2014 IEEE 16th Conference on Business Informatics

Modeling Access Control Transactions in Enterprise Architecture

Khaled Gaaloul
Public Research Centre
Henri Tudor
Luxembourg
Email: khaled.gaaloul@tudor.lu

Sérgio Guerreiro
Lusófona University
Campo Grande, 376, 1749-024 Lisbon, Portugal
Email: sergio.guerreiro@ulusofona.pt

Hend*erik* A. Proper
Public Research Centre
Henri Tudor
Luxembourg
Email: erik.proper@tudor.lu

***Abstract*—Enterprise architecture (EA) aims to provide management with appropriate indicators and controls to steer and model service-oriented enterprises. However, the management of EA models change is a challenging task due to complex dependencies when dealing with security constraints such as access control. In this paper, we motivate the use of an access control model in EA. More specifically, we present the role-based access control (RBAC) standard as a mean to model access control transactions in EA. To that end, we present (i) how the concepts of RBAC can be modeled into the ArchiMate enterprise architecture modeling language, and (ii) how RBAC's enforcement is supported with the DEMO enterprise modeling methodology via the business transaction concept. These attempts will help us to identify the conceptual link between RBAC, ArchiMate, and DEMO meta models in order to create a consistent lightweight model for access control in EA. Finally, we illustrate the application of the proposed approach through the handling of an e-Government scenario.**

I. Introduction

Enterprise Architecture (EA) is generally considered to provide a good steering instrument to analyze the current state of the enterprise (As-is), identify and describe alternative future states (To-be), guard the cohesion and alignment between the different aspects of an enterprise such as business processes and their ICT (Information and Communications Technology) support [1].

ArchiMate is an Open Group standard [1], [2] for the modeling of enterprise architectures[1], emphasizing a holistic, but complete, view of the enterprise. This means that architects can use ArchiMate to model, amongst others, an organization's products and services, how these products and services are realized/delivered by business processes, and how in turn these processes are supported by information systems and IT infrastructure.

Security is nowadays considered, by the industry, as a major concern that has gained increasing focus to research for new solutions. An example of such effort are the recent calls for cyber security projects research under the scope of the digital agenda for Europe, by the initiative Horizon 2020[2].

ArchiMate lacks security guidelines for modeling an enterprise from access control perspectives [3], [4], [5]. ArchiMate follows a coarse grained approach and, more important, ArchiMate models are mainly used to share a common understanding of the organization between stakeholders having different interpretations of it, but, these models are usually workaround when information systems development begins. For instance, the ArchiMate [2] business roles separation implicitly bounds the capability of an unauthorized access to the artifacts. Moreover, usually EA models do not reflect explicit access control requirements, ending up as the responsibility of the developer to implement the fine-grained security mechanisms properly. Security is an architectural dimension that, since design time, should be fully and explicitly prescribed. Furthermore, a fine-grained access control prescription obliges the full specification of business transactions dynamics.

There exist several IT Governance frameworks that have some focus on enterprise security. One of the most known frameworks is the Control Objectives for Information and related Technology (COBIT) [6] which has specific internal IT related goals with security (e.g., security of information, processing infrastructure and applications). There is also the ISO/IEC 2700 standard [7] which has a practice guide addressing access control issues from an IT perspectives. Moreover, EA frameworks such as Zachman framework [8] and the Open Group Architecture Framework (TOGAF) include principles and guidelines which provide an overall recommendation to secure enterprise information architectures. Nevertheless, neither standards nor frameworks are at the same level of abstraction when modeling access control. Organizational and business/IT alignments are poorly described. For instance, the conceptualization definition and granularity of access control are still missing by EA modelers where access control goes from specification to enforcement, which is the contribution of this paper..

In this paper we conduct an initial experiment about access control management in EA. The contribution can be summarized in twofold: (1) the first part presents the interest of considering access control in existing EA languages. In doing so, we introduce the role-based access control (RBAC) standard [9], [10], and experiment its relevance to ArchiMate. We conduct the same experience with DEMO methodology [11]. (2) The second part evaluate both experiments where we show common features between RBAC, ArchiMate, and DEMO. Despite the relevance and interest, we observe some limitations when modeling then enforcing access control in ArchiMate and DEMO. To address this issue, we present an approach based

[1] http://www.opengroup.org/archimate/
[2] http://ec.europa.eu/digital-agenda/en/cybersecurity

978-1-4799-5779-8/14 $31.00 © 2014 IEEE
DOI 10.1109/CBI.2014.26

on the conjunction of RBAC for the access control definition and DEMO for its enforcement using transaction concepts to bridge the core concepts from RBAC to ArchiMate. The goal is to define a lightweight model supporting access control in ArchiMate.

This research is based on a simplification of the design-science research (DSR) as proposed by Hevner [12] and Winter [13]. The methodology applied is divided according to the two processes of design science research in information system, *Build* and *Evaluate*. The build process is composed by two stages: conceptual definition and conceptual model construction; whereas and the evaluation process is composed by only one stage of a use case. The first stage, conceptual definition, has two main milestones: concepts domain and domain definitions with regards to RBAC, ArchiMate, and DEMO domains. We identify and define relevant concepts for EA supporting access control (see Sect. II and Sect. III). Furthermore, an analysis of the relations between concepts is required to understand the conceptual model that is constructed in Sect. IV. The second stage, evaluation, is done based on the observational case study and an experimental design. The evaluation part is illustrated using an e-Government case study in Sect. V. Finally, we conclude and present future work in Sect. VI.

II. Access Control

Organizations use access control mechanisms to mitigate the risks of unauthorized access to their data, resources, and systems. An access to a resource is determined based on the relationship between the requester and the organization or owner in control of the resource. In other words, the requester's role will determine whether access will be granted or denied. Several access control models exist to address changes in organizational structures, technologies, organizational needs, technical capabilities, and organizational relationships.

The RBAC model is a widely implemented mechanism for protecting system resources standardized by the American National Standard for Information Technology (ANSI). The RBAC model needs only to be made to role assignments, which are significantly fewer than individual assignments [10]. The RBAC model relies on user authentication, which in turn relies on identity management and defines relationships between the main concepts of Users, Roles and Permissions. RBAC's constraints restrict permissions depending on contextual information such as separation of duties (SoD) [14]. The RBAC model is presented in Fig. 1 and its core concepts are textually described in Table I based on the foundational work of [9].

The limitations identified in the RBAC model are related to the nature of permissions (P) which is not specified in the RBAC standard. A permission is an association between a transformation procedure and an object. It can be fine-grained or coarse-grained and each system implementation has to decide which kind of granularity fits. For instance, a permission can be thought as an object-method pair or a class-method pair in an object-oriented environment. In this paper, we consider a permission as an approval of executing (i.e. operation) an object (i.e. resource) [5].

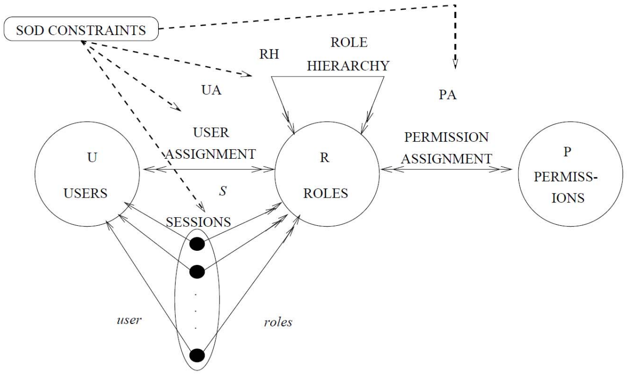

Fig. 1. The RBAC model (Adapted from [9])

TABLE I. RBAC concepts, obtained from [9]

Concept	Definition
User (U)	A Person with a given U in a given system. Often human, but can also be systems.
Role (R)	R is a responsibility, *e.g.*, job functions defined for an organization.
Role Hierarchy (RH)	It defines a partially ordered role hierarchy in an organization.
User Assignment (UA)	Each U has a set of associated R.
Permission (P)	An approval of a mode of access to a resource.
Session (S)	S is a mapping between U and possibly many R. During a session, we define a *Subject* as a single user associated with an active role.
Permission Assignment (PA)	Each R has a set of associated P.
SoD Constraint	Applied to: sessions, UA, RH and PA. Constraints restrict permissions depending on contextual information by ensuring that mutually exclusive roles must be invoked to complete a sensitive task.

III. Access control in Enterprise Architecture

This section is about the *Build process* of the design-science research methodology. We present the conceptual definition step where we motivate the need of RBAC in EA. We start with the ArchiMate modeling language where access control requirements can be modeled using RBAC. This is followed by the access control enforcement with the DEMO methodology. These models are chosen based on our research focus and their features and limitations are explained step-by-step in the following sections.

A. Modeling RBAC in ArchiMate

The ArchiMate language defines three main layers: (1) The Business layer offers products and services to external customers, which are realized in the organization by business processes (performed by business actors or roles). (2) The Application layer supports the business layer with application services which are realized by (software) application components. (3) The Technology layer offers infrastructure services (e.g., processing, storage, and communication services) needed to run applications, realized by computer and communication devices and system software [1].

The scope of this paper remains at the organizational level (i.e., roles, actor, business process, etc.). Hence, we focus on ArchiMate business layer meta model. A description of the main concepts is defined in Table II.

TABLE II. ARCHIMATE BUSINESS LAYER OBTAINED FROM [2], [15]

ArchiMate concept	Definition
Business actor	Individual persons (e.g., customers or employees), but also groups of people (e.g., departments or business units) within the organization.
Business role	A role that an actor fulfills in an organization. Importantly, this role is usually defined as the work carried out by an actor.
Organizational service	A unit of functionality that is meaningful from the point of view of the environment. The following concepts realize a service [2]: *Business processes, business functions, business interactions*. Moreover, a business process/function is "a unit of internal behavior, performed by one or more roles within the organization". Finally, a business interaction is "a unit of behavior similar to a business process or function, but it is performed in a collaboration of two or more roles within the organization".
Business event	A business event is something that happens (externally) and may influence business processes, functions or interactions. A business event is most commonly used to model something that triggers behavior, but other types of events are also conceivable: e.g., an event that interrupts a process.
Business object	An entity that "is manipulated by behavior such as business processes or functions".

Figure 2 describes the mapping between the ArchiMate business layer model and the RBAC model. Note that we use only an excerpt of RBAC and ArchiMate to focus on concepts relevant for the scope of this paper. Moreover, we define a permission (P) as an approval of executing one-to-many operation(s) on one-to-many resource(s) for a sake of simplicity (i.e. permissions granularity's issues). Concepts mapping has been facilitated by the application of mapping techniques [16], and explained as follows:

- In RBAC, we define a *User* as a specialization of a *Business actor*.
- In RBAC, we define an organizational *Role* as a specialization of a *Business role*.
- In RBAC, a *Permission* depicts a *Business behavior* in ArchiMate once an *event* is triggered. This defines a *Resource* as a specialization of a *Business object*, and an *Operation* as a specialization of an *event*.

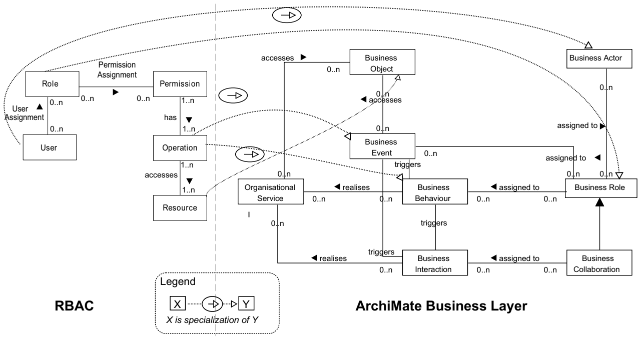

Fig. 2. Mapping of ArchiMate business layer with RBAC concepts (Adapted from [5])

B. Modeling RBAC in DEMO

Design and Engineering Methodology for Organizations [11], for short DEMO, offers a theoretical perspective for thinking and design associated way of working, along with a methodology composed of seven steps. It introduces capabilities to deal rigorously with the dynamic aspects of the process-based business transactions using an essential ontology that is compatible with the communication and production, acts and facts that occur in reality between actors in the different layers of the organization. A DEMO business transaction model encompasses two distinct worlds: *(i)* the transition space and *(ii)* the state space.

The DEMO transition space is grounded in the Ψ-theory whereas the basic transaction pattern includes two distinct actor roles: the Customer and the Producer. The goal of performing such a transaction pattern is to obtain a new fact. A business transaction is performed by a sequence of coordination and production acts that leads to the production of the new fact encompassing: *(i)* the order phase that involves the acts of request, promise, decline and quit, *(ii)* the execution phase that includes the production act of the new fact itself and *(iii)* the result phase that includes the acts of state, reject, stop and accept. Firstly, when a Customer desires a new product, he requests it. After the request for the production, a promise to produce the production is delivered by the Producer. Then, after the production, the Producer states that the production is available. Finally, the Customer accepts the new fact produced. The DEMO basic transaction pattern aims at specifying the transition space of a system that is given by the set of allowable sequences of transitions. Every state transition is exclusively dependent from the current state of all surrounding transactions. There is no memory of previous states.

The DEMO state space delivers the model for the business transactions facts, which are products or services, and that are obtained by the business transaction successful execution. Throughout the business transaction execution more intermediate facts are required.

At some level, DEMO controls the business transactions [11] because it guarantees that the transactions are formed accordingly with an essential ontology that is compatible with the communication and production, of the acts and facts, which occurs at operation time between the organizational actors. However, because the actors are autonomous to take alternative actions throughout time, sometimes named as workarounds, they do not strictly follow the business transactions models [17]. Hence, the organization needs to enforce dynamic control mechanisms to react whenever misalignment between the business transaction models and the actors operation occur. The immediate reaction is typically to revoke the actors accesses, to bound the impact of their actions. Later, after evaluation, the misalignment may be considered innovative and added as new business transaction models. In the immediate reaction scope, access control is the solution that offer the possibility of applying a given access policy to the operation.

The following concerns are pointed by RBAC [18] to access control enforcement: *(1)* the capability to define and then to dynamically manage the access configurations and *(2)* the capability to run-time control the actor accesses to the different artifacts. In Figure 3 these concerns are modeled using three DEMO transactions: T01, T02 and T03, with their correspondingly initiator (CA01 User) and executor actors (A01 Access controller and A02 Access provisioner).

Regarding the artifacts that are access controlled, more

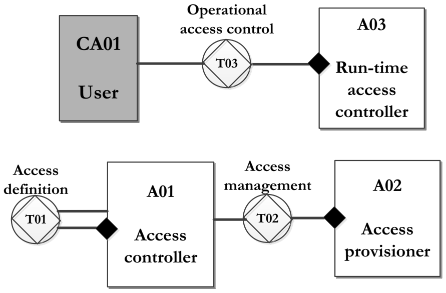

Fig. 3. Access control designed by DEMO business transactions

detail is needed to understand what is inside the DEMO business transactions. In the proposal of [19] access control is embedded in the actors operations, integrating the RBAC body of knowledge with the DEMO business transaction concepts. As stated, a DEMO business transaction represents the elementary organizational building block and is supported in [20] by the Ψ-theory that provides insight about the coordination and production activities occurring in a universal pattern [11]. Moreover, [21] details that a business transaction is a model representation of a given organizational reality that is valid within a specific time-frame, and that should include who is responsible for each part of the business transaction and the comprehensive definition of systems state and transition. Following the general system theory [22] three fundamental spaces are considered for any system:

- the state space, representing the set of allowable states of a system;
- the transition space, representing the set of allowable sequences of transitions of a system;
- the actor role space, representing the set of allowable competences, authorities and delegations of a system.

Accordingly with these three distinct spaces, Fig. 4 proposes an enforcement of RBAC in DEMO business transactions using a formal ontology specification (WOSL). Each business transaction is enforced by its state (*DEMO: Fact type*), its transition (*DEMO: Action Rule*) and its actor role (*DEMO: Elementary Actor Role*).

The *DEMO: Fact type* and *DEMO: Action Rule* are related with the RBAC *Permission*. On one hand, a *DEMO: Fact type* is a systems entity that is created, removed or updated in some way during the business transactions operation, *e.g.*, create a receipt information. By the other hand, a *DEMO: Action rule* is an entity that executes an atomic part of the business transactions operation, *e.g.*, requesting a product. Usually, the RBAC *Permission* is related with a systems function, but in the scope of DEMO business transactions this solution delivers a much finer grained capability of controlling access to any state or any transition element. To guarantee the consistency, a mutual exclusive law restriction imposes that each *Permission* cannot be used at the same time for a *DEMO: Fact type* and a *DEMO: Action Rule*. However, in practice, if the business transactions are properly decomposed to only handle a fact type inside one action rule then it is enough to only integrate the *Permission* with *DEMO: Action rule*.

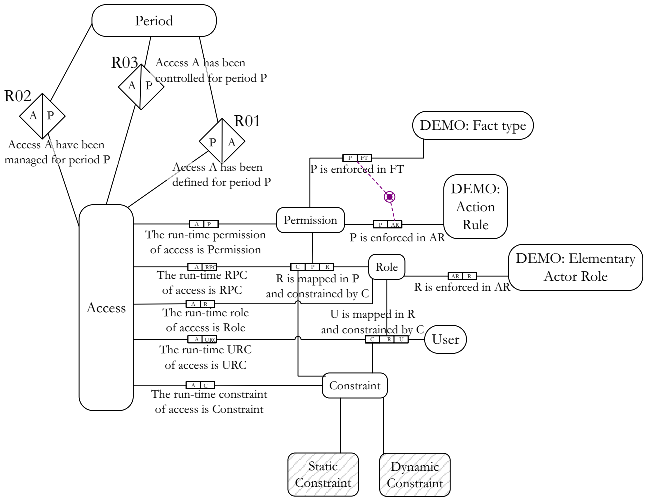

Fig. 4. WOSL for RBAC representation using DEMO (Adapted from [19])

The *Access* follows the RBAC model encompassing *User*, *Role*, *Permission* and *Constraint*. Each *User* is mapped with *Role*, and each *Role* is mapped with *Permission*. A *Constraint* is specialized by the categories *Dynamic Constraint* and *Static Constraint* representing the design restrictions that allow configurable access authorization, *e.g.*, separation of duties.

To summarize, *Access* encloses the desired access configuration for a given *Period* of time. Meaning that *Access* is valid only within a time *Period*, and that it demands a previous definition of all the previous concepts. Relating Fig. 3 and 4, *R01* represents the result of executing the transaction *T01*, as such *R02* occurs when *T02* is executed. *R03* represents the finished execution of *T03*, where the access is granted or revoked depending on the previous provisioned access configuration.

This access control solution is thus able to accommodate dynamic change throughout time imposed by the organization access managers and controllers [23]. The access configuration is not a static definition and enforcement, but rather it evolves along with the organizational needs. The core concern is to be able to respond to the initial requirement of an immediate reaction when a misalignment is found at run-time.

C. Synthesis

We have conducted two different experiments of modeling access control in EA. The first experiment has shown that ArchiMate can be enriched with access control concepts by means of mapping techniques with RBAC. During design time, modeling access control in ArchiMate is a good solution to have an overall picture of actors, roles, business, and their underlying IT infrastructures.

The second experiment has introduced a different modeling of access control in EA. Using DEMO, we also identified similar concepts to RBAC. Moreover, we stressed the dynamic behavior of access control using DEMO's transaction concept. During run-time, we are able to manage and provision access

TABLE III. RBAC - DEMO META MODEL CONCEPTS MAPPING

RBAC *vs.* DEMO	**Mapping rationale for concepts**
Permission *vs.* State space	The permission to the set of allowable states of a system. For instance, creating, removing, updating or reading is assigned to any artifact belonging to business transactions definition.
Permission *vs.* Transition space	The permission to the set of allowable sequences of transitions of a system. For instance, executing an atomic part of the business transactions operation: requesting a product.
Role *vs.* Elementary actor role	A role in RBAC corresponds to a DEMO elementary actor role.
Permission *vs.* transition and state space	For each permission, a mutual exclusive law exists between a *Permission* for state space and *Permission* for transition space.
User Assignment (UA) *vs.* Elementary Actor role	UA has no mapping in DEMO.

control for a requester. Nevertheless, the management of parallel access control enforcement within different units of an organization will lead to a DEMO spaghetti model. In this paper, our goal is to create a consistent and lightweight model for access control in EA. The DEMO transaction concept would then play a role of both modeling and monitoring access control enforcement at the organizational level of ArchiMate in conjunction with RBAC.

IV. ACCESS CONTROL IN BUSINESS TRANSACTIONS

This section proposes the enforcement of the access control models concepts in the Enterprise Architecture models. The genesis of this enforcement derives from the synthesis of the two experiments introduced in the previous section and in the belief that security should be a concern to be directly enforced in EA. We concentrate on resolving the semantic heterogeneity through concept mapping and integration rules as presented in [24]. This step defines the conceptual construction of the design-science research methodology.

A. Mapping RBAC to DEMO

Table III proposes the mapping approach between RBAC and DEMO business transactions as defined in Sect. III-B. It follows the rationale of (1) using a fine-grained conceptual definition for a business transaction (encompassing both the state space and the transition space) to clearly define the RBAC Permissions and (2) using the actor role modeling to administrate RBAC roles. In this way, the RBAC limitations of coarse/fine-grained permission definition and high effort for provisioning are overcome. The idea, is that the knowledge contained in the DEMO business transaction models is used for the benefit of RBAC model definition. By grouping these two models, we achieve the best qualities of each one: business focus and security focus.

B. Mapping DEMO business transactions to ArchiMate

We present the mapping between the concepts of the DEMO business transactions and the ArchiMate business layer. Table IV follows the rationale of the mapping results presented in [25].

TABLE IV. DEMO - ARCHIMATE META MODEL CONCEPTS MAPPING

DEMO *vs.* ArchiMate	**Mapping rationale for concepts**
Actor *vs.* Business role	An actor in DEMO refers to a social role played by a subject in an organization. Such a social role corresponds to the definition of a business role in ArchiMate where roles are typically used to distinguish responsibilities.
Transition space *vs.* Business behavior or event	It defines an act performed by a subject member of a social role. Its scope is about contribution/coordination for services. In the ArchiMate context, it corresponds to the realization of an organizational service via a business process or a function (business behavior) or a business event (e.g. external request).
Transaction *vs.* Business interaction	In DEMO, transactions are always initiated and executed by different roles. This emphasizes the interaction aspect that we can find in ArchiMate, where a business interaction requires more than one role to perform an organizational service.
State space *vs.* Business object	It is the result of an act. In ArchiMate, it represents an element accessing a business object and encapsulated within an organizational service.

TABLE V. OVERALL RBAC AND DEMO AND ARCHIMATE CONCEPTUAL MAPPING

RBAC	**DEMO**	**ArchiMate**
User (U)		Business actor
User assignment (UA)		
Role (R)	Elementary actor role	Business role
Role hierarchy (RH)		
Permission (P)	State space	Business object
Permission (P)	Transition space	Business behavior or event
Permission assignment (PA)	State and transition space	
Session (S)		
	Business transaction	Business interaction
SoD constraint		

C. Synthesizing all concepts: RBAC, DEMO and ArchiMate

The idea is to use the strong points of each set of concepts in order to model a full access control business transactions in EA. We propose in Table V an overall conceptual mapping between the three previous set of concepts: RBAC, DEMO, and ArchiMate. By one hand, regarding conceptual gaps, the run-time operation concerns, such as the specific concepts of *User*, *Session*, *User assignment*, *SoD constraint* and *Role hierarchy* are not fully supported by DEMO and/or ArchiMate. Therefore, if we pursue the endeavor for controlling the dynamic aspects that occur at operation-time then these concepts should be embedded in EA models since design-time. By the other hand, the RBAC concepts, such as the *Role*, and the *Permission* reveal that a fine-grained enforcement is demanded at the DEMO business transactions level. Currently, DEMO business transactions encompass both the coordination (communication, management) and the production worlds (executing or producing the goods or services) but do not include security enforcement. Conversely, combining the DEMO business transaction spaces (namely the state and transition) with the *Permission* RBAC concept allows a directly, and detailed, conceptual integration between them. Furthermore, extending this integration to ArchiMate business level will offer a lightweight security pattern enforcement at EA modeling landscapes; thereby depicting in one model organization's actors, business processes, applications, and IT infrastructure (see Fig. 5).

The evaluation of the overall mapping is done based on the observational case study and an experimental design as described in [12]. The evaluation part is illustrated using an e-Government case study and then tested with the specified security enforcements in Sect. V.

V. Illustrative Example

In this section, we present the *Evaluation process* of the design-science research methodology. We instantiate our approach to model a fine-grained access control in a real world scenario from an e-Government case study [26]. Mutual Legal Assistance (MLA) defines a collaborative scenario involving national authorities of two European countries, named Eurojust, regarding the execution of measures for protection of a witness in a criminal proceeding. The description of the case study is organized in two parts. In the first part, we model MLA from an EA perspective using ArchiMate. The second part concerns the security requirements to be integrated within ArchiMate where RBAC and DEMO specifications will operate.

A. MLA architecture

Here we describe the MLA process cross Eurojust organizations A and B. At the business level, we define the main actors: Prosecutor A and Judicial Authority Officer (JAO) B. The work consists of granting access to an external role Prosecutor when issuing an MLA request (see the business interaction 'Send MLA Request' on the top of Fig. 5). The analyze of the request is done by the actor JAO B who will give access to the specified files of the business object 'MLA documents' (see the internal business process 'Process MLA' in Fig. 5). The reason of the business interaction is that the organizational service MLA requires two roles (Prosecutor and JAO) to be executed.

At the application level, Eurojust integrates services such as MLA service and CMS (Case Management Service) to process data on the individual cases on which Eurojust national members are working in temporary work files (see application services, components and data objects in Fig. 5).

The value of the MLA business service relies upon the security of the exchanged documents between the Prosecutor and the Judicial Authority Officer (i.e. Criminal Records file in Fig. 5).

B. Access control transaction

Here we explain how ArchiMate concepts interact with RBAC and DEMO mappings to manage access control. The goal is to define a lightweight model supporting access control in ArchiMate based on our mapping efforts in Sect. IV. A lightweight model will avoid yet another extension of ArchiMate that may turn to a *spaghetti model*.

We leverage specific ArchiMate concepts to illustrate how to model access control within EA models, and then extend the security in business transactions operations. We remark that the access control enforcement presented in the MLA example is a summarized instantiation of the previous conceptual mapping from Sect. IV. To that end, we present how ArchiMate's roles are assigned using RBAC and then ArchiMate's interactions are enforced based on DEMO transactions.

The first consideration is that the organizational service MLA in ArchiMate requires two roles (Prosecutor and JAO) to be assigned to two different actors (RBAC's users) with different permissions. The second consideration recalls that ArchiMate business interactions should follow the DEMO business transaction semantics in order to comply to the specification of all the communication and coordination transition states. In this MLA example, the 'Send MLA Request' exemplifies this recommendation.

Besides DEMO semantics, fine-grained access control should be explicitly enforced. To this end, whenever a business interaction needs access to a given state, *e.g.*, a 'MLA document', then a dedicated access is explicit designed through the Judicial Authority Officer (JAO) role. JAO has the unique competence to grant or revoke the fine-grained access to specific business interactions or state. This recommendation is explicitly added to the model in order to guarantee its inclusion on the information systems development. Furthermore, all the ArchiMate artifacts that are eligible to be under augmented security should follow this recommendation.

C. Discussions

The benefits of designing a fine-grained access control within an EA model are manifold: more confidence of the end users to use the correspondingly systems; strict competences fulfillment to any organizational actor without overlapping responsibilities; and the capability to audit who has been involved in the business transactions operations in order to adhere to external regulations.

The main requirements for designing EA with access control are: *(i)* a full compliance between DEMO and ArchiMate is demanded from the Enterprise Architect ([25] presents a metamodel integration between DEMO and ArchiMate that could facilitate this effort) and *(ii)* an explicit enforcement design of the critical artifacts access with an actor role to allow the run-time verification compliance.

In a more detailed business case, to guarantee independence, the competence of Chief Security Officer should be separately designed from the business transactions actor roles. For simplification, in this example, it is assigned directly to the JAO role.

VI. Conclusion and Future Work

In this paper, we have proposed an approach to model access control in ArchiMate grounded in the concept of DEMO business transaction. We introduced a formal mapping between RBAC, DEMO and ArchiMate modeling techniques, and showed the reasoning behind the conceptual definition that aids for modeling then enforcing access control in ArchiMate. We proposed a conceptual mapping based on the conjunction of RBAC for the access control definition and DEMO for its enforcement using the business transaction concept to bridge the core concepts from RBAC to ArchiMate. The result was a lightweight model supporting access control in ArchiMate. Moreover, our mapping efforts did not turn ArchiMate into a spaghetti model, quite the opposite, it helped in offering another viewpoint in ArchiMate from an access control perspective.

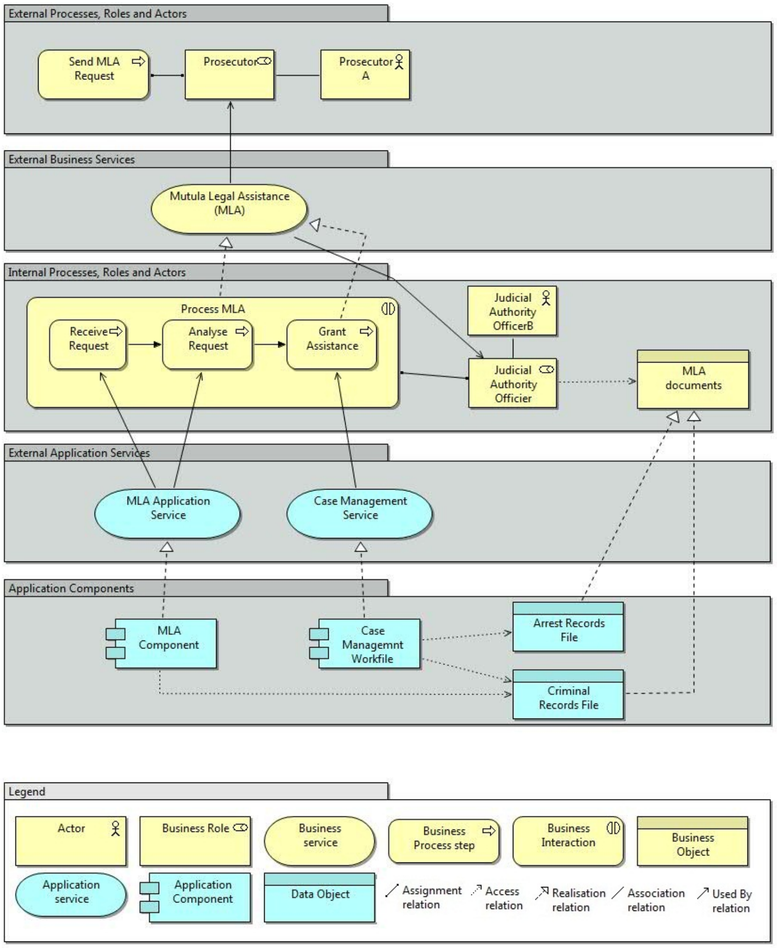

Fig. 5. (Partial) enterprise architecture ArchiMate model using RBAC and DEMO mappings

Besides this achievement, we also discussed the need to enforce a dynamic access control policy. An access configuration is only valid for a given instant in time under specific conditions. However, an organization is a living organism that demand continuous change to react to the imposed exceptions from the surrounding environment. In this sense, access configuration demands complementary capabilities for provisioning and management to deal properly with those exceptions. Therefore, we proposed an approach to enforce the access configuration continuously in the operation of the business transactions accordingly with the emergent needs.

We believe that a step in future research can be represented by adopting this model to the whole EA framework by extending the three-layer in ArchiMate. Also, we are considering the ArchiMate motivation extension, where concepts such as drivers and goals could reflect security guidelines in EA frameworks.

References

[1] M. M. Lankhorst *et al.*, *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, Berlin, Germany, 2005.

[2] M.-E. Iacob, H. Jonkers, M. M. Lankhorst, H. A. Proper, and D. Quartel, *ArchiMate 2.0 Specification*. The Open Group, 2012.

[3] C. Feltus, E. Dubois, H. A. Proper, I. Band, and M. Petit, “Enhancing the Archimate standard with a responsibility modeling language for access rights management,” in *Proceedings of the Fifth International Conference on Security of Information and Networks*. New York, NY, USA: ACM, 2012, pp. 12–19.

[4] E. Grandry, C. Feltus, and E. Dubois, “Conceptual integration of enterprise architecture management and security risk management,” in *The Fifth Workshop on Service oriented Enterprise Architecture for Enterprise Engineering (SoEA4EE 2013), an International Workshop of the 17th IEEE International EDOC Conference*, IEEE, Ed., Vancouver, BC, Canada, 2013.

[5] K. Gaaloul and H. A. Proper, “An access control model for organisational management in enterprise architecture,” in *The 9th International Conference on Semantics, Knowledge & Grids*, IEEE, Ed., Beijing, China, 2013, pp. 135–149.

[6] I. G. Institute, *COBIT 4.1*. ISA, 2007.

[7] ISO/IEC, “ISO/IEC 27002: Information technology Security techniques Code of practice for information security management,” 2005.

[8] J. Zachman, “A framework for information systems architecture,” *IBM Systems Journal*, vol. 26, no. 3, 1987.

[9] G.-J. Ahn and R. S. Sandhu, “The rsl99 language for role-based separation of duty constraints,” in *RBAC ’99: Proceedings of the fourth ACM workshop on Role-based access control*. New York, NY, USA: ACM, 1999, pp. 43–54.

[10] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[11] J. L. Dietz, *Enterprise ontology: theory and methodology*. Springer Verlag, 2006.

[12] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Q.*, vol. 28, no. 1, pp. 75–105, Mar. 2004.

[13] R. Winter, "Design science research in europe," *European Journal of Information Systems - EUR J INFOR SYST, vol. 17, no. 5*, pp. 470–475, 2008.

[14] R. A. Botha and J. H. P. Eloff, "Separation of duties for access control enforcement in workflow environments," *IBM Systems Journal*, vol. 40, no. 3, pp. 666–682, 2001.

[15] M. M. Lankhorst, H. A. Proper, and H. Jonkers, "The Architecture of the ArchiMate Language," *Enterprise, Business-Process and Information Systems Modeling*, pp. 367–380, 2009.

[16] N. Noy and M. Musen, "The prompt suite: interactive tools for ontology merging and mapping," *International Journal of Human-Computer Studies*, vol. 59, no. 6, pp. 983–1024, 2003.

[17] S. Guerreiro and J. Tribolet, "Conceptualizing enterprise dynamic systems control for run-time business transactions," in *ECIS 2013 Research in Progress.*, 2013.

[18] N. I. o. S. NIST and T. T. A. U. D. of Commerce, "An Introduction to Computer Security: The NIST Handbook," October 1995.

[19] S. Guerreiro, A. Vasconcelos, and J. M. Tribolet, "Dynamic business transactions control - an ontological example: Organizational access control with demo." in *KEOD*, J. Filipe and J. L. G. Dietz, Eds. SciTePress, 2011, pp. 549–554.

[20] J. Dietz, J. Hoogervorst, A. Albani, D. Aveiro, E. Babkin, J. Barjis, A. Caetano, P. Huysmans, J. Iijima, S. Van Kervel, H. Mulder, M. Op 't Land, H. Proper, J. Sanz, L. Terlouw, J. Tribolet, J. Verelst, and R. Winter, "The discipline of enterprise engineering," *International Journal of Organisational Design and Engineering*, vol. 3, no. 1, pp. 86–114, 2013.

[21] S. Guerreiro, "Business rules elicitation combining markov decision process with demo business transaction space," in *Business Informatics (CBI), 2013 IEEE 15th Conference on*, 2013, pp. 13–20.

[22] L. Bertalanffy, *General Systems Theory*. George Braziller, New Yok, 1969.

[23] S. Guerreiro, A. Vasconcelos, and J. Tribolet, "Enterprise dynamic systems control enforcement of run-time business transactions," in *Advances in Enterprise Engineering VI - Second Enterprise Engineering Working Conference, EEWC 2012, Delft, The Netherlands, May 7-8, 2012. Proceedings*, ser. Lecture Notes in Business Information Processing, A. Albani, D. Aveiro, and J. Barjis, Eds., vol. 110. Springer, 2012, pp. 46–60.

[24] S. Zivkovic, H. Kühn, and D. Karagiannis, "Facilitate modelling using method integration: An approach using mappings and integration rules," in *Proceedings of the Fifteenth European Conference on Information Systems, ECIS 2007, St. Gallen, Switzerland, 2007*. University of St. Gallen, 2007, pp. 2038–2049.

[25] S. Kinderen, K. Gaaloul, and H. A. Proper, "Bridging value modelling to archimate via transaction modelling," *Software & Systems Modeling*, pp. 1–15, 2012.

[26] T. A. R4eGov, "Towards e-administration in the large," March 2006, http://www.r4egov.eu/.