Views for Access Control Management in Enterprise Architecture

Tiago Agostinho
Instituto Superior Técnico,
Universidade de Lisboa
Lisbon, Portugal
tiago.agostinho@tecnico.ulisboa.pt

ABSTRACT

Access control management has an important role in organizations allowing access to information according to the employee's functions. Information is a property of organizations and is their responsibility the management of access controls properly. The goal is to ensure that authorized subjects have the least permissions to carry out the desired operations on the desired objects in the different layers of enterprise architecture of organizations. Access control can decide "who can do what?", protects the objects from unauthorized access, being a fundamental security mechanism in organizations. Different employees in the organization have different roles which require different accesses to properly perform their duties. The policies in access controls can be derived from business rules, laws, or corporate culture. Access controls must address different levels of granularity, be flexible and robust to fulfill the needs of organizations.

Author Keywords

Access Controls; Access Control Models; Security; Role; Views; Query; Organization; Policies; Monitoring; Authorizations; Information Systems; Archimate; RBAC; ABAC

INTRODUCTION

Security has gained special focus, in organizational context nowadays, following the news about cyber security attacks around the world that harm organizations from different sectors and dimensions. Security represents an important concern in organizations, as it is seen as a need for organizations' subsistence.

Access control is an important component to ensure security and privacy requirements in organizations, and prevent unauthorized access. The information and knowledge that reside in organizations (e.g. stored in Information Systems) is an important asset to them, and is their duty to protect this asset against improper use and access.

Access control, which can be physical, technical, or administrative, is a mechanism to ensure information security[8]. Access control models ensure access to specific information throughout the organizations, defining and restricting which subjects can perform which actions on which objects. The question we want to get the answer is simple, "Who can do what?". The goal of access control systems is to provide means to protect resources from unauthorized access attempts.

Every organization has its own policies derived from business rules, legislation, or organization culture (not necessarily technical decisions).

Policies can be implements at different levels (layers) of the Enterprise Architecture. Policies can vary between different Information systems within the same organization, and may be complementary to each other. The policies objectives can vary from organization to organization, but generally ensure fraud prevention, data protection, separation of duties, and conflict of interest.

The least privileges should be ensured in each access tentative providing the minimum accesses possible to subjects to perform their duties. It is important for an organization to have a wide view of access rights in the different layers of enterprise architecture for all employees. This view upon the access rights sometimes is vendor-dependent and results in the main part of the infrastructure also belonging to the same vendor.

Changes are constant within an organization with employees being hired and leaving from organizations or sometimes changing their areas, and these changes must be reflected in access rights and should be monitored and analyzed by tracking every change in access rights. On the other hand, the access control system should be robust and flexible enough to support the access control fulfilling the organizations' needs. Sometimes the fulfillment of organization needs requires different levels of granularity, from organizational level down to a specific employee, and this approach is not always present in the products available on the market.

An organization may have different information systems to support its business processes, and an information system may implement access control mechanisms at different points and levels. Sometimes these systems belong to different vendors, and the information architecture varies between vendors. Because business processes may have to cross information systems' implementation, employees may have to interact with different systems during business process execution, and access rights may not be aligned between the different information systems.

Sometimes the information associated with access controls does not reside in a single source and is structured in different ways and layouts. This can be a cumbersome if the information needs to be crossed between different sources to analyze a subject's access within the organization.

Organizations needs different access controls at different

levels of the enterprise architecture, such as at the Business Layer, Application Layer, or Technology Layer, among others. Thus, access control can be applied only not only at the business layer, but throughout the entire organization, crossing the different enterprise layers.

Similar permissions should result in similar rights on different information systems, but sometimes it is not easy to align permissions between different information systems because there is no comprehensive overview of accesses in the organization. Additional effort is required to implement and maintain aligned the same organizational access control policies in different information systems because the access rights architecture of these systems is different. When access control policies are maintained system-by-system, this may lead to a lack of security consistency across systems. It is also difficult to maintain a consistent and holistic view of the global access rights status for all information systems in organization.

Flexibility is needed to maintain different levels of granularity covering different sizes of entities in organizations.

An inadequate ACMS may lead to pressure from security regulators (internal or external from organizations) seeking the best practices for protecting of information assets.

In other words, without a centralized view is not easy to manage access rights, neither to compare them across the organization, nor align them with business requirement.

The heterogeneity above mentioned in terms of access controls leads to an administrative challenge for IT.

This paper proposes the creation of different centralized views in the Access Control Management System (ACMS) to analyze and monitor access controls in organizations with a holistic approach using Atlas[24]. The focus is on protecting resources associated to different enterprise layers of organizations from unauthorized subject access.

BACKGROUND

Archimate[2] [12] is an independent modeling language for Enterprise Architecture Management (EAM) developed by The Open Group. The Archimate language provides the ability to describe, analyze, and visualize the various relationships between different architecture domains of the enterprise environment. It is also possible to visualize the construction and operation of business processes, organizational structures, information flows, IT Systems, and technical and physical infrastructure. Archimate provides a graphical language for representing of Enterprise Architecture over time, making it possible to create a holistic and complete view of the enterprise. The modeling language also provides an approach to describe and analyze different domain architectures and their relationships and dependencies[12].

The three core layer of Archimate used in this paper are the Business Layer which describes products and services in organizations, the Application Layer which supports the Business Layer with Application Services and Application Components, and the Technology Layer which provides infrastructure services.

Archimate also has the aspects defined in *Active Structure Aspect* that represent the structural elements, *Behavior Aspect* that represent the behavior performed by the Actors, and *Passive Structure Aspect* that represent the objects where the behavior is performed.

Archimate provides views which are comprehensible for stakeholders, supports the decision making, and allows to analyze the impacts in the entire organization.

Atlas Atlas[24] is a web-based Enterprise Architecture tool that helps organizations to keep their architectural models updated in a world where organizations are constantly changing. In Atlas, the generated viewpoints are time-dependent and it is possible to analyze the models variance over the time. Using the timeline, it is possible analyze the past enterprise models (AS-WAS models), the enterprise present models (AS-IS models), and the future enterprise models (TO-BE models) visualizing the transformations in enterprises over the time.

Atlas has embedded the by enterprise cartography, which is the process responsible for abstracting, collecting, structuring, and representing architectural artifacts and their relationships analyzing the enterprise reality. Atlas helps to maintain and update the enterprise cartography representation with minimum effort, and detect inconsistencies before they occur.

Atlas allows full definition of meta models, supports custom configured interfaces, and allows configuration of specific forms where the user sees specific properties. Atlas also provides analytical elements such as charts, dashboards and architectural views (blueprints). This tool supports configuration of behavior associated with blueprints using queries and rules.

RELATED WORK

Usually, the access policies that protect resources are categorized as **Discretionary Access Control (DAC)** or **Mandatory Access Control (MAC)**. Different access control models were developed to serve different purposes to control the access to information in organizations. In DAC, [1] subjects, as such object owners, can manipulate the authorizations of other subjects to access objects. In MAC, the policy decisions are determined by a central authority such as system administrator.

Access Control List (ACL) appeared when Lampson [15] proposed a notion of subject and object with a simple type of access control, applied to systems with shared objects, where subject IDs are attached to each object IDs in a matrix. Each entry in the table represents the access rights of a subject on an object for the intersection of column and row. This solution is not scalable because the number of entries increases exponentially with the number of subjects and objects, and the matrix can become sparse because subjects do not require access rights to all objects. To avoid sparse tables, Graham and Denning [9] went further deep and defined access attributes which each subject can perform over each object. Graham and Denning proposed models where is provided a triple (S, A, O) to the monitor of O. The monitor access the matrix to determine the position M[S, O], and if the access attribute is in the matrix position, access is granted, otherwise, access is denied.

Role Based Access Control (RBAC), initially proposed by Ferraiolo, Cugini, and Kuhn [5] was developed to complement DAC and MAC. The administration difficulties in large commercial organizations with DAC are unpredictable because it is difficult to control and manage all access rights that owners give to objects. Other models were very restrictive because of its original military purpose as demonstrated by Clark and Wilson [3]. The concept is simple, establish

permissions to subjects based on their responsibilities and qualifications in organizations[5]. RBAC[6] solved some of the problems presented above by collecting permissions into roles, which typically represent the user permissions in the organizations. The RBAC model is composed by several entities: subjects, roles, permissions, sessions, operations, and objects. In RBAC, the permission to perform operations are not assigned individually to users; instead, operations are associated with roles. Each role defines the permissions for each subject responsibility, and the assignment of each individual permission to a role reduces the effort and complexity to maintain the access control system because access rights are not maintained for each individual subject. A subject may be assigned to one or more roles, and a role may be assigned to one or more permission in a many-to-many relationship. The subject once assigned to a role gains the access rights for the permissions assigned to the role. The concept definition of roles simplifies the access control management and reduces the cost and potential errors during the subjects' permission assignment. RBAC requires an additional configuration effort that can be challenging and time consuming for many organizations. Permissions are assigned to roles and are indirectly are associated to subjects.

RBAC is widely implemented in organizations of different sectors such as commerce, health, or government. The main weakness of RBAC is the huge initial time-consuming to configure the roles structure. The complexity of RBAC can increase with the complexity of organizations and in some cases, the number of roles can be higher than the number of users. The number of roles may increase in an organization with the number of shared functions between departments. Hence, in order to preserve the principle of least privileges, a new role should be created and assigned to him, with the permissions of shared activities. This type of situation may happen when the organizations are not well structured and have no functions well defined or when organizations have shared services.

The lacks of RBAC were the trigger for the appearance of **Attribute Based Access Control (ABAC)** and its standardization[10] was an important step for his acceptance in the community after several proposals of the model in the literature. RBAC brought a simple, centralized, and easy method to manage access rights, but presented noted limitations, mainly when applied to internet and distributed systems. RBAC is often cumbersome to set up and manage, and the users' role sometimes is not easy to be express in access control policies. ABAC provides flexibility and there is no need to express individually the relationship between subjects and objects. Sometimes organizations have complex structures and policies, but ABAC provides a fine-grain approach to cover these situations, while in RBAC this is only possible with a high number of permissions and roles, becoming the solution not scalable[4]. ABAC can provide dynamic access control decisions involving environment attributes in the decision making [23]. Xin, Krishnan and Sandhu [25] demonstrated with ABACα the flexibility and custimization of ABAC to be implemented in the previous models well accepted such as MAC, DAC, and RBAC. The essence of ABAC is based on the evaluation of attributes assigned to subjects and objects, requested actions, and environment conditions used in policies to grant or deny access rights. The history of accesses attempts (permit or denied) and the changes in policies and attributes can be stored using blockchain for audit purposes[23]. ABAC goes beyond the traditional access control models because provides flexible frameworks, allowing his enforcement in a distributed and interconnected enterprise world. The policies govern access rights based on attributes regardless of the number of users and objects. ABAC is a way to grant or deny the access of subjects to objects based on attributes, used in policies definition. Attributes are assigned to subjects, objects, and environment throughout relationships. Environment is a new concept, not considered in RBAC, that represents the environment conditions under the subject attempts to grant the access over the object. Access rights can change with the changes in attributes, keeping the policies and ensuring a dynamic access control system. When a new subject is created in the system, it is only necessary to maintain its attributes, without any other special assignment or configuration in terms of access control. **The access rights in ABAC may change with policy changes or with attribute changes.** The policy rules in the systems are defined using policy languages that supports attributes, such as XACML (XML-based) [18]. The access control policies in ABAC are only dependent from the attributes and ABAC frameworks to limit the access of subject to object, unlike in RBAC where the policies are defined with the manual user-role assignment. The policies may be expressed using logical operators (e.g., AND, OR, =), to compare attribute values, returning a Boolean statement (granted or denied) (e.g. *"object.author == subject.id"* or *"TIME* $\geq$ *8AM"*). Policies defined using logical operators can be quite rich and complex. Hence, with this flexibility, is not necessary to specify individual relationships between each subject and each object neither additional management when subject and objects are created or deleted. When new subjects join the organization, policies and objects do not need to be modified. **eXtensible Access Control Markup Language (XACML)**, which provides a standard framework for ABAC implementation/deployment, is an eXtensible Markup Language (XML) designed to express security policies[18], as well as, access requests and responses needed to interact with ACMS, reaching an authorization decision. Each XACML request is composed by subject(user), action(operation), object and environment key-value pairs. 1 shows a simplified XACML access scenario where a subject requests access to an object using the access mechanism. This mechanism uses policies, subject attributes, and object attributes to determine and enforce the allowed subject operations on the object. The policies are enforced by the mechanism [10], which is also responsible for collecting information about subjects, objects, and environment to make the decision. Each policy is evaluated and rendered in the Policy Decision Point (PDP), and the decision is enforced in the Policy Enforcement Point (PEP).

Kuhn, Coyne, and Weil[14] mentioned that the merge between the best features of RBAC and ABAC can provide an effective access control for distributed and rapidly changing systems. This solution was designed to overcome the drawbacks of RBAC, such as time consumption for initial setup of roles, or "roles explosion" to fulfill certain requirements of organizations. Kern and Walhorn [13] proposed a model to dynamically assign users to roles using rules based on user's

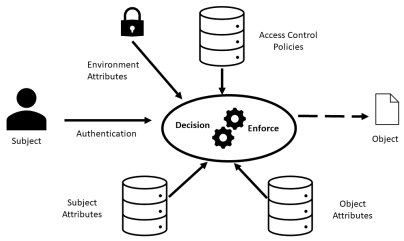

Figure 1. Basic ABAC-XACML Scenario

attributes. Kern and Walhorn argue that attributes can be used to automate the process of roles assignment. Kuhn, Coyne, and Weil[14] proposed a combination of strategies to take advantages from the strenghts of RBAC and ABAC. The strategies to integrate RBAC and ABAC, RBAC-A (RBAC with attributtes), are present in 1. ABAC facilitates the specification of access rules, but to determine/analyze the permissions for a user, the rules should be executed in the same order in which the ACMS applies them. The options 7, 8, and 9 represent a hybrid RBAC/ABAC designs composed by user ID, roles, and attributes. Options 7,8, and 9 are different possibilities where RBAC and ABAC can be combined. Some of the following models combine features of ABAC and RBAC models into a hybrid model.

Security and privacy are a non-functional requirement that affects business processes and IT systems[20]. These security requirements may be imposed by law, corporate risk management, or customers. Enterprise Architecture can provide a holistic view about the current state of the Enterprise (AS-IS), helps to define the different possible states in the future (TO-BE), allowing the relationship analysis between the different enterprise layers. Gaaloul, Guerreiro, and Proper [7] experimented the approach of access control management in Enterprise Architecture. They proposed an access control in Enterprise Architecture using the RBAC model applied to Archimate[2], establishing a correspondence between RBAC and Archimate entities.

PROPOSED SOLUTION

The objective of this paper is to propose a practical solution for access control in the enterprise environment using the enterprise architecture framework Archimate[12], ABAC model, and the ATLAS [24] platform. In this paper, we propose to the merge the entities of Core Archimate framework [2] and the ABAC access control model to define policies used to restrict the access from Subjects to Objects throughout the organization, particularly in Information Systems. The question that we intend to answer is **"Who can do what and in which circumstances?"**. The term "circumstances" adds to the equation the environment variable that can restrict access for subjects to objects in terms of location or time.

Solution Synthesis

The solution consists of a presentation layer for creating, updating and analyzing access control rights based on entities and their attributes. The enterprise architecture is based on graphical models that provide a holistic view that helps analyze and design access policies needed in organizations. The entire ACMS, since policies administration to entities and attributes administration is managed by an access control administrator.

The solution proposed here is only applicable to the access control inside an organization, as access control beyond the boundaries of an organization is out of scope of this paper. The solution only considers the subjects' authorization, assuming that each subject is already authenticated by the system (if applicable). Access control can be implemented in many places inside the organizations, but our focus in this paper is on access control of Information Systems, business processes, and IT infrastructure inside organizations.

The entities' representation of organization follows the three Archimate[12] Core Layers (Business Layer, Application Layer, and Infrastructure Layer). The presented solution proposes to fill the security requirements using Archimate[2] and the fine-grain and flexible policy approach provided in ABAC access control.

To face the problems previously presented, we propose to completely externalize the access control management from each information system, creating an access control orchestration, called ACMS (Access Control Management System), inside the organizations. Each access can be checked in ACMS. If the access is reflected in ACMS views, access must be **Granted**, otherwise access must be **Denied**.

The approach presented in this paper allows the same access control policies to be reused across different information systems (with respect to the application layer), thereby establishing consistency in access control policies, improving efficiency, and reducing the time required to maintain access control policies. Alignment of policies between different information systems reduces security breaches in organizations. This paper aims to increase operational efficiency and review in access control management by access control administrator, centralizing the access rights in a simple point of knowledge, with the same access information architecture shared by the entire organization.

Solution Structure

The solution presented here is a presentation layer that covers all the entire process of access control management in an organization. The solution covers access controls in the three core layers of Archimate[2]. The solution structure is divided into the following three parts:

- Creation and maintenance of subjects, objects, actions, environments, and their respective attributes. Creation and maintenance can be made individually or in a massive way. In the case of action and environment is not expected a large number of instances (comparing with the number of subjects and objects) and in this way the entities' maintenance can be performed manually. The mass update will be performed using the upload tools available in Atlas. The attributes associated with subjects, objects, actions, and environment may vary between organizations.

- Definition and updating of policies using entities and attributes. The access control graphical queries will reflect the organizational policies, affecting the core Archimate Layers. Policies will be built using the attributes of objects, subjects, actions, and environments. Policies definition will

Table 1. Combination of strategies to integrate RBAC and attributes

Option	User ID	Role	Attribute	Model	Permission Mapping
0	0	0	0	undefined	-
1	0	0	1	ABAC-basic	A_1,..., $A_n \longrightarrow$ perm
2	0	1	0	undefined	-
3	0	1	1	ABAC-RBAC hybrid	A_1,..., R, $A_n \longrightarrow$ perm
4	1	0	0	ACLs	$U \longrightarrow$ perm
5	1	0	1	ABAC-ID	U, A_1,..., $A_n \longrightarrow$ perm
6	1	1	0	RBAC-basic	$U \longrightarrow R \longrightarrow$ perm
7	1	1	1	RBAC-A, dynamic roles	U, A_1,..., $A_n \longrightarrow$ R $\longrightarrow$ perm
8	1	1	1	RBAC-A, attribute-centric	U, R, A_1,..., $A_n \longrightarrow$ perm
9	1	1	1	RBAC-A, role-centric	$U \longrightarrow R \longrightarrow A_1$,..., $A_n \longrightarrow$ perm

be made using the graphical query editor available in Atlas. Graphical query editor also provides the ability to import and reuse previous defined policies. Our solution also contemplates complex queries which are the result of individual policies aggregated.

- Report and analysis of the current status of access rights. Report and analysis will be supported by views based on policy queries previously generated. The different formats available in Atlas that will be used in this paper are **ACL view, and Board View**.

Access Control Elements

The first steps to design an access control system is to define the main elements and how they interact with each other. Each authorization element has its specific properties and roles during the access control management in organizations. The elements that this solution proposes are presented below:

- Subject – Subjects set represents all active entities in the organizations that may require access to an object (passive entity). Typically, the active entities are those specified in the Archimate core framework. A subject has at least one mandatory attribute that is used for unique identification.
- Object – Objects set represents all passive entities that are protected by access rights and may be accessed by subjects (e.g. files or documents). Typically, the passive entities are those defined in the Archimate core framework. An object has a mandatory unique identifier.
- Attribute – Attributes are used to characterize subjects, objects, actions, and the environment. Attributes of entities are used to specify the policies. Attributes are linked to Subjects, Objects, Actions, and Environment through a relationship. Attributes can be a set of values (eg. roles) or a single value (eg. id). Attributes may be a value or a reference to an Archimate entity. Subject attributes can be compared to object attributes (e.g. subject.id == object.owner) or compared against constants (e.g. subject.department == "IT") during query policies definition. The entities' attributes can be collected from different sources in organizations.
- Action - Describes what a Subject wants to do over an Object. Action may implies only access to the object (e.g. read) or change the object status (e.g. update or delete). The board views display the actions through arrows from Subjects (active entity) to Objects (passive entity).
- Environment – Specifies the environment in which access control is requested (e.g. hour, time, location). This element also plays an important role to grant or deny the access of Subjects to Objects. Environment conditions must be specified during policies' definition and are then implemented in graphical queries and reflected in the views.

Policies

Policies are defined by laws, business or organizational culture present in each organization. They are then translated into queries and implemented in the access control management system. A policy is a query used to define the subject's accesses using attributes (from subjects, objects, actions, and environment), and other policies.

The policies that use logic-based formulas in an organization can be quite complex. When we have multiple queries to define a policy, the order in which the queries are performed may have an impact on the final result.

Policies will be developed in Atlas[24], an enterprise architecture tool, using a graphical query tool design, and a blueprint designer, simplifying the work of the access control administrator.

Atlas also allows the generation of queries based on XML and its translation between graphical and programming approach is also possible. Atlas provides the possibility to reuse policy queries in other queries. Policies will be created using attributes of subjects, objects, and environment.

Policies are built identifying the subject, object, action, and environment (if applicable) in a written sentence and then translating it into a query. Taking as example the sentence *"Only doctors can change patient medical records"*, the subject is *"doctors"*, the action is *"change"*, and the object is *"medical records"*. This means that access will be granted for doctor's requester and denied for the remain requestors. Action will be reflected in the views if the subject has access to the object. If there is no relationship in views between subjects and objects, it means that the subjects have no permissions to perform any actions with the object.

The views may change with the attribute values changes or policy changes.

Graphical Queries

Policies, independently from which source, are written in natural language. Then, in these sentences are clearly identified the four elements of ABAC access control (Subject, Object, Action, and Environment) which will used to built the graphical query.

The result of the query will be the objects where subjects can perform some kind of action in a particular context.

Taking as example the sentence "Thomas can read purchase orders of its working site", "Thomas" we consider as a unique subject identifier, "read" is the action, "purchase orders" is the object, while "its working site" is the environment. In this case, environment provides a dynamic approach to the policy, because when Thomas changes the working site, he will see the purchase orders of that new site without the policy changing, only the subject attributes.

Graphical queries can be built based on the attributes of each entity or based on the relationships between entities, or both.

Organizational Entities

This paper proposes treating the role as an entity, as described in Archimate, which describes the functions performed by business actors and where they can be assigned.

In this paper, the active entities in Archimate represent the Subject in ABAC, while the passive entities in Archimate represent the Object in ABAC. Action and Environment are concepts without conversion between ABAC and Archimate in this paper, but entities in Atlas will be create to represent them. All the remaining entities in Core Archimate Framework may be used to define policies in Atlas.

Entities Maintenance

Atlas[24] allows entities to be created and updated with the help of upload tool or individually using a data explorer where entity attributes can be updated. The upload tool offers the possibility of mass update, which is desirable in large organizations and makes the solution scalable.

The upload tool performs a mass upload of a file with a specific structure for each type of entity in Atlas. The upload tool will be used for demonstration purposes to simulate the integration between ACMS and all Information Systems in the organization. In a real scenario, the upload tool is replaced by webservices in integrations. Policies will be managed exclusively in Atlas.

Views Generation

Reporting and auditing are essential controls in access management and systems security. Once policies are established, a practical and simple way to analyze accesses to verify that they comply with organizational requirements is needed. This solution proposes views generated from query policies to analyze access rights from different perspectives. The type of views proposed using core Archimate framework are presented below:

- ACL View – This view intends to display the relationship between Business Actors and Application Components in Atlas. Business Actors represent the rows, while Applications Components represent the columns. Clicking on each table cell, will be possible to analyze the Data Objects where the Business Actors can perform the actions.

- Board View – This type of view will display the objects that a subject has access rights to in a board. It will be possible to apply filters at the subject, and object levels to get a clean view of what the system administrator wants to analyze.

DEMONSTRATION

We will demonstrate our solution using a fictitious case study related to access control in a Hospital environment which crosses the different Archimate[2] Enterprise Layers.

The goal of this demonstration is to understand the practical point of view of this solution and how it can add value to organizations in terms of access controls. We will also demonstrate the flexibility and adaptability of our solution to control the access in organizations. In this cases study, we will start by defining the policies of each organization and the entities of each layer in the Archimate [2] core framework. Then ,the policies are translated into graphical queries that are applied to the different views. These queries are the base for the views where is possible to check which objects are accessed by each subject. For the case study, two views are presented which allow us to analyze the access from different points of view.

The access control views presented for the case study is *Access Control Big Picture*, where it is possible to analyze the Business Actors accesses in different Archimate core layers of Enterprise Architecture, and *Business Actors VS Applications*, where it is possible to analyze the list of Data Objects that Business actors can access in different Application Components.

Access Control Big Picture view shows the accesses linking the active entities and passive entities through an arrow. The source of the arrow is the active entity, while the target of the arrow is the passive entity accessed by the active entity. **Accesses determination may cross the boundaries of Archimate Core Layers.**

Business Actors VS Applications view shows the accesses in an ACL matrix, where is possible to verify the Data Objects that Business Actors can access by Application Components. The rows of the view are Business Actors, while the columns are Applications Components. When the Business Actors have access to Data Objects, the cells in Atlas ACL Matrix appears with different colors (non-blank color) and clicking over the cell will be raised a pop-up with a list of Data Objects that Business Actor can access in a specific Application Component. The policies proposed in this section are written in natural language and then translated into a graphical query in Atlas. In graphical queries, it is also possible to analyze the flow and the relationship of entities used in the query.

Policies may have different levels of granularity being applicable since the entire organization to a specific employee in the organization. Policies can also be defined to affect a set of employees with the same attributes (e.g. same role, same department, same location, etc, ...). The proposed solution also supports the definition of policies with RBAC approach in Archimate (policies based on Business Roles in Archimate). Queries with different levels of granularity will be defined for each case study to demonstrate the robustness and flexibility of our solution.

Hospital Case Study

The case study is related to a Hospital organization. Hospitals leads with medical records that are considered very sensitive and valuable data, and in this way, considering the risk management associated, they must have a very restricted access control policies.

Since we are leading a non-real case scenario, the information systems here presented were gathered from different

documentation sources related with security issues in Hospital Information Systems [17][16]. The view does not simulate all accesses in Business Processes related to a Hospital environment, just focuses on core Business Processes and some support Business Processes.

In the hospital environment (specially in Information Systems), protecting the confidentiality of health information, while ensuring authorized physicians can access it conveniently, is a crucial requirement. Patient data security has high importance for hospital's reputation. In addition to medical records, hospitals also lead with financial and administrative data (among other sensitive data) which requires a different access controls. Policies belong to the hospital and may cross the applications scope, being also applied to the different Archimate Layers, including locations (e.g. physical access to specific hospital rooms). The access control mechanism should be able to dynamically grant permission to a physician to access any data related to healthcare activity. The following lines describe the main policies that we will implement in the hospital case study.

- Staff assigned to Radiology Rooms can create Radiology Documents.
- Staff working in Administrative Office can access the Billing System.
- Staff working in Administrative Office can change all documents in Financial IS
- Doctors can read only the medical records (e.g. X-ray Image, Blood Test Result) for patients assigned to them.
- Doctors can create discharge document for patients assigned to them.
- Nurses can only read the medical records of patients assigned to the same Location of them.
- Pharmacists can read Prescriptions that appear in Pharmacy IS.
- Visitors can access Reception, Waiting Rooms, Pharmacies, and Patient Wards locations.
- James (Doctor) has access to the ICU Rooms.
- Edward (Pharmacist) can access the Pharmacy IS.
- Mark (Laboratory Staff) can create Laboratory Results Records.
- Laboratory Results can be shared by Laboratory Staff.
- Insurance subscriptions can be processed by Reception Staff.
- Reception Staff can change all Business Objects assigned to the Admission Business Process.
- In the Patient treatment Business Process, Nurses can read the Patients' Medication & Dosage Form.

Hospital Big Picture

The access control Hospital Big Picture generated using Atlas tool is shown in the figure 3, where is possible to analyze all entities that Business Actors can access in hospital environment. Each arrow starts at Business Actor and finishes at an entity that can be accessed by the business actor. Figure 3 displays all accesses of all Business Actors, but it is possible to filter the accesses arrows achieving a more clean view and analysis using the Atlas tool. Filters can be applied in the source (Business Actors) or in the target entities accessed (e.g. Locations, Data Objects, Business Objects).

This view is not restricted to accesses of Business Actors to the Archimate application layer (Application Components and Data Objects), being possible also analyse the Business Actors accesses to Business Objects and physical accesses to Locations.

Hospital Big Picture board view was built with graphical queries representing the policies previously defined for this case study. The queries can be checked in figure 2.

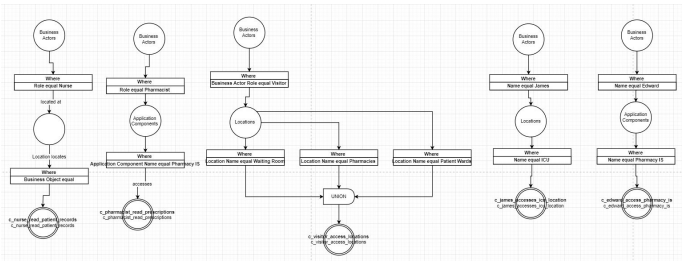

Figure 2. Queries used in policies' definition

Hospital Business Actors VS Applications

This view intends to demonstrate the accesses of Business Actors to Data Objects in the different Information Systems presented in the hospital. The view covers only the policies defined above that are relevant to reflect the accesses in Applications Components. Figure 4 displays an ACL view with a pop-up raised showing the Data Objects that Business Actors can access assigned to a specific Application Component. In this case study, each Application Component represents a different Information System with a specific purpose in the Hospital environment.

EVALUATION

This paper follows the Design Science Research Methodology (DSRM) approach proposed in [19]. In this section, the results of the demonstration are discussed and compared with the presented objectives

The solution proposed in this paper was demonstrated using board and ACL views in the Atlas tool [24] using the Hospital case study.

Our solution is evaluated using the artifact evaluation proposed by Prat et al.[21] that helps DSR researchers, providing a holistic view of artifact evaluation. Prat et al.[21] proposed a DSR paradigm to build and evaluate a taxonomy of evaluation methods for IS Artifacts. The taxonomy is divided into six dimensions: criterion, evaluation technique, form of evaluation, secondary participants, level of evaluation, and relativeness of evaluation. The first dimension answers the question *"what"* while the remaining dimensions answer the question *"how"*.

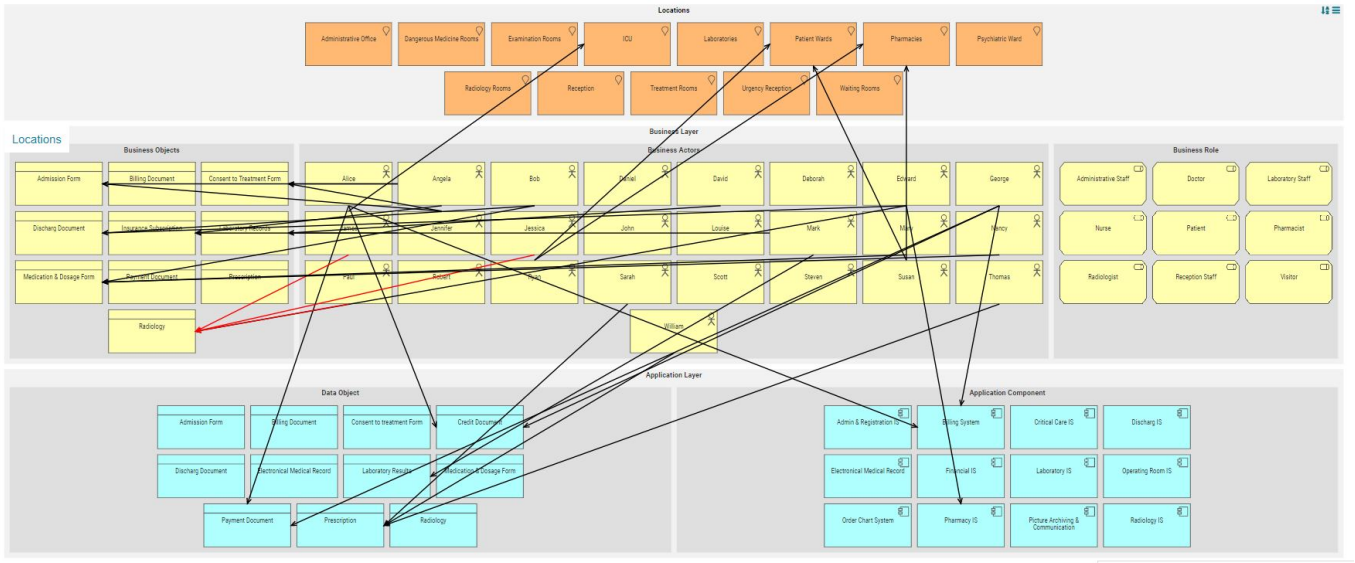

Figure 3. Big Picture View with all accesses allowed in Hospital

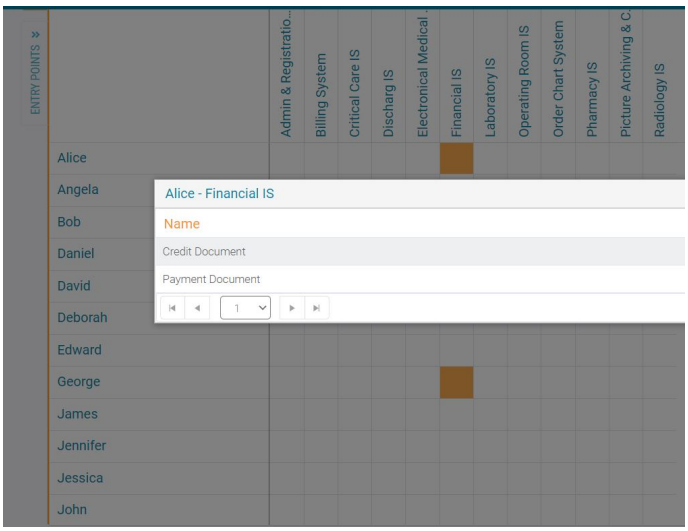

Figure 4. Business Actors VS Applications ACL View

The model also proposes a hierarchy of evaluation criteria with the fundamental dimensions of the system which are goal, environment, structure, activity, and evolution. These dimensions are deeply categorized in evaluation criteria and sub-criteria.

The evaluations criteria selected for our solution were: Goal - **Validity** and **Technical Feasibility**, Environment - **Usefulness**, and Structure - **Completeness**. Regarding to evaluation methods, the following were selected: Evaluation Technique - **Illustrative Scenario**, Form of Evaluation - **Analysis**, Secondary Participants - **Practitioners**, Level of Evaluation - **Instantiation (Ficticious Example)**, and Relativeness of Evaluation - **Relative**.

For each evaluation criteria we provide a definition for it, and the context application in this paper evaluation.The definitions provided are present in Prat et al. work[22].

- **Validity** - "Validity means that the artifact works correctly, i.e. achieves its goal correctly." In our work this will be demonstrated changing policies and entities attributes.To consider the solution validated, the access control changes must be reflected in both views in the same way. In other words, if a Business Actor gains access to a target entity in one view, that access must be reflected in all views. We started by changing the attribute values of entities in the Data Explorer of Atlas tool. We then selected a query which used the specific attributes changed. When the attributes relevant for the query were changed, the accesses were reflected in the board and ACL views of access control. **Using this practical example in Atlas, we verified that the artifact works correctly.**
- **Technical Feasibility** - "Evaluates, from a technical point of view, the ease with which a proposed artifact will be built and operated." This criteria measures how easy it is to develop a new query to translate policies into the access control views. We interviewed an expert in access control management of Information Systems to gathers its feedback about the practical implementation of our solution. This specialist works daily with accesses based on RBAC model in an ERP system used in the largest companies in the world. Our solution details and the basic Atlas concepts were clearly explained to him presenting the main benefits. The case study was also presented and the specialist was encouraged to implement a query reflecting a new policy in the Hospital Case Study. The interviewed internalized the main concepts and produced the desired outcome. The specialist's feedback was that some adaption time is required for this new approach and some initial effort is required to create the views, but after this time, the workload of access control management is less compared to other approaches such as RBAC.
- **Usefulness** - "The degree to which the artifact positively impacts the task performance of individuals". This aspect is

evaluated in terms of effort required by the access control administrator in maintaining access control policies and entity attributes. Usefulness was demonstrated using the upload files to update automatically the classes and attributes in Atlas Data Explorer. The automatic update will reduce the effort required by the access control administrator because in a real scenario this information can be integrated using API's. Compared to other models, such as RBAC, no need manual update executed by access control administrator when the data changes.

- **Completeness** - "The degree to which the structure of the artifact contains all necessary elements and relationships between elements." In one hand, the elements aspect is provided by Archimate[2] framework core elements and their relationships. Each element represents an entity throughout enterprise structure. Archimate is an Enterprise framework that provides a large number of entities in different layers of organizations. For each kind of entity exists relationships with different meanings and strengths. On the other hand, ABAC provides the necessaries flexibility and fine-grain approach enough for organizations' policy definition. **Archimate entities and their relationships combined with ABAC model provide flexibility, robustness, and fine-grain approach to design and implement the access controls reflecting the organizations' reality.**

The solution artifact is considered successful because the views generated can clearly display all users' access within the organization. Accesses in the views changes when the entity attributes change or when the policies change. Our solution also reduces maintenance costs for the access control administrator.

CONCLUSION

Contributions

Security starts by the definition of suitable access control policies within an organization. The problem this paper seeks to solve is the mismatch of access controls in components of different layers in enterprise architecture, with particular emphasis on the mismatch of Information Systems misalignment, providing a centralized access control system to manage and analyze accesses throughout an entire organization. **Our solution provides a holistic approach where it is possible to verify, in a central system, the relationship of enterprise elements in terms of access control.**

The views developed in this paper are based on graphical queries, which provide a flexible configuration with fine-grain approach, and robustness to reflect the policies used by organizations.

The solution proposed in this paper is a merge between the flexible access control model ABAC[11] and the enterprise architecture framework Archimate[2].

While ABAC provides a flexible attribute-based policy definition, Archimate can provide the entities, relationships, and views used to analyze the accesses throughout the organization.

Our work enables the implementation of access controls in different layers of Archimate core framework.

More precisely, our contribution is summarized in the following topics:

- Flexible and fine-grain access control model based on Archimate enterprise architecture framework and its entity relationships and attributes.
- Holistic views for access control monitoring and analysis in organizations
- Centralized repository ACMS for policies' definition based on graphical queries, entities, and their attributes

Our approach will reduce the workload of access control administrator because all access control information is centralized in one system, which reflects the reality of the entire organization.

The access control changes happen frequently with employees being hired, leave the organization, or change their roles within organizations. The policies based in attributes will reflect the changes in access controls with few work by the access control administrator.

In order to demonstrate the functionality of our solution, we developed two illustrative scenarios with different scopes within the organizations. Hospital case study has access controls implemented in different layers of Archimate[12] while ERP case study considers the accesses in different modules and sub-modules of an ERP system.

According to the results obtained, we can state that the goals of our work were accomplished, since in both illustrative scenarios were possible to demonstrate and evaluate the access control model proposed in our solution.

Limitations

The main limitations identified in the proposed solution are the following:

- It is assumed that Information Systems allow the application of policies defined centrally in ACMS.
- It is assumed that the access control administrator can gather the policies from different enterprise architecture layers to build the graphical queries.
- It is assumed that the data required to update ACMS entities is possible to gather and is generated and provided in a suitable manner.
- Our solution implementation in the Atlas tool does not contemplates access control audit trail for entities.

These drawbacks make the solution dependent from external factors to update properly the entities' attributes and policies.

Future Work

After the development this paper, as future work, we identified the following considerations to overcome the limitations aforementioned.

- Integration development to send the attributes' changes from different sources to ACMS in real time . This includes the development of a specific API common to all entities that integrates with ACMS.
- Integration development of entity to check access controls (e.g. Information Systems) with ACMS as PDP does in

XACML framework. Entities can request the access providing the four variables of ABAC (subject, object, action, and environment) and would be returned a response with possibilities *"Granted"* or *"Denied"*. The response will be based on the policies already defined in graphical queries used to create the views.

- Development of audit trail feature to monitor the policies changes, entities changes, and access control changes. This feature will provide a quite complete analysis of access control over time. Sometimes an analysis of access controls in the past is needed to audit entities or to know the entire history of a subject.

The future work proposal will integrate ACMS with the remaining entities related to access control, which should improve the work experience for the access control administrator.

REFERENCES

[1] 1987. A Guide to Understanding Discretionary Access Control in Trusted Systems. *National Computer Security Center, Fort George G. Meade, Maryland* (September 1987).

[2] 2019. ArchiMate® 3.1 Specification. *The Open Group Standard* (November 2019).

[3] D. Clark and D. Wilson. 1987. A comparison of commercial and military computer security policies. *IEEE Symposium of Security and Privacy* (April 1987), 184–194.

[4] E.Yuan and J. Tong. 2005. Attributed Based Access Control (ABAC) for Web Services. *Proceedings of the IEEE International Conference on Web Services (ICWS'05)* (2005).

[5] D. Ferraiolo, J. Cugini, and R. Kuhn. 1995. Role-based access control (RBAC): Features and motivations. *Proceedings of 11th Annual Computer Security Application Conference* (December 1995), 241–248.

[6] D. Ferraiolo, R. Sandhu, and et al. 2001. Proposed NIST Standard for Role-Based Access Control. *National Institute of Standards and Tecnhology* 4, 3 (August 2001), 224–274.

[7] K. Gaaloul, S. Guerreiro, and H. Proper. 2014. Modeling Access Control Transactions in Enterprise Architecture. *IEEE 16th Conference on Business Informatics* (2014).

[8] K. Gaaloul and H. Proper. 2013. An Access Control Model for Organisational Management in Enterprise Architecture. *2013 Ninth International Conference on Semantics, Knowledge and Grids* (2013), 37–43.

[9] G. Graham and J. Denning. 1972. Protection - Principles and practice. *Proc. Spring Joint Computer Conference* (May 1972), 417–429.

[10] V. Hu, D. Ferraiolo, and et al. 2014. Guide to Attribute Base Access Control (ABAC) Definition and Considerations. NIST Special Publication 800-162. *National Institute of Standards and Tecnhology* (2014).

[11] V. Hu, D. Kuhn, and D. Ferraiolo. 2018. Access Control for Emerging Distributed Systems. *National Institute of Standards and Technology* 51 (October 2018), 100–103.

[12] A. Josey, M. Lankhorst, I. Band, H. Jonkers, and D. Quartel. 2016. An Introduction to the ArchiMate® 3.0 Specification. *The Open Group* (June 2016).

[13] A. Kern and C. Walhorn. 2005. Rule Support for Role-Based Access Control. *Proc. 10th Association for Computing Machinery Symposium on Access Control Models and Technologies* (June 2005), 130–138.

[14] R. Kuhn, E. Coyne, and T. Weil. 2010. Adding Attributes to Role-Based Access Control. *IEEE Computer* 43, 6 (June 2010), 79–81.

[15] B. Lampson. 1971. Protection. *5th Princeton Conference on Information Sciences and Systems* (March 1971), 437–443.

[16] M. Masrom and et al. 2010. Activity-oriented access control to ubiquitous hospital information and services. *Information Sciences* (2010), 2979–2990.

[17] M. Masrom and A. Rahimly. 2015. Overview of Data Security Issues in Hospital Information Systems. *Pacific Asia Journal of the Association for Information Systems* 7, 4 (December 2015), 51–66.

[18] B. Parducci and H. Lockhart. 2013. *eXtensible Access Control Markup Language (XACML) Version 3.0*. Technical Report. OASIS. `https://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf`

[19] K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee. 2007. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems* 24, 3 (2007), 45–77.

[20] R. Pilipchuck, S. Seifermann, and R. Heinrich. 2018. Aligning Business Process Access Control Policies with Enterprise Architecture. *CEEC, Ljubljana, Slovenia* (November 2018).

[21] N. Prat and et al. 2014. Artifact Evaluation in Information Systems Design - Science Research - A Holist View. *Pacific Asia Conference on Information Systems* (2014).

[22] N. Prat and et al. 2015. A Taxonomy of Evaluation Methods for Information Systems Artifacts. *Journal of Management Information Systems* 32, 3 (2015), 229–267.

[23] S. Rohani, R. Belchior, R. Cruz, and R. Deters. 2021. Distributed attribute-based access control system using permissioned blockchain. *World Wide Web (2021)* 24 (March 2021), 1617–1644.

[24] P. Sousa, R. Leal, and A. Sampaio. 2018. Atlas: the Enterprise Cartography Tool. *Proceedings of 8th the Enterprise Engineering Working Conference Forum* 2229 (2018).

[25] X.Jin, R. Krishnan, and R. Sandhu. 2012. A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC. *DBSec: IFIP Annual Conference on Data and Applications Security and Privacy* (2012), 41–55.