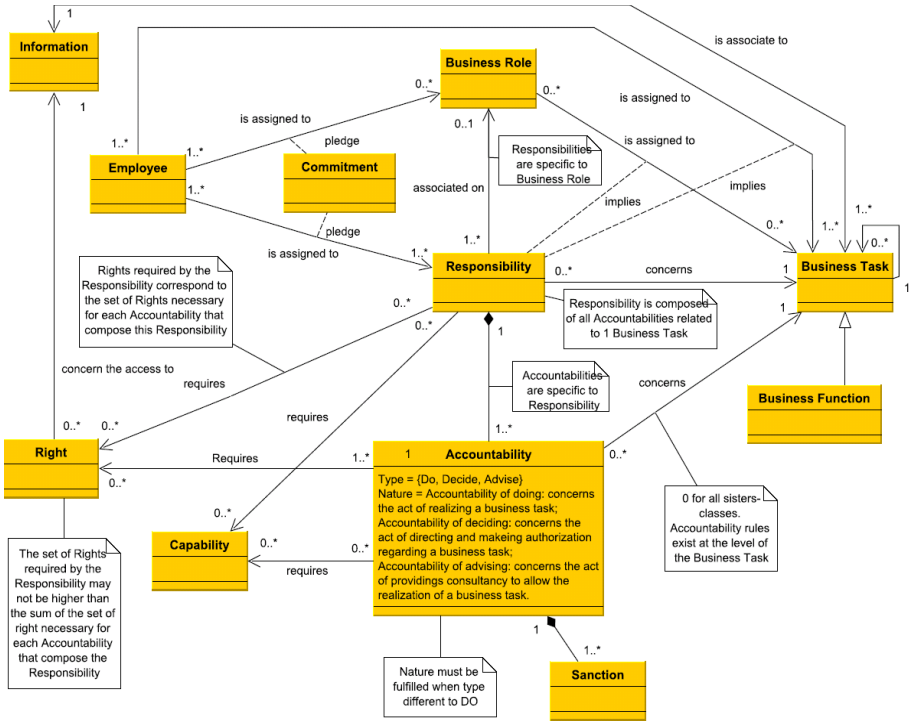

Fig. 1. The responsibility metamodel UML diagram.

The responsibility metamodel and its most meaningful concepts have been defined in previous works of the authors [16]. The most significant ones, for access rights management, are: the concept of ***responsibility,*** which is composed of all ***accountabilities*** related to one single ***business task*** and that, in order to be honored, require ***rights*** (the resources provided by the company to the employee, among which the access rights to ***information***) and ***capabilities*** (the qualities, the skills or the resources intrinsic to the employee). The ***accountability*** represents *the obligation related to what has to be done concerning a business task and the justification that it is done to someone else, under threat of* ***sanction(s)*** [16]. Three types of accountabilities can be defined: the accountability of doing which concerns the act of realizing a business task, the accountability of advising which concerns the act of providing consultancy to allow the realization of the task and the accountability of deciding which concerns the act of directing and making decisions and providing authorization regarding a business task. An ***employee*** is assigned to one or more responsibility, which may be, additionally, gathered in ***business role(s)***.

3 ECA EA metamodel

To support the management of its information systems (IS), the European Commission has developed a dedicated architecture framework named CEAF[2] that has been deployed in several other European institutions and especially the European Court of Auditors (ECA). The particularity of the CEAF is that it is business and IT oriented and provides a framework for the business entities in relation with IT usage and supporting infrastructure. Considering the business as being at the heart of the framework allows continual business/IT alignment. In addition to its four perspectives, namely "business", "functional", "application" and "data", the CEAF also contains a set of architecture standards that gather methods, vocabulary and rules to comply with. One such rule is, for instance, at the business layer, that *the IT department of ECA (DIT) responsible for the management of information technology, needs to understand the business activities to automate them*. The DIT has defined its own enterprise architecture metamodel, the ECA EA metamodel based on the CEAF (see Fig. 2). This ECA EA is formalized using an entity-relationship model and is made operational using the Corporate Modeler Suite[3]. It is made of the same four vertical layers as the CEAF, each representing a perspective in the architecture, i.e.:

- The **business layer,** formalizing the main business processes of the organization (process map and process flows in terms of activities).
- The **functional layer,** defining the views needed to describe the business processes in relation with business functions and services.
- The **application layer,** describing the IT applications or ISs and the data exchanges between them.
- The **technical layer,** describing the IT infrastructure in terms of servers, computers network devices, security devices, and so forth.

[2] CEAF means European Commission Enterprise Architecture Framework.
[3] Modeler Suite from CaseWise (http://www.casewise.com/products/modeler)

Each layer includes a set of generic objects, relevant for the layer, and may contain different types of views. Each view is based on one diagram template (Fig. 2). The concepts which are relevant in the context of this paper (i.e. to be integrated with the one of the responsibility metamodel) are described in the next section.

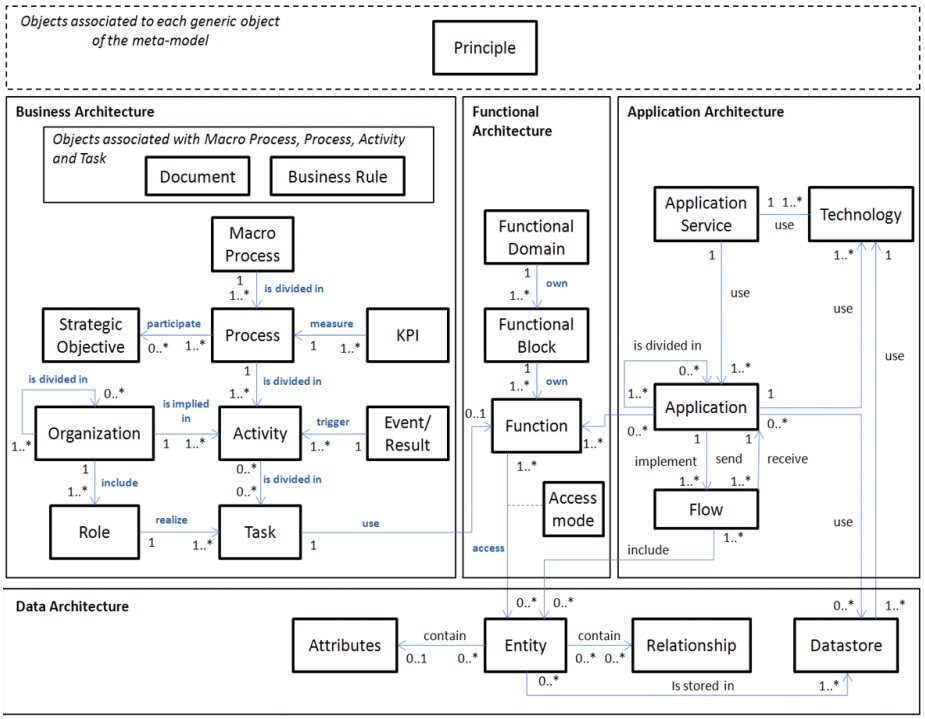

Fig. 2. ECA EA metamodel UML diagram

4 Integrated ECA EA-responsibility metamodel

In this section, the integration of the ECA EA metamodel with the responsibility metamodel is presented. The method proposed by [17] was used for integrating the metamodels. The three steps of the method are (1) preparation for integration, (2) investigation and definition of the correspondences and (3) integration of both metamodels.

4.1 Preparation for integration

Preparing the integration first goes through a primary activity for selecting the subset of concepts from the metamodels relevant for integration. Secondly, a common language for representing both metamodels is selected.

1) Subset of concepts concerned by the integration

This activity of selecting the appropriate subset of concepts considered for the integration has been added to the method of [17] and is required to address the

concepts from the metamodels that are meaningful for the assignment of accountabilities regarding business tasks to the employees and for the definition of the rights and capabilities required therefore. The subset of concepts concerned by the integration, in the ECA EA metamodel of Fig. 2, includes:

- The concept of **role**. This concept is used, according to the ECA EA metamodel documentation, to represent the notion of entity executing a task of a process. It is associated to the concept of a task that it realizes and to the concept of organization to which it belongs.
- The concept of **task**. This concept is used to describe how the activities are performed. A task is achieved by a single actor (not represented in the ECA EA metamodel), is performed continuously and cannot be interrupted. The task is associated to the concept of role which realizes it, to the concept of activity that it belongs to and to the concept of function that it uses.
- The concept of **function**. This concept enables to break-down an IS in functional blocks and functionality items within functional domains. A function block is defined by the business concepts that it manages on behalf of the IS, combining the functions (functions related to business objects) and production rules of the data that it communicates. It is associated to the concept of task, of IS (the application) that implements it and of entity that it accesses in a CRUD mode (Create, Read, Update and Delete).
- The concept of **entity**. This concept represents the business data items conveyed by the IS or handled by an application. In the latter case, it refers to information data. It means that the physical data model implemented is not described in systems/database. The entity is accessed by the function, is associated to flow, is defined by attributes and relationships and is stored in a datastore.
- The concept of **application**. This concept represents a software component that contributes to a service for a dedicated business line or for a particular system. Regarding its relation with other concepts: the application is used by the application service, is made of one or more other application(s), uses a technology, sends and receives flow items and implements functions.

In the responsibility metamodel (see Sect. 2), the following concepts defined in [16] are kept: responsibility, business role, business task, right, capability, accountability and employee.

2) Selection of a common representation language

For the integration step, UML is used because it is precise enough for this purpose, standard and commonly used. As a consequence, the ECA EA metamodel formalized using the entity-relation model has been translated into a UML class diagram (Fig. 2).

4.2 Investigation and definition of the correspondences

In [17], the author explains that this second step consists in analyzing the correspondences between classes of the two metamodels. These correspondences exist if correspondences among pairs of classes exist and if correspondences between instances of these classes taken pair-wise can be generalized. The correspondences can be identified by analyzing the semantic definitions of the classes and can be validated on instances in models created by instantiating both metamodels for different case studies. Based on the definitions of concepts and on the authors' experience with the case study presented in Sect. 5, three correspondence cases between the concepts of the ECA EA metamodel and the responsibility metamodel have been identified:

- **Role** from the ECA EA metamodel and **business role** from the responsibility metamodel: the concept of role in the ECA EA metamodel is represented in the business architecture, is an element that belongs to the organization and realizes business tasks. Hence, it reflects a business role rather than an application role and corresponds, as a result, to the business role of the responsibility metamodel (cf. application role / Role Based Access Control [15]).
- **Entity** from the ECA EA metamodel and **information** from the responsibility metamodel. The concept of entity in the ECA EA metamodel is equivalent to the concept of information from the responsibility metamodel. Instances of both concepts are accessed by a human or by an application component and specific access rights are necessary to access them.
- **Task** from the ECA EA metamodel and **business task** from the responsibility metamodel. The concept of task in the ECA EA metamodel and the concept of business task from the responsibility metamodel semantically have the same meaning. The task from the ECA EA metamodel composes the business architecture and corresponds to a task performed on the business side. According to the definition of the ECA concept, it can be noticed that the task is performed by a single actor. This is a constraint that does not exist in the responsibility metamodel and that needs to be considered at the integration step.

4.3 Integration of metamodels

The third step defined in [17] corresponds to the integration of both metamodels. During the analysis of the correspondences between the metamodel concepts, some minor divergences have been observed. Notwithstanding the influence of these divergences, to consider that a sufficient correspondence exists between the elements and to consider them during this third step of integration, these divergences are analyzed in depth and the correspondence rules formalized in order to obtain a well defined and precise integration.

Consequently, to construct the integrated metamodel that enriches the ECA EA metamodel with the responsibility metamodel, a set of integration rules has been defined. Therefore, it is decided that (1) when a correspondence exists between one concept from the ECA EA metamodel and one concept from the responsibility metamodel, the name of the concept from the ECA EA metamodel is preserved, (2) when the concept of the responsibility metamodel has no corresponding concept in the ECA EA metamodel, this concept is integrated in the integrated metamodel and the name from the responsibility metamodel is used, (3) when a correspondence exists with conflicts between the definition of the concepts, the concepts are integrated in the integrated metamodel, the name of the concept from the ECA EA metamodel is preserved and additionally integration constraints to be respected are included in the case of using the integrated metamodel. Finally, (4) when concepts differently exist in both metamodels, the integration preferences are motivated case by case. In the sequel, correspondences between classes are first considered and then correspondences between associations between classes.

1) UML Classes integration

a) Classes that correspond exactly:

The **role** from the ECA EA metamodel and the **business role** from the responsibility metamodel exactly match. The **entity** from the ECA EA metamodel and the **information** from the responsibility metamodel also exactly match.

b) Classes that only exist in one metamodel

Employee, **responsibility**, **right** and the type of rights to access information, **capability** and **accountability** only exist in the responsibility metamodel. **Function** only exists in the ECA EA metamodel.

c) Classes that correspond under constraints

The **business task** from the responsibility metamodel and the **task** from the ECA EA metamodel correspond partially. In the ECA EA metamodel, a task is performed by a single actor. The ECA EA metamodel description does not define the granularity level of a business task and, for instance, does not define if "doing a task", "advising for the performance of a task" or "making decision during the realization of a task" are considered as three tasks or as a single one. In the first case, three actors may be assigned separately to each of the three propositions although, in the latter case, only one actor is assigned to it. In the responsibility metamodel, many employees may be assigned to many responsibilities regarding a business task. It can be observed that, in practice, this is often what happens for responsibility, for instance in courts during trials. Therefore, it can be considered, in the integrated metamodel, that a task may be concerned by more than one accountability, themselves composing responsibilities assigned to one or more employees. For instance, let us consider the task to deploy a new software component on the ECA network. There is a first responsibility to effectively deploy the solution. This responsibility is assigned to an IT system administrator who is accountable towards the manager of his unit. This means that he must justify the realization (or absence thereof) of the deployment and that he may be sanctioned positively/negatively by the unit manager. The latter, concerning this deployment, is responsible to make the right decisions, for instance, to decide the best period of the day for the deployment, to give the go/no go for production after performing test, and so forth. This responsibility is directly handled by the unit manager who must justify his decision and is sanctioned accordingly by his own superior, for instance, the department manager, and so forth. This illustration explains how many responsibilities may be related to the same task but assigned to various employees or roles.

d) Classes that exist differently in both metamodels

The concept of ***access right*** from the responsibility metamodel and the concept of ***access mode*** from the ECA EA metamodel are represented differently. The concept of access right is a type of rights in the responsibility metamodel which semantically corresponds to an access mode in the ECA EA metamodel. In the ECA EA metamodel, the entity is accessed by the concept of function that, additionally, is associated to a task and to an application of the IS that implements it. As a result, the access right is already considered in the ECA EA metamodel, but it is directly associated to the concept of task by the intermediary of function. In the integrated metamodel, the concept of function that is interesting to consider as allowing the connection between concepts from the business architecture, from the application architecture and from the data architecture, is preserved. However, to restrict the usage of a function only for what is strictly necessary, it is not considered that it is associated to a task, but that it is required by a responsibility and necessary for accountability. As such, an employee with the accountability of doing a task gets the right to use a certain function, an employee with the accountability of deciding about the execution of a task gets the right to use another function, and so forth. For example, to record an invoice, a bookkeeper requires the use of the function "encode new invoice". This function is associated to a write access to the invoicing data.

Additionally, the financial controller who controls the invoice requires the use of the "control invoice" function that is associated to a read access to the same invoicing data.

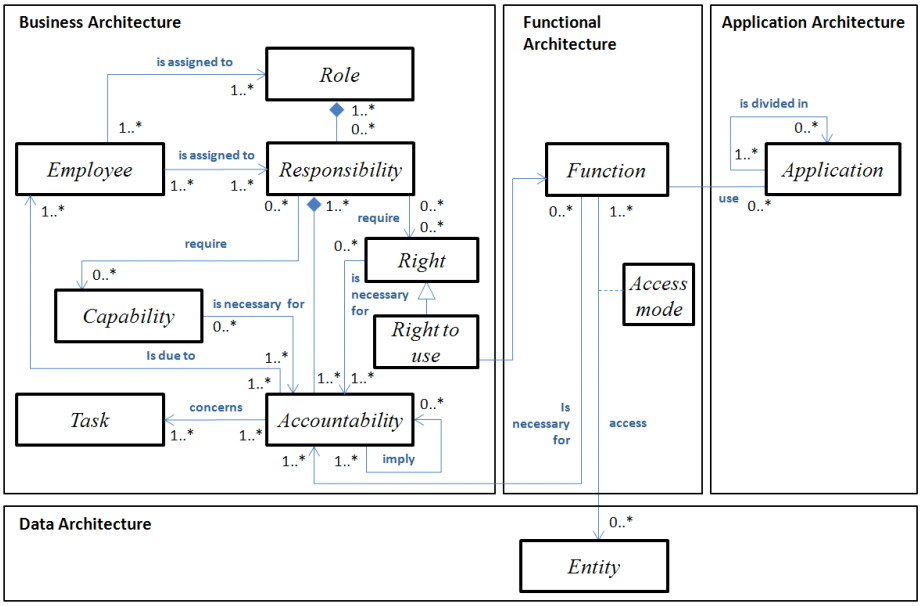

Fig. 3. The responsibility metamodel integrated with the ECA EA metamodel

2) UML associations integration

a) UML associations from the responsibility metamodel that complete or replace, in the integrated metamodel, the UML associations from the ECA EA metamodel

The direct UML association between a role and a task in ECA EA metamodel is replaced by a composition of associations: "a business role is a gathering of responsibilities, themselves made of a set of accountabilities concerning a single business task". This composition is more precise and is therefore retained. The UML association between the task and the function it uses in the ECA EA metamodel is replaced by two UML associations: "an accountability concerning a single business task requires right(s)" and "one type of right is the right to use a function"

b) UML associations from the responsibility metamodel, that do not exist in the ECA EA metamodel

The following associations are present only in responsibility metamodel and are simply included in the integrated metamodel: "a responsibility requires capabilities", "a responsibility requires rights", "an employee is assigned to one or more responsibility(ies) and to one or more business role(s)", "a capability is necessary for a business task" and "a right is necessary for a business task".

The metamodel resulting from the integration is shown in Fig. 3.

5 OIM process case study

This section reports on the exploitation of the integrated metamodel developed in the previous section on a real-world case study from a European institution in order to validate its applicability and its contribution to the engineering of more accurate access rights. The integrated metamodel was applied for the management of the access rights provided to employees involved in the *User provisioning and User account management processes*. The case study has been performed over fourteen months, from January 2011 to February 2012. During this period, twelve meetings were organized with the DIT managers of the institution and with the access right administrator to model and assess the processes and to elaborate and assign a set of thirteen responsibilities.

5.1 Process description

The user provisioning process is about providing, adapting or removing access rights to a user depending if he is a newcomer arriving at the Court, an employee or an external staff member whose status or job changes or if he is temporarily or definitely leaving the Court. Employee or external staff status changes when, for instance, his job category, department or name changes or when the end date of his contract is modified. The management of the users' identity and access rights are areas in which the DIT is hugely involved. Indeed, since each employee of the ECA needs different access rights on the various ISs, these access rights must be accurately provided according to the user profile.

To manage these rights, the DIT has acquired the Oracle Identity Management (OIM) tool. This tool is central to the identity and user accounts management activity and, as illustrated by Fig. 4, is connected, on the one hand, to the applications that provision the user profiles (COMREF and eAdmin[4]) and, on the other hand, to the user directories that provision access rights rules (eDir, Active Directory (AD), Lotus Notes (LN), and so forth). COMREF is the central human resource database of the European Commission used by the HR management tool Sysper2[5]. The main COMREF database is located in the EC data center and contains a set of officials and employees' information items such as the type of contract, occupation, grade, marital status, date of birth, place of work, department, career history and so forth. This information is synchronized every day with the COMREF_ECA[6] datastore and with the OIM tool. In parallel, additional information is also uploaded in the OIM tool for the subset of data relative to ECA workers (employees or external staff), directly from the ECA, e.g. the office number, the entry ID card, the phone numbers, the telephone PIN code, and so forth. This information is also daily synchronized with the central COMREF database.

At the business layer, processes have been defined to support the activities of the employees who manage (such as the system administrators) or use the system (such as the secretaries who fill in the data related to the PIN codes or phone numbers). The case study focuses on one of these processes, the user provisioning and user account management process. This process aims at defining an ordinate set of tasks to manage

[4] eAdmin is a tool to manage administrative data such as office numbers

[5] Sysper2 is the Human Resource Management solution of the European Commission that supports the personnel recruitment, career management, organization chart, time management, etc.

[6] COMREF_ECA is a dedicated mirror in Luxembourg of the COMREF database for the officials and employees of the ECA

the request, establishment, issue, suspension, modification or closure of user accounts and to, accordingly, provide the employees with a set of user privileges to access IT resources. More specially, the case study focuses on the evolution of this process, due to some recent enhancement of the automation of the provisioning loop between the COMREF database and OIM, and on the new definition of the responsibilities of the employees involved in this process.

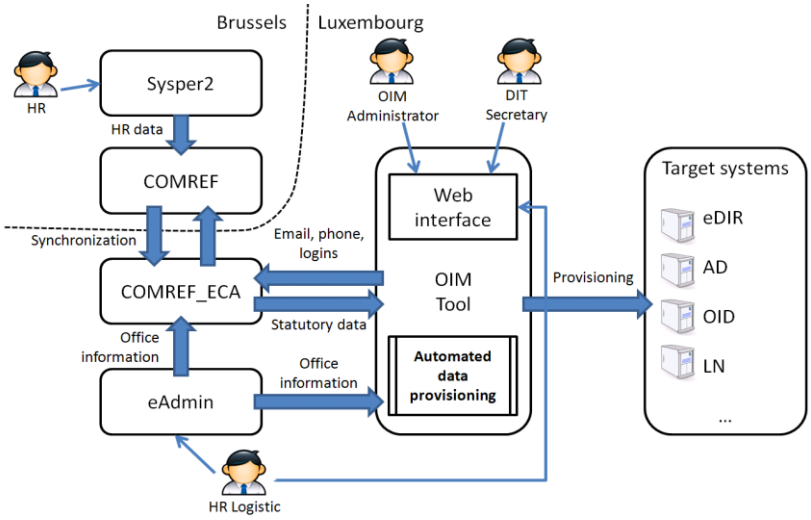

Fig. 4. Overview of the ECA OIM architecture

5.2 Definition and assignment of the responsibilities

A sequence of four steps is applied to model the responsibilities of the employees involved in the upgraded user provisioning and user accounts management process.

1) Identification of business tasks

The business tasks are defined by instantiating the concepts of task from the integrated metamodel (Fig. 3). In this step, the tasks for which responsibilities have to be defined are identified, but tasks that are performed by an application component and for which defining the responsibility is inappropriate according to the definition of the responsibility in Sect. 2 are not considered. After the provisioning process enhancement, six tasks are remaining. These tasks are: "Release Note d'information[7]", "Complete Sysper2 data entry", "Assign an office number using eAdmin", "Assign a phone number and a PIN code", "Enter phone number and PIN code in OIM" and "Perform auto provisioning and daily reconciliation".

2) Identification of the accountabilities

The accountability, as explained in Sect. 2, defines which obligation(s) compose(s) a responsibility for a business task and which justification is expected. In the ECA EA-responsibility metamodel, this concept of accountability has been preserved since it is important to distinguish what really are the accountabilities of the ECA employees regarding the business tasks. In this step, for each of the tasks, the existing accountabilities are reviewed for each of the responsibilities. Mainly, three of them

[7] In English: Information note

have been retained. The obligation to "Do" that composes the responsibility of performing the task, the obligation to "Decide about" that composes the responsibility of being accountable for the performance of a task and the obligation to "Advise" that composes the responsibility to give advice for the performance of the task. For example, three types of accountability concern the task "Assign a phone number and a PIN code" and the task "Assign an office number using eAdmin". Three examples explained later in the text are provided in Tables 1-3.

Table 1. Responsibility OIM 7.

Responsibility OIM 7	
Task	Assign an office number using eAdmin
Accountability	Doing
Employee	Barbara Smith
Accountable towards	Reynald Zimmermann
Backup	Antonio Sanchis
Role	Logistic administrator
Backup Role	Logistic Head of Unit
Right	Read-Write access in eAdmin
Capability	eAdmin manipulation training

Table 2. Responsibility OIM 1.

Responsibility OIM 1	
Task	Release "Note d'Information"
Accountability	Doing
Employee	All
Accountable towards	Gerald Hadwen
Role	Human Resources Directorate/ RCD
Backup Role	RCD Unit Manager
Right	Read HR workflow, Read Information Note template and Use editing tool
Capability	Ability to edit official documents and HR training
Task	Release "Note d'Information"

Table 3. Responsibility OIM 10.

Responsibility OIM 10	
Task	Enter phone number and PIN code in OIM
Accountability	Deciding
Employee	Francis Carambino
Accountable towards	Marco Jonhson
Backup	Philippe Melvine
Role	OIM Administrator
Backup Role	IAM Service Manager
Right	Read-Write access to OIM tool-Phone number application and Read-Write access to OIM tool-PIN code application
Capability	Computer sciences education, two years experience in OIM administration

1) Identification of the rights and capabilities

The rights and capabilities are elements required by a responsibility and necessary to achieve accountabilities (Fig. 1). Both concepts have, naturally, been introduced in the integrated metamodel in Fig. 3. In this step, it is analyzed, accountability by accountability, which capabilities and which rights are necessary to realize the accountability. In the integrated ECA EA-responsibility metamodel, the access right (which is a type of right) is no more directly associated to the realization of an action

involving an information (e.g. read a file), but is a right to use a function that realizes, together, an action (e.g.: CRUD) regarding an entity and the use of an application that manipulates this entity. For instance, the Responsibility OIM 7 (Table 1) assigned to Barbara Smith requires using the function that realizes Read-Write access in eAdmin.

Once the responsibilities have been modeled, they can be assigned to employees, considering their role in the organization. As explained in Fig. 3, a responsibility may be assigned directly to an employee or to a role.

2) Assignment of the responsibilities to the employees

In the case study, some responsibilities are directly assigned to employees and others are assigned to roles. For instance, the Responsibility OIM 1 (Table 2) is made of the accountability to do the task "Release Note d'information". This responsibility is assigned to the role Human Resources Directorate/ RCD (recruitment career development), although the Responsibility OIM 10 (Table 3) is made of the accountability to verify the task "Enter Phone number and PIN code in OIM" and is assigned directly to the employee Francis Carambino.

5.3 Case study analysis

The instantiation of the responsibilities, after the mapping of the responsibility metamodel with the ECA EA metamodel, brings a set of thirteen responsibilities, from which the following results are observable.

1) Better definition of accountabilities of employees regarding the tasks

Before the case study was performed, the description of the process according to the sole ECA EA metamodel did only provide a list of the roles responsible to perform the tasks. As a result, this description was not accurate enough to know which employees perform which tasks, and which other employees decide about it, give advice and so forth. For instance, some employees did not appear in the process description, although they were involved in it. This was for instance the case of the IAM[8] Service Manager. The description of the process, according to the integrated metamodel gives a clear view on all the accountabilities and their assignments to the employees.

2) Explicit formalization of capabilities required by employees to meet their accountabilities

Before the case study, the description of the process did not address the employee capabilities necessary to perform accountabilities. Employees were assigned to responsibilities without previously knowing if they were capable of assuming them. The description of the process, according to the integrated metamodel, clearly highlights the capabilities necessary to perform the tasks. For instance, to "Complete Sysper2 data entry", the employee needed both a Sysper2 and SQL training and, if someone else is assigned to this responsibility, the same training is required.

3) Explicit formalization of the rights and access rights required by the employees to meet their accountabilities

Another difference in the process description after the case study is that the right, and more specifically the access rights, needed to perform an accountability are clearly enumerated. For instance, to "Complete Sysper2 data entry", it is necessary to have the access right to Read-Write and Modify all Sysper2 functions and the right to use another system called RETO[9].

---

[8] Identity and Access Management

[9] RETO (Reservation TOol) is a personal identification number booking tool common to all institutions

4) Possibility to associate tasks to responsibilities or to roles

The final improvement is the possibility to assign a task, either to a role or to a responsibility rather than directly to an employee. This possibility offers more flexibility and reduces the risk of providing access rights to employees that do not need them. As an example, all employees with the role of Human Resources Directorate/ RCD are assigned to the responsibility to "Release Note d'information", although only one employee advises about the assignment of offices. Some other concepts of the responsibility metamodel have not been introduced yet in the integrated metamodel and have not been illustrated in the case study. Indeed, as explained in Section 2, checking the employee's commitment during the assignment of a responsibility or a role was not in the scope of this case study. However, some other cases in the ECA have shown that the commitment influences the way employees accept their responsibilities. For instance, in 2010, ECA bought a highly sophisticated tool to support problems management. During the deployment of the tool in production, the employees have not been informed about their new responsibilities related to the usage of the tool. As a result, they did not commit to these responsibilities and the tool has not been used properly or up to the expectations. The same problem occurred at a later stage when a decision was made to use a tool to manage the CMDB[10].

6 Conclusions

The paper has presented a method to improve the alignment between the different layers of an enterprise architecture metamodel and, thereby, to enhance the management of access rights provided to employees based on their accountabilities. This method is based on the integration of an enterprise architecture framework with a responsibility metamodel. The integration of both metamodels has been illustrated using a three-step approach proposed by [17] and has been applied to the ECA EA metamodel, an EAF of a European institution. A validation has been realized on a real case study related to the user provisioning and user account management processes. The objectives of this case study were to validate (1) the applicability of the integrated metamodel and (2) the engineering of more accurate access rights comparing to the solutions reviewed in [16]. The validation has been performed in four phases. First, the accountability of the employees regarding the tasks of the process has been defined. Next, the capabilities required to perform these accountabilities have been formalized. Thirdly, the required rights and access rights have been formalized. Finally, the employees have been associated to responsibilities or to roles. The output of these phases was a set of thirteen responsibilities. The validation shows that using the combination of the ECA EA and the responsibility metamodel brings benefits compared to using ECA only. Additionally, compared to the other approaches, the method offers other possibilities and advantages, including more precise definition of accountabilities of employees regarding tasks, explicit formalization of the rights and capabilities required by the employees to perform the accountabilities (traceability between accountabilities and rights), and formal associations of employees to responsibilities or to business roles. The approach has also been validated, in parallel, with other processes from the healthcare sector and are available in [18].

[10] Configuration Management Database, in accordance with ITIL

Acknowledgements

This work has been partially sponsored by the *Fonds National de la Recherche Luxembourg*, www.fnr.lu, via the PEARL programme

References

1. C. Feltus, M. Petit, F. Vernadat, Enhancement of CIMOSA with Responsibility Concept to Conform to Principles of Corporate Governance of IT, 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'09), 2009, Moscow, Russia.
2. C. Feltus, M. Petit, and E. Dubois, Strengthening employee's responsibility to enhance governance of IT: COBIT RACI chart case study. 1st ACM Workshop on Information Security Governance. ACM, New York, NY, 2009.
3. D. Clark and R. Wilson. A comparison of commercial and military computer security policies. Security and Privacy, IEEE Symposium on, 0:184, 1987.
4. M. J. Covington, W. Long, S. Srinivasan, A. K. Dev, M. Ahamad, and G. D. Abowd. Securing context-aware applications using environment roles. Symposium on Access Control Models And Technologies (SACMAT '01): pp. 10-20, New York, NY, USA, 2001.
5. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli. Proposed nist standard for role-based access control. ACM Trans. Inf. Syst. Secur., 4(3):224-274, 2001.
6. A. H. Karp, H. Haury, and M. H. Davis. From abac to zbac: The evolution of access control models. Control, 2009.
7. M. Covington and M. R. Sastry. A contextual attribute-based access control model. On the Move to Meaningful Internet Systems, OTM 2006 Workshops, pp. 1996-2006.
8. B. Lang, I. Foster, F. Siebenlist, R. Ananthakrishnan, T. Freeman. A exible attribute based access control method for grid computing. Journal of Grid Computing, 7(2): 169-180, 2008.
9. R. Crook, D. Ince, and B. Nuseibeh. Modelling access policies using roles inrequirements engineering. Information and Software Technology, 45(14):979-991, 2003.
10. Q. He and A. I. Anton. A framework for privacy-enhanced access control analysis in requirements engineering. In Proc. of the 9th Requirements EngineeringFoundation for Software Quality (REFSQ 09), 2003.
11. G. Neumann and M. Strembeck. A scenario-driven role engineering process for functional rbac roles. In SACMAT '02, New York, NY, USA, 2002. ACM.
12. M Lankhorst (ed.) and the ArchiMate team (2004), ArchiMate Language Primer.
13. Zachman, John A. The Zachman Framework For Enterprise Architecture: Primer for Enterprise Engineering and Manufacturing By. Engineering, no. July: 1-11, 2003.
14. Open Group, The. TOGAF (The Open Group Architecture Framework). 2009.
15. C. Feltus, M. Petit, and M. Sloman, Enhancement of Business IT Alignment by Including Responsibility Components in RBAC, 5th Busital workshop, Hammamet, Tunisia, 2010.
16. C. Feltus, M. Petit, E. Dubois, *ReMoLa*: Responsibility Model Language to Align Access Rights with Business Process Requirements, Fifth International Conference on Research Challenges in Information Science (RCIS 2011), May 19-21, 2011, Gosier, Guadeloupe.
17. M. Petit. Some methodological clues for defining a unified enterprise modelling language. In proc. of the International Conference on Enterprise Integration Modeling Technology (ICEIMT '01), pp. 359-369, Deventer, The Netherlands, 2003.
18. C. Feltus, E. Dubois, E. Proper, I. Band, M. Petit, Enhancing the ArchiMate® Standard with a Responsibility Modeling Language for Access Rights Management, in Proc. of the 5th ACM International Conference on Security of Information and Networks (SIN 2012), India.