TÉCNICO LISBOA

Views for Access Control Management in Enterprise Architecture

Tiago da Purificação Agostinho

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa

Examination Committee

Chairperson: Prof. Pedro Tiago Gonçalves Monteiro
Supervisor: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa
Member of the Committee: Prof. Sérgio Luís Proença Duarte Guerreiro

November 2022

Acknowledgments

I would like to thank my family for their friendship, encouragement, support, and caring over all these years, for always being there for me through thick and thin and without whom this project would not be possible. I would also like to thank the understanding for my absence throughout all these years.

I would also like to acknowledge my dissertation supervisor Prof. Pedro Manuel Moreira Vaz Antunes de Sousa for their insight, support, and sharing of knowledge that has made this Thesis possible.

Last but not least, to all my friends and colleagues that helped me grow as a person and were always there for me during the good and bad times in my life.

To each and every one of you – Thank you.

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. This work proposes the creation of different views in the Access Control Management System (ACMS) to analyze and monitor access controls in organizations with a holistic approach.

Keywords

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

Resumo

A gestão do controlo de acessos tem um papel importante nas organizações permitindo o acesso à informação de acordo com as funções do colaborador. A informação é propriedade das organizações e é da sua responsabilidade gerir o controle de acessos de forma adequada. O objetivo é garantir que os sujeitos autorizados tenham as permissões mínimas para realizar as operações desejadas sobre os objetos nas diferentes camadas da architetura empresarial das organizações. O controlo de acessos decide "Quem pode fazer o quê?", sendo um mecanismo de segurança fundamental dentro das organizações. Funcionários diferentes nas organização têm funções diferentes e precisam de acessos diferentes para realizar suas tarefas adequadamente. As políticas de controlo de acessos podem ser derivadas das regras do negócios, legislação ou cultura organizacional. O controlo de acessos deve ter diferentes niveis de granularidade, deve ser flexivel e robusto de forma a cumprir os requisitos das organizações. Este trabalho propõe a criação de diferentes visões no sistema de gestão de controlo de acessos (ACMS) de forma a analisar e monitorizar o controle de acessos nas organizações de forma holistica.

Palavras Chave

Controlo de Acessos; Modelos de Controlo de Acessos; Segurança; Funções; Autorização; Vistas; Query; Organização; Políticas; Monitorização; Sistemas de Informação; Archimate; RBAC; ABAC

Contents

List of Figures

Acronyms

AAP	Attribute Administration Point
ABAC	Attribute-Based Access Control
ACL	Access Control List
ACMS	Access Control Management System
API	Application Programming Interface
DAC	Discretionary Access Control
DSR	Design Science Research
DSRM	Design Science Research Methodology
EA	Enterprise Architecture
EAM	Enterprise Architecture Management
ERP	Enterprise Resource Planning
HIS	Hospital Information System
MAC	Mandatory Access Control
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point
PRP	Policy Retrieval Point
RBAC	Role-Based Access Control

SOA Service Oriented Architecture

TCSEC Trusted Computer Security Evaluation Criteria

XACML Extensible Access Control Markup Language

XML Extensible Markup Language

1

Introduction

Contents

1.1 Research Motivation

Security represents an important concern in organizations, as it is seen as a need for organizations' subsistence.

Access control, which can be physical, technical, or administrative, is a mechanism to ensure information security [1]. 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 has access to what?". The goal of access control systems is to provide means to protect resources from unauthorized access attempts. Access control is an important component to ensure security and privacy requirements in organizations, and prevent unauthorized access.

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.

Every organization has its own policies derived from business rules, legislation, or organization culture (not necessarily technical decisions). These policies can be used to specify access controls and verify under what conditions each subject can access each object.

Policies are a set of rules to prevent unauthorized access to resources that 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 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. 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, such as the access to information systems, or accesses to specific documents at the business level. This view upon the access rights, in terms of application layer and technology layer of the organization, 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 is composed by many layers like 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.

The proposal of this work is to present an access control management solution based on views and centered in a system to manage and analyze the accesses of organizations with Atlas [2]. In our scenario, the focus is on protecting resources associated to different enterprise layers of organizations from unauthorized subject access.

1.2 Problem Description

Organizations needs different access controls at different levels of the enterprise architecture, such as at the Business Layer, Application Layer, or Technology Layer. 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.

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.

Access rights management is a concern to any organization, in which each subject should have the least rights necessary to perform its duties. Access rights are usually implemented and managed on a system-by-system basis by an access control system administrator. 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.

When each information system has its own access decision and enforcement points, it is expensive to update access control policies in different information systems. 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.

The flexibility of customizing access control systems is sometimes insufficient to meet organizational

needs and may lead to a lack of security. For example, Role-Based Access Control (RBAC) [3] is widely used in business contexts, but managing access rights based on roles may not be specific enough, or in certain cases subjects may only need some access rights contained in a role to perform their tasks. This approach does not follow the principle of least privilege. Flexibility is needed to maintain different levels of granularity covering different sizes of entities in organizations (e.g. organizations, departments, teams, employees).

There are models with flexibility, such as Attribute-Based Access Control (ABAC) [4], but some issues in the model have been ignored or proposed as future work or it is very complex to implement considering the organizations' needs [5]. On the other hand, models such as RBAC [3] are easier to implement, but are not flexible enough to face the organizations' needs and may not provide the desired fine-grain approach required organizations.

There are some access rights management features, but many of them focus on network resources (devices) and are not flexible enough to cover most of the business needs at Business Layer level. Some solutions on the market have an access control management system, but are vendor-specific and require that part of the infrastructure be owned by that vendor. An inadequate Access Control Management System (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 holistic 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.

1.3 Dissertation Objectives

Access controls can be managed and interpreted from different points of view. For example, is possible to analyze which subjects can access a particular object, which permissions the individual roles have, or which objects a particular subject can access.

To address the problems mentioned in Section 1.2 , the goal of this work is to develop a orchestration system to generate different views that will support the access controls management. These views will enable the analysis and monitoring of accesses throughout the organization.

Views will be created using graphical queries that reflect the organization policies. Policies may have different sources and may affect different layer of enterprise architecture in organizations. Policies are based on attributes of subjects, objects, actions, and environment.

Policies, entities, and their respective attributes will be stored in a repository accessed by ACMS.

Since the views are based on orchestrated systems, the consistency of access rights as well as the efficiency in the maintenance them is increase. The goal is to implement policies using logical formulas (queries) to grant or deny the accesses which makes the implementation simple, easy and flexible. Views will show the current status of access rights, acting as a facilitator to the access control administrator.

A demonstration is then conducted in two fictitious scenarios is performed to simulate how the proposed solution behaves in organizations for two different scopes.

1.4 Research Methodology

The research methodology present in this work will follow the six steps proposed in DSRM (Design Science Research Methodology) [6]. The goal of DSRM is to provide guidance to researchers that works on design science. The methodology steps are described below:

1. Problem identification and motivation – In this step, the specific research problem is defined and the value of a solution is justified. In this work, this is identified in section 1.1 and section 1.2. The basis for problem identification and motivation were scientific papers and articles related to access control models, enterprise architecture, and related problems.

2. Define the objectives for a solution – In this section, the solution goals are defined considering the problem definition and the knowledge of what is possible and feasible. In this work, this is defined in section 1.3.

3. Design and development – This step is responsible for the artifact creation. The artifacts are potentially constructs, models, methods, or instantiations. In this work ,the design is described in chapter 4 while the development is done during the dissertation.

4. Demonstration – This step demonstrates the use of the artifact to solve at least one instance of the previously identified problem. In this step, the artifact may be used in experimentation, simulation, case study, proof, or other appropriate activities. The demonstration is performed in chapter 5.

5. Evaluation – In this step, is analyzed and measured how well the artifact supports a solution to the problem described. This step involves the comparing the goals of the solution to the actual results observed during the demonstration. The evaluation step of this work is performed as described in chapter 6.

6. Communication – Communicate the problem and its importance, the artifact, its utility and novelty, the rigor of the design, and its effectiveness for relevant audiences. Communication is provided in an extended summary delivered at the same time of this dissertation.

1.5 Document Structure

The remaining document is structured in the following sections as follows. This section, Introduction, provides the scope of this work. This section contains the research motivation for the proposed work, the problem description, the dissertation objectives defined, and the research methodology used. chapter 2 provides the background of this work and presents the core frameworks and tools analyzed and used during this work. The chapter 3 refers to the state of the art, referring the similar and relevant work performed by the community to solve the same or similar problem here presented. An analysis is also presented that highlights the positive aspects and limitations of previous solutions and compares them. The chapter 4 describes a solution proposed in this work, and also defines the scope where the work is inserted. This section also presents the models' design and the details that are relevant to the solution. The chapter 5 demonstrates the developed solution in two distinct fictitious scenarios. The chapter 6 describes the means to evaluate and validate our proposed solution. This section describes some guidelines and steps that to be followed to validate the solution. The chapter 7 concludes this document, provides a final remark on this work, including contributions, limitations, and a discussion for future work.

2

Background

Contents

This chapter - Background - presents the Archimate framework [7] [8] and the tool Atlas [2], which are two cornerstones of this work. In the following, we analyse the purpose of two tools and explain what they serve in our work.

2.1 Archimate

Archimate [8] [7] 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 design and operation of business processes, organizational structures, information flows, IT Systems, and technical and physical infrastructure. Archimate provides a graphical language for representing 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 architecture together and their relationships and dependencies [8]. Enterprise architectures can use ArchiMate to model, among others, business processes, how business processes are supported in Information Systems, and how these IS are maintained by IT Infrastructure. The flexible approach of the Archimate language allows architects and other stakeholders to use their own views on the Enterprise Architecture (EA). The three Archimate core layers are defined as follows [1]:

- Business Layer - describes the products and services in the organization that are realized through business processes and executed by active entities such as actors or roles.
- Application Layer - supports the business layer through application services realized by application components.
- Technology Layer - provides infrastructure services (e.g., storage, processing, network communications) required to run the applications, realized through physical devices and system software.

Beyond the three layers mentioned above , Archimate has Strategy, Implementation & Migration, and Motivation layers in the full framework, but in this work we will only use the layers of the core framework. Archimate also has the aspects defined in *Active Structure Aspects*, which represent the structural elements, *Behavior Aspects* which represent the behavior performed by the Actors, and *Passive Structure Aspects* which represent the objects on which the behavior is performed. The core concepts of Archimate are presented in combining core layers and aspects.

Building and maintaining an updated and coherent EA is a complex task because it involves stakeholders with different backgrounds which may use different notations. Archimate provides views that are comprehensible for stakeholders, supports decision making, and enables analysis of impact across the organization.

Table 2.1: Archimate core concepts

	Passive Structure	**Behavior**	**Active Structure**
Business	Business Objects	Business Services, Business Functions, Business Processes	Actors and Roles
Application	Data Objects	Application Services and Application Functions	Application Components and Application Interfaces
Technology	Artifacts	Infrastructure Services and Nodes	Devices, Networks and, System Software

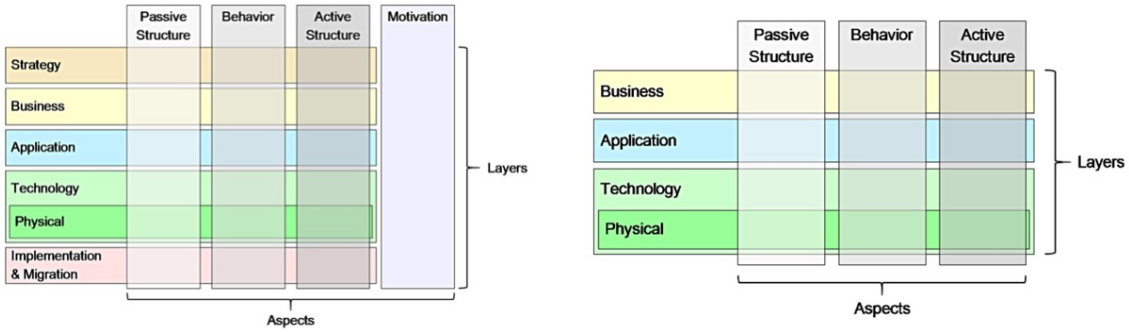

Figure 2.1: Archimate full framework and Archimate core framework

2.2 Atlas

Atlas [2] is a web-based enterprise architecture tool that helps organizations to keep their architectural models up-to-date 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 to analyze the enterprise models of the past (AS-WAS models), the enterprise models of the present (AS-IS models), and the enterprise models of the future (TO-BE models), visualizing the transformations in enterprises over the time.

Atlas has embedded the process of enterprise cartography which, is 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 minimal 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 the configuration of behavior associated with blueprints using queries and rules.

During our research work, we will use the features *Blueprint Explorer, Data Explorer, and Query Explorer* in Demonstration (chapter 5) and Evaluation (chapter 6) phases of Design Science Research Methodology (DSRM).

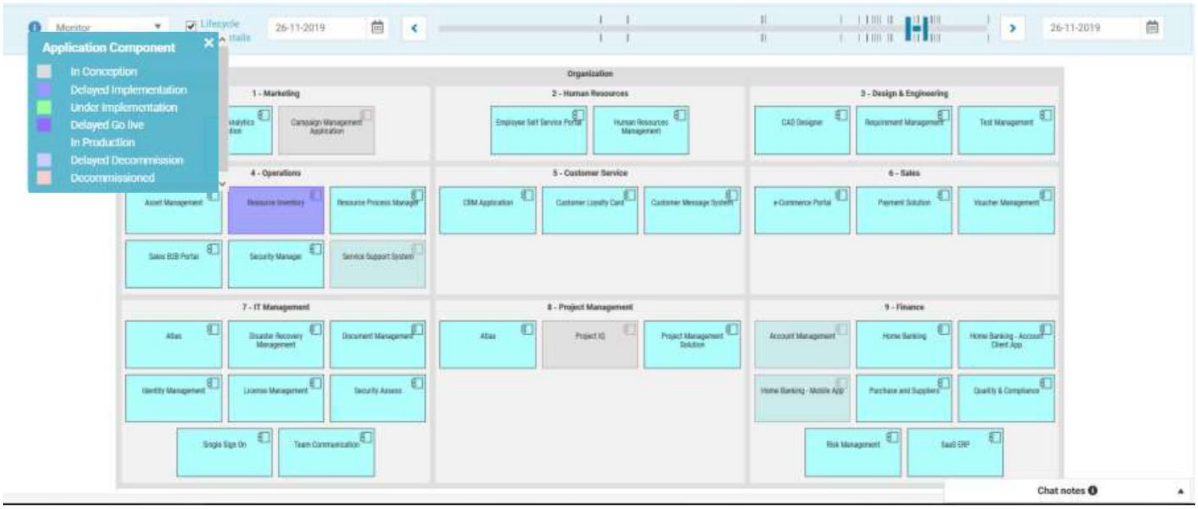

Figure 2.2: Sample of Atlas Blueprint View

3

Related Work

Contents

This section presents the contributions from the community related to the proposed problem. The literature gathering and review related to enterprise access control and its relevance for this work is presented below.

3.1 Mandatory Access Control and Discretionary Access Control

Typically, access policies that protect resources are categorized as Discretionary Access Control (DAC) or Mandatory Access Control (MAC). Different access control models have been developed to serve different purposes to control the access to information in organizations.

In DAC [9], subjects can manipulate the authorizations of other subjects to access objects.

In MAC, policy decisions are determined by a central authority such as the system administrator, in the opposite of DAC where accesses are managed by objects' owners.

3.2 Access Control List

Lampson [10] proposed a notion of subject and object with a simple type of access control applied to systems with shared objects, where subject IDs are linked to individual object IDs in a matrix. The rows represent the subjects and the columns represent the objects. Each entry in the table represents a subjects' access rights to an object for the intersection of the column and row. In other words, each entry [i,j] in the matrix represents the access granted to subject i over the object j. The matrix can be read either by rows or by columns.

Later, Graham and Denning [11] went further deep and defined access attributes that any subject can perform over any object. They presented three different problems to be solved: representing the protection state, allowing subjects to access objects only as permitted by the protection state, and allowing subjects to change the protection state in specific situations.

Graham and Denning proposed an access control model based on a state machine in which the protection state of the system is described by a matrix. The attributes are the actions that subject S is allowed perform on object O. There is a monitor associated to each type of object to validate the access of these objects. Graham and Denning proposed models that start when S initiates the access to O to execute an action A. Then is provided a triple (S, A, O) to the monitor of O. The monitor accesses the matrix to determine the position M[S, O]; if the access attribute is in the matrix position, access is granted, otherwise access is denied. Graham and Denning also included a set of rules that allow manipulation of matrix entries by subjects.

This manipulation depends on the attributes assigned to subjects, and the approach allows untrusted subjects to grant access rights to someone who does not have to them, granting the system rights to

unauthorized subjects because there is no control over which rights are passed from one subject to another, as demonstrated by Harrison, Ruzzo, and Ullman [12], taking as a baseline the work developed by Graham and Denning. In other words, it is impossible to ensure that an authorized subject does not receive access rights improperly during the chain of subjects' delegation.

This solution is not scalable because the number of entries increases exponentially with the number of users and objects, and the matrix can become sparse because subjects do not need access rights to all objects. The sparse can be avoided if the access rights are stored in a triple with the form of (S_i, A_{ij}, O_j) where S_i is subject, A_{ij} is action and O_j is objects. This access control model is not suitable for large organizations because of the large number of entries to maintain.

3.3 Bell and LaPadula Model – Multi Level Security (MLS)

Information is property of organizations and the subjects should not be able to set up their own permissions. The previous statement follows a MAC approach and is one of the MLS base. Bell and LaPadula [13] proposed a mathematical model, with well-defined security properties and proofs, to evaluate security in information systems, which can be defined by the statement "no-read-up, no write down ".

The solution presented by Bell and LaPadula [13] can be summarized in two rules that express the relationship between subjects and objects. In the proposed solution, each subject and each object is labeled with an attribute called security level. The **first rule (simple security property)** referred that the subject only can read classified information from objects if its security level is greater than or equal to the security level of the object. The **second rule (*-property – star property)** referred that a subject is allowed to simultaneously has access to read one object and write access to another one if the security level of the second is greater than or equal to the security level of the first. The subject only can write into an object only if the security level of the object is higher than the security level of the subject.

The *-property prevents the transfer of data from an object with a higher security level to an object with a lower security level. This model was developed for military purposes and is one of the most effective models for maintaining confidentiality. Examples of military security levels are *unclassified*, *confidential*, *secret*, and *top-secret*.

3.4 Biba's Model

Biba's integrity model [14] was introduced as an alternative to the model of Bell and LaPadula. While in Bell and LaPadula the main concern is on confidentially, in Biba's model the main concern is on integrity. As with other models related with integrity, the concern is to prevent that unauthorized subjects change

data in protected objects.

The model is characterized by the statement “read up, write down”. This model was developed with the purpose of preventing subjects from corrupting data in objects with a higher security level than their own integrity level, or the subjects from being corrupted by data from objects with a lower integrity level than the subjects, in order to prevent the untrusted modification of information.

3.5 Trusted Computer Security Evaluation Criteria

In 1983, the publication of standards took place by the hand of DoD in the book called Trusted Computer Security Evaluation Criteria (TCSEC) [15], also well-known as the Orange Book due his cover.

The purpose of this book is to provide the standards for manufacturers of hardware and software, showing how they can build their commercial products satisfying the trust requirements.

The book is divided into two parts: Criteria, and Rational and Guidelines. The criteria part consists of four hierarchical divisions (named from division D to division A, ascending divisions – A is the highest division) providing the basis for the efficiency of security controls. Each division contains classes for evaluating the level of trust. Division D, minimal protection, includes one class reserved for systems that have been evaluated but failed to meet the security requirements for a higher evaluation class. Division C, discretionary protection, is divided into two classes, C1 and C2, which contain requirements for DAC. Division B, mandatory protection, provides three escalating classes of requirements, which contains requirements for MAC. Division A, verified design, contains one class of requirements for formal design specification and verification. The second part is divided into six sections and provides a discussion for basic objectives, and the rationale and policies behind the development of the criteria. The TCSEC approach to authorization is not applicable to the commercial sector as initially intended.

3.6 Clark and Wilson Model

Clark and Wilson [16] compared the policies between military organizations, such as the multi-level security described in the orange book, and commercial organizations presenting the differences. They claim that a lattice model is not sufficient to characterize integrity policies, and different mechanisms are necessary to control disclosure and ensure integrity.

Clark and Wilson [16] stated that the main concern for commercial systems is integrity rather than secrecy. Hence, they proposed a commercial-oriented model to ensure information integrity based on two security principles: **separation of duties (SoD) and well-formed transactions**.

The concept of well-formed transactions ensured that a subject may not manipulate object data arbitrarily, but only in a restricted way that preserves or ensures data integrity. In the operations performed,

the information system should transit from one consistent state to another consistent state. Hence, the information that is in a valid state, should remain in that state after the execution of a transaction. It is common, in well-formed transactions, the recording of all modifications in a log to be audited later.

Separation of duties (SoD) is ensured in a business process if each activity is executed by a different subject. For example, in the business process of procurement, the activities of purchase an item and process payment should be executed by two different subjects. The Clark and Wilson model consists in two types of rules: certification rules and enforcement rules, in a total of nine rules (named from C1 to C5 and from E1 to E4). The integrity of data prevents fraud and errors. The information is modified when authorized by trusted people.

3.7 RBAC - Role Based Access Control

RBAC, originally proposed by Ferraiolo, Cugini, and Kuhn [17] 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 the rights that owners give to objects, and other models were very restrictive because of their original military purpose as demonstrated by Clark and Wilson [16].

The concept is simple, to establish permissions to subjects based on their responsibilities and qualifications in the organizations [17]. Role-based access control [3] solved some of the problems presented above by grouping permissions into roles, which typically represent user permissions in organizations, where each user has predefined permissions associated. The RBAC model consists of several entities: subjects, roles, permissions, sessions, operations, and objects.

A subject may use one or more sessions, and each session assigns a subject to one or more roles. A role may or not may be active in each session. In RBAC the permission to perform operations are not assigned individually to users, but operations are associated with roles.

Each role defines the permissions for each user responsibility, and the assigning of each individual permission to a role reduces the effort and complexity of maintaining the access control system because access permissions are not maintained for each individual user.

A subject may be assigned to one or more roles and a role may be assigned to one or more permissions in a many-to-many relationships. 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 management of the system and reduces the cost and potential errors during the users' permission assignment.

RBAC requires an additional effort to configure, which could represent a challenge and time consumption to many organizations.

In RBAC standard [3], the user is always associated at least to one role in a session, and user permissions may change during the sessions' life. Permissions are assigned to roles and are indirectly

associated to users. RBAC96 framework [18] is composed by four conceptual models mentioned below:

- $RBAC_0$ contains the core concept of the model;
- $RBAC_1$ contains the addition of the role hierarchy to $RBAC_0$;
- $RBAC_2$ contains the addition of static and dynamic constraints to the core concepts;
- $RBAC_3$ contains all aspects of $RBAC_0$, $RBAC_1$ and $RBAC_2$.

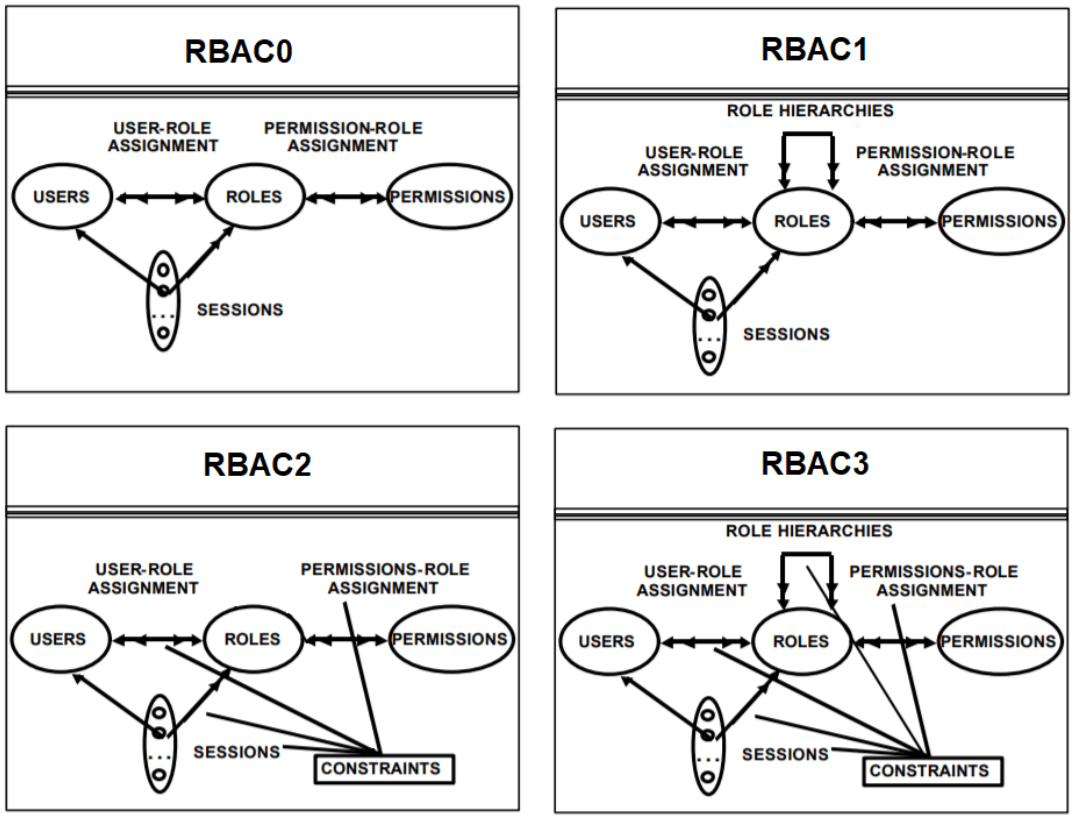

Figure 3.1: RBAC Models

Role hierarchies are suported in RBAC, which define a seniority relationship between roles, with senior roles inheriting permissions from their juniors. Role hierarchy also acts as a facilitator to access control administrators simplifying the assignment, review, and revocation of permissions. A subject assigned to a role at the top of the hierarchy has also indirectly assigned the permissions of lower roles in the hierarchy.

RBAC also includes user-role activation, where it can be possible explicitly declare that the user-role relationship is active, and constraints for user-role assignment where two or more roles are mutually

excluded to be assigned to users. Permissions are positive only, and denials can be handled with constraints.

The four models (the fourth model includes the features of the first three) presented in RBAC96 [18] provided a strong basis for RBAC standardization [3].

$RBAC_2$ constraints include static separation of duties (based on user-role assignment) and dynamic separation of duties policies (a user is authorized to belong to a role only if that role is not mutually exclusive with other roles which the user), based on role activation, initially proposed by Clark and Wilson [16] to reduce the probability of fraud when more than one individual in the process is involved (for example, one individual make a request and another individual approves the request).

RBAC is widely used 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 and the assumption that access rights are centralized in an organization. This type of approach is not verified in some cases, such as web-based applications, where access rights may be distributed and not centralized.

The complexity of RBAC may increase with the complexity of organizations, and in some cases the number of roles may be higher than the number of users (considering many-to-may user-role assignment). The number of roles may increase in an organization with the number of shared functions. For example, an employee may be assigned to the production and the maintenance departments and perform only some activities in each department for the function assigned to him. 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 organizations are not well structured and do not have well-defined functions, or when organizations have shared services. In both cases, the RBAC complexity increases and it is not easy to perform a centralized management.

3.8 ABAC - Attributte Based Access Control

RBAC brought a simple centralized and simple method to manage the access rights, but had limitations, specially when applied to internet and distributed systems, where centralized management is more difficult. RBAC is often cumbersome to set up and manage, and the user roles sometimes is not easily expressed in access control policies.

The lacks of RBAC were the trigger for the appearance of ABAC and its standardization [4] was an important step for its acceptance in the community after several proposals of the model in the literature.

ABAC provides flexibility and there is no need to express the relationship between subjects and objects individually. 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, which makes the solution not scalable [19].

ABAC is flexible allowing the possibility to be configured like RBAC, DAC, or MAC. Xin, Krishnan, and Sandhu [20] demonstrated with ABACα the flexibility and custimization of ABAC to be implemented in the previous well-accepted models such as MAC, DAC, and RBAC.

ABAC has some advantages comparing to other access control models such as:

- Provides a fine-grain approach and flexible access control by allowing a multiple attributes in policies used to make access control decisions;
- The implementation of complex policies is simple;
- It can provide dynamic access control decisions incorporating environment attributes into the decision making [21].

ABAC is also well adapted to access control in distributed systems and can be used to prevent security breaches and fraud [22].

The essence of ABAC is based on the evaluation of attributes assigned to subjects and objects, requested operations, and environment conditions used in policies to grant or deny access rights. The history of access attempts (permit or denied), and the changes in policies and attributes can be stored using blockchain for audit purposes [21].

Biswas, Sandhu, and Krishnan [23] proposed LaBAC, where users are assigned to a label called uLabel and objects to a label called oLabel. Labels can represent atomic values (e.g. age) or a set of values (e.g. roles).

In LaBAC, authorization policies are represented only using the enumeration of user labels (uLabel) and object labels (oLabel). A policy is composed by a subset of tuples combining user labels and object labels. There is a function to check authorization using following criteria:

- Subject s is assigned a value ul;
- Object o is assigned a value ol;
- Policy p for action a contains the tuple (ul, ol).

In LaBAC, a user label value can be assigned to multiple users and an object-label value can be assigned to multiple objects. The policy in LaBAC is defined as a subset of tuples resulting from the combinations of user labels and object labels. The authorization is checked by a function that receives user, object and, action.

Authorization policies in ABAC can be specified by logical formulas (e.g. programming languages) with the attribute values and by enumeration. ABAC goes beyond traditional access control models by providing flexible frameworks that enable its enforcement in a distributed and interconnected enterprise

world. Policies govern the access rights based on attributes independently the number of users and objects.

ABAC [4, 5] represents the most flexible solution described in this document with a fine-grain approach to the access control needs of organizations. The flexibility allows a large number of variables' combination to built rules and policies. The ABAC framework includes four layers in its architecture: enforcement, decision, administration, and access control data. Each element should have a unique identifier (such as a name or ID). The following elements are commonly present in most ABAC systems:

- Subjects (S) - Represents the set of all subjects that may access the system. These subjects can be users or a process working on behalf of a user. Subjects may across the organization boundaries when exists information sharing between organizations.
- Objects (O) – Represents the set of all objects protected by the system.
- Attributes (A) – Represents the set of all attributes in the system.
- Actions (Actions) – Actions represents the desired actions that a subject can execute over the desired object. For example, "create file" or "delete document".
- Policies (P) – Represents the set of all policies in the system. Policies are a set of rules that allow the subject to access the object. Each policy is a set of rules defined according to the principles or culture of the organization. A policy may belong to a policy set. Policies provide the means for describing what needs to be secure.

ABAC is a way to grant or deny subjects access to objects based on attributes. Attributes are assigned to subjects, objects, and environment throughout relationships. Attributes are classified into the following categories:

- Subject Attributes – Attributes that belong to the subjects of the system. These attributes can be subject ID, birth date, job title, role, security clearance, and so on.
- Object Attributes – Attributes that belong to the resources of the system. These attributes contain information about the object such as size, file type, or author.
- Environment Attributes – These attributes belong to the current state of the system. These attributes may contain the current hour, the current date, IP address, or the number of users in the system.

Environment is a new concept not considered in RBAC and represents the environment conditions under which the subject tries to grant the access to the object. For example, a subject can only access the object during working hours, otherwise, access is denied.

Figure 3.2 demonstrates a basic scenario of ABAC since the authentication process until the object's access is guaranteed to the subject.

Access rights can change with the changes in attributes, without changing in policies, and ensuring a dynamic access control system. When a new subject is created in the system, only is needed to maintain its attributes, without any other special assignment or configuration related to access control.

The access rights in ABAC may change with policy changes or with attributes changes.

Policy rules in systems are defined using policy languages that support attributes, such as Extensible Access Control Markup Language (XACML) (XML-based) [24], created by OASIS.

Access control policies in ABAC depend only from the attributes and ABAC frameworks to limit the access of subject to object, unlike RBAC where policies are defined with the manual user-role assignment.

Policies may be expressed using logical operators (e.g., AND, OR, =) upon attribute values, returning a Boolean statement (granted or denied) when compares the attributes, for example, *"*$object.author == subject.id$*"* or *"*$TIME \geq 8AM$*"*. Policies defined using logical operators can be quite extensive and complex. Hence, with this flexibility, is not necessary to specify individual relationships between each subject and object, neither additional management when subjects and objects are created or deleted.

Attributes are defined and maintained by the system administrator or can be integrated with an Application Programming Interface (API). When new subjects join the organization, policies and objects do not need to be modified. For example, it is usual many of the subject attributes be common to HR data of the user. In this case, an integration can facilitates the system administrator's life.

3.9 XACML - eXtensible Access Control Markup Language

XACML, which provides a standard framework for ABAC implementation/deployment, appeared with the emergence of Service Oriented Architecture (SOA), is an eXtensible Markup Language (XML) designed to express security policies [24]. XACML also contemplates access requests and responses needed to interact with ACMS. The response is based in the authorization decision. Each XACML request is composed by subject(user), action(operation), object and environment key-value pairs.

Figure 3.2 illustrates a simplified XACML access scenario where a subject requests access to an object through the access mechanism. This mechanism uses policies, subject attributes, and object attributes to determine and enforce the subject operations allowed upon the object. The policies are enforced by the mechanism [4], which is also responsible for collecting information needed about subjects, objects, and environment to make the decision. Each policy is evaluated and rendered in Policy Decision Point (PDP) and the decision is enforced in Policy Enforcement Point (PEP).

XACML policy specification language allows authorization policies definition using logical operators

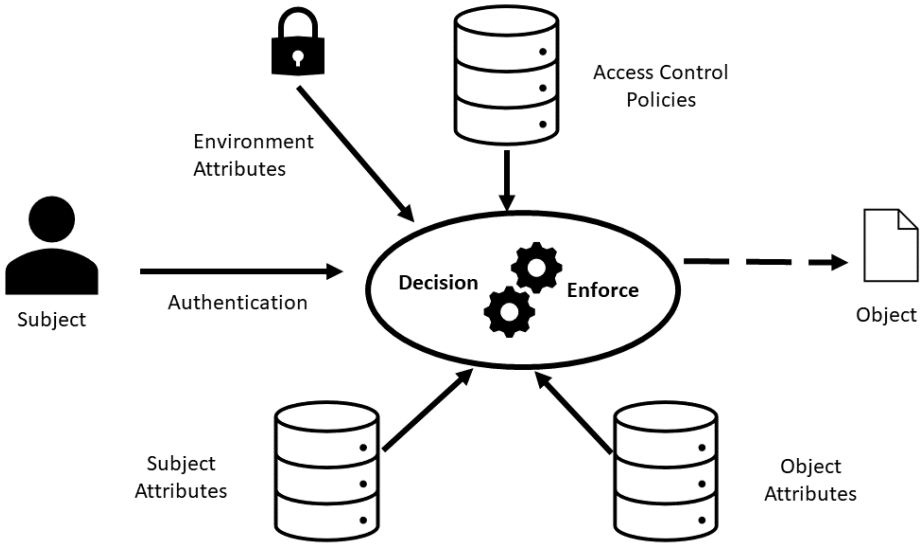

Figure 3.2: Basic ABAC Scenario

(e.g., AND, OR, $\geq$, $\neq$) over attribute values. XACML is composed of the following components [25]:

- XACML policy language, based on Extensible Markup Language (XML) (single and standardized common language) , where the specification of access control requirements is done using policy sets, policies, and rules. Rules are defined using subject, object, and environment attributes. XACML syntax and semantic can be extended to fulfill different requirements in different application scenarios.
- XACML request/response protocol, XML based, where the subject request is evaluated against previously implemented policies and the decision is returned in the response.
- XACML reference architecture to specify the modules, store policies, entities and attributes in repositories, and enforce access control decisions based on policies, executing decision-making.

3.9.1 XACML Policy Language

XACML is a policy language, that supports multiple access control policies, and allows logical operators (e.g. AND, OR, $\geq$, $\neq$) to formulate authorization policies.

Policy sets, policies, and rules are hierarchically related, with policy sets at the on top of the hierarchy and rules at the bottom. Rules are not independent entities and must be encapsulated in a policy. A rule is composed by a target, conditions, effect, obligation expressions, and advice expressions. Only effect

is a mandatory element in a rule, all others are optional.

The target contains the combination of attribute values associated to subject, object, and action to which the rule will be applied.

The condition element is a Boolean expression that should be fulfilled to the rule be satisfied.

The effect is the outcome of the rule and two values are possible: permit or deny.

The obligations expressions are actions that must be performed before or after an access request. Obligations expressions and advice expressions are executed by PEP. Advices can be ignored by PEP, but obligations must always be executed.

A policy is composed by a target, set of rules, obligation expressions, advice expressions, and a rule-combining algorithm, being the last one used for conflict resolution. A policy set is composed by target, set of policies, policy combining algorithm, obligation expressions and advice expressions.

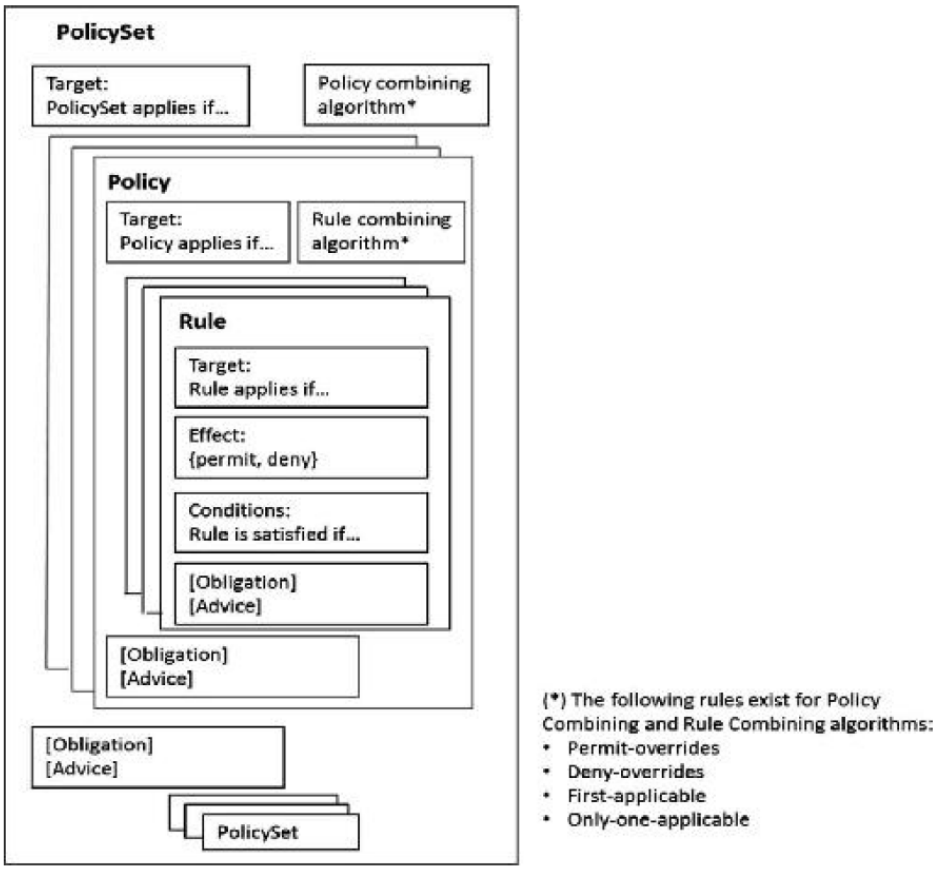

Figure 3.3: XACML Policy Structure

3.9.2 XACML Context

The context of XACML standard represents the requests and responses which are specified under a XML schema. A XACML context request is composed by subject, resource, action, and environment entities. These entities can contain multiple values.

XACML response results from the evaluation of policies by a given XACML request. The response structure is composed by decision, status and obligations (optional). The possible values for decision are permit, deny, not applicable when there are no policies applicable, or indeterminate when some error occurs during policy evaluation.

3.9.3 XACML Data Flow

XACML provides a standard architecture which contains well identified modules with different roles in the authorization process. Each deployment consists of at least one data service, that includes a PEP and a PDP. The PDP queries policies are stored in a Policy Retrieval Point (PRP). If the information in the request of access is not sufficient, the PDP can request additional information from Policy Information Point (PIP) (attributes of subjects, objects, or environment and their associated values). The information in stored PIP and PRP represents the access control data and defines the current authorization state.

Policy Administration Point (PAP) represents the architecture part where the rules, policies, and policy sets are defined using a XACML policy language. This information about access rights is stored in PRP. Additionally, the XACML framework may include an Attribute Administration Point (AAP) to manage data stored in PIP. AAP is used to maintain the attribute names and values of subjects and objects. The functional modules of XACML framework can be implemented in a higher-level language.

Access control verification starts when PEP receives a request and sends it to PDP. Then PDP computes the decision based on the information stored in PRP and PIP. PDP returns the evaluation result to PEP, which is responsible to perform the PDP's decision.

3.10 Merge ABAC and RBAC

Typically, users have a set of operations in information systems assigned to them based on their job functions or organizational role. The ability to perform these operations is called privileges. Sometimes is required additional constraints in RBAC privileges, considering environment variables, to achieve the desired access.

RBAC is usually described as a manual assignment of users to roles, considering the specific functions of each user. Al-Kahtani [26] proposed a model to dynamically assigning users to roles based on rules defined by the organization. These rules take into account the attributes of users and the

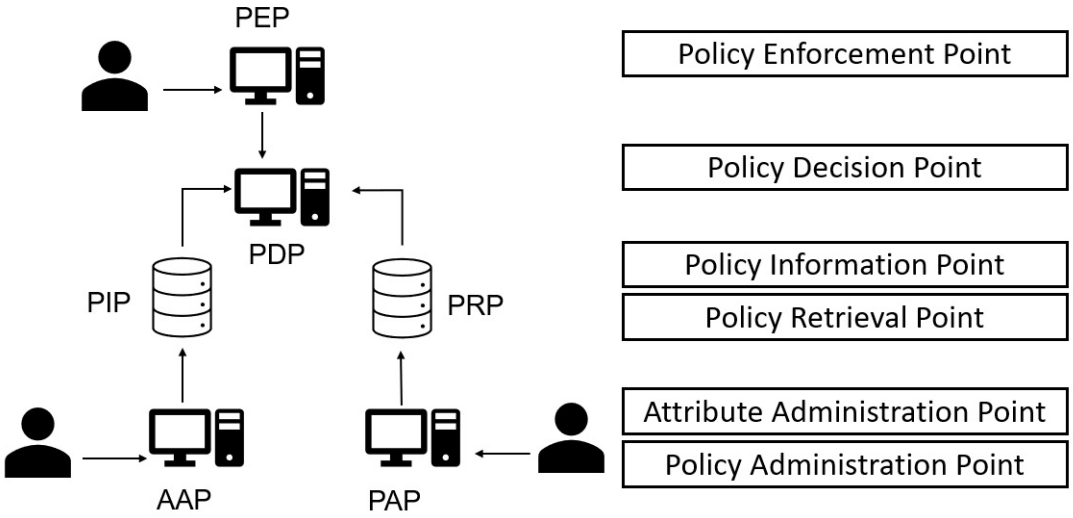

Figure 3.4: XACML Architecture

constraints of security policies in the organization.

Kuhn, Coyne and Weil [27] mentioned that merging the best features of RBAC and ABAC can provide effective access control for distributed and rapidly changing systems. This solution was designed to overcome the disadvantages of RBAC such as time consumption for initial roles' setup or "roles explosion" to fulfill certain requirements of organizations.

Kern and Walhorn [28] proposed a model for dynamically assigning users to roles using rules based on user attributes. Kern and Walhorn argue that attributes can be used to automate the process of roles assignment.

Kuhn, Coyne, and Weil [27] proposed a combination of strategies to take advantages from the strenghts of RBAC and ABAC. The strategies for integrating RBAC and ABAC, RBAC-A (RBAC with attributes), are presented in Table 3.1. ABAC becomes easy the specification of access rules, but for the determination and analysis of users' permissions, the rules should be executed in the same order in which the ACMS applies them. 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 in which RBAC and ABAC can be combined. Some of the models below combine features of ABAC and RBAC models into a hybrid model, balancing the features of each one of them.

- Option 0 is undefined because has no access control in a system without user, role, and attributes
- Option 1 is a pure approach of ABAC systems because only exist attributes in access control.
- Option 2 has users, but no roles nor attributes. This means that roles are not assigned to users and all permissions are accessible becoming the option undefined.

- Option 3 is a weak hybrid model because roles are treated as another attribute and roles are not assigned to a user id.
- Option 4 is a standard Access Control List (ACL) already described in this work where exists the assignment of permissions to users.
- Option 5 treats the user ID as a special attribute of user.
- Option 6 is a standard RBAC system where users are assigned to roles and roles are assigned to permissions. This model incorporates the basic concepts of RBAC already described.
- Option 7 provides dynamic roles. The roles are determined based on users and attributes (eg. system attributes).
- Option 8 is an attribute-centric approach, a pure ABAC system where roles are treated like other attributes (e.g. location, time).Roles are not a set of permissions but only an attribute called *"role"*. Option 8 also provides a mapping between user id and role.
- Option 9 is a role-centric approach and evaluates ABAC rules based on attributes. The rules are only used to constraint the set of permissions allowed for subjects, not to expand them.The role-centric constraints based on attributes provides flexibility to RBAC ensuring the principle of least privileges, limiting the risk exposure, and decreasing the complexity of users' permission review.

Table 3.1: 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

3.11 Access Control in Enterprise Architecture

Security and privacy are a non-functional requirement that affects business processes and IT systems [29]. 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), and enable analysis of relationships between the different enterprise layers. Gaaloul, Guerreiro, and Proper [30] experimented the approach

of access control management in EA. They proposed an access control in EA using the RBAC model applied to Archimate [7], establishing a correspondence between RBAC and Archimate entities.

4

Solution Proposal

Contents

This chapter refers to the 3 rd step of DSRM: Design and Development [6], and it describes our proposal for solving the research problem presented.

The objective of this work is to propose a practical solution for access control in the enterprise environment using the enterprise architecture framework Archimate [8], ABAC model, and the ATLAS [2] platform. In this work, we propose to the merge the entities of Core Archimate framework [7] 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.

4.1 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 research work. 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 work is on access control of Information Systems, business processes, and IT infrastructure inside organizations.

The entities' representation of organization follows the three Archimate [8] Core Layers (Business Layer, Application Layer, and Infrastructure Layer). The presented solution proposes to fill the security requirements using Archimate [7] and the fine-grain and flexible policy approach provided in ABAC access control.

To face the problems presented in section 1.2, we propose to completely externalize the access control management from each information system, creating an access control orchestration, called ACMS, 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 research work 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 research work aims to increase operational efficiency and review in access control management by access control administrator, orchestrating the access rights in a simple point of knowledge, with the same access information architecture shared by the entire organization.

4.2 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 [7]. 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 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 research work are **ACL view, and Board View**.

4.3 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.

4.4 Policies

Policies are defined by laws, business or organizational culture present in each organization. They are then translated into queries/rules 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 [2], 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.

4.5 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. **Then graphical queries can be built based on the attributes of each entity or based on the relationships between entities, or both**.

4.6 Organizational Entities

This research work 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 research work, 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 research work, 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.

4.7 Entities Maintenance

Atlas [2] 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.

4.7.1 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.

5

Demonstration

Contents

This chapter represents the 4th chapter of DSRM: Demonstration [6]. In this phase, we will demonstrate the use of our solution by applying it to two non-real-world scenarios.

Case Study 1 is related with the access control in a Hospital environment crossing different Archimate [7] Enterprise Layers while Case Study 2 simulates the access control in various modules of an Enterprise Resource Planning (ERP). These two case studies have a totally different scopes trying to demonstrate the flexibility and adaptability of our solution for access control in organizations.

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 control. In the two case studies, we will start by defining the policies of each organization and the entities of each layer in the Archimate core [7] framework. Then, the policies will be translated into graphical queries that are applied to the different views, as proposed in 4. These queries will be the base for the views where it is possible to check which objects are accessed by each subject. For each case study, will be presented two views, allowing us to analyze the accesses from different points of view.

The access control views presented for two case studies are *Access Control Big Picture*, where it is possible to analyze the Business Actors accesses in different Archimate core layers of EA, and *Business Actors VS Applications* where 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 (any entity which a Business Actor can access) by an arrow. The source of the arrow is the active entity, while the target of the arrow is the object which is accessed by the active entities. **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.

In *Access Control Big Picture*, the first entity in the query flow is the source entity, which in our case is the business actors of Archimate [7], and the last entity in the query flow is the entity accessed by the business actor. The last entity in the query is the target object in policies which can belong to different layers in Archimate. The target objects can be Locations, Data Objects, Business Objects, Application Components, Devices, Artifacts, etc.

The policies proposed in our research work are written in natural language and then translated into a graphical query that can be analyzed in appendix A. In graphical queries, it is also possible to analyze the flow and the relationship of the 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.

5.1 Case Study 1

This 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.

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.

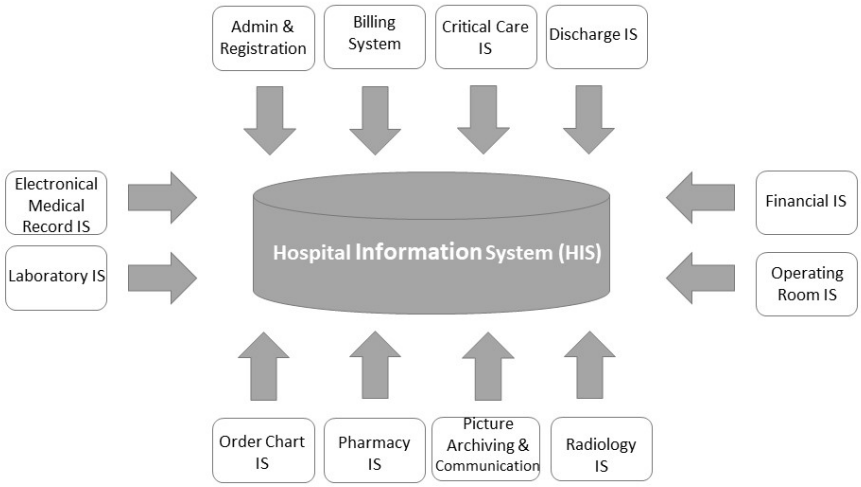

Figure 5.1: Hospital Information System (HIS) Schema

The purpose of each Information System within the Hospital environment is briefly described below.

- **Admission and Registration** - This Information System stores all admissions of patients to the Hospital, including patients admitted for emergency reasons. The check out process is also handled in this Information System.
- **Billing System** - This Information System is responsible to post every bills to customers. Health insurance processing is also done in this IS.
- **Critical Care IS** - This Information System stores patient data related with its clinical situation during the stay in ICU. The access for this IS is very restricted.

- **Discharge IS** - This Information System stores and processes the discharge information performed by doctors.
- **Electronical Medical Record IS** - Electronical Medical Record Information System is integrated with other IS which store patient's data. Its main goal is to provide a wide view about present and past patient data.
- **Financial IS** - This Information System is responsible for accounting and financial part of all businesses process in the Hospital.
- **Laboratory IS** - This Information System stores the results of patients' laboratory tests in full detail.
- **Operating Room IS** - Stores all procedures performed and incidents related to a patient surgery.
- **Order Chart IS** - Provides and stores instant medical information for doctors and nurses during diagnosis and treatment phases.
- **Pharmacy IS** - Used by pharmacists to check the prescription orders created by doctors.
- **Picture Archiving & Communication System** - Medical imaging technology, which stores and provides access to patient images.
- **Radiology IS** - Assists radiology services in storing, manipulation and retrieving information.

5.1.1 Hospital Big Picture

Since we are leading a non-real case scenario, the information systems presented here were gathered from various documentation sources related to security issues in Hospital Information System (HIS) [31] [32]. The view does not simulate all accesses in Business Processes related with a Hospital environment, keeping the focus on core Business Processes and some support Business Processes.

The access control Hospital Big Picture created using Atlas tool is displayed in figure 5.2, where it is possible to analyze all entities that Business Actors can access in the hospital environment. Each arrow starts at Business Actor and finishes at an entity accessed by Business Actors. Figure 5.2 displays all accesses of all Business Actor, but it is possible to filter the accesses arrows to get a more clean view and analysis with 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). The example in figure 5.3 displays all Business Actors that can access the Admission Form and all entities that George can access.

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 A.1.

5.1.2 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 5.4 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.

Hospital Business Actors VS Applications ACL view was created with graphical queries relevant to this view. The queries can be check in figure A.4.

5.2 Case Study 2

This case study is related with access controls in an ERP system. ERPs are used by many companies around the world to manage and integrate different parts of their business. Many ERPs consists of different modules such as finance, human resources, sales, purchasing, planning, production, etc... which represents the different parts of business on the organization. These kind of systems must evolve to follow and fulfill the business needs and can be ineffective if a company does not implement properly the requirements.

One of the main concerns regarding access controls in the Information Systems is to ensure that users have the minimum privileges necessary to perform their tasks, and in the ERPs this is not an exception. Users, after authentication, can access different modules within ERPs but always in alignment with their functions/roles, and/or organization policies.

The ERP case study considers ACMS as an external entity of the Information System. The policies used in this case study are shown below:

- Sales Director can read Sales Reports, Quotations, Sales Order Documents, Delivery Documents, and Billing Documents.
- Recruiter can read CVs and Offer Proposals.
- Payroll Specialist can read Payslips.
- Maintenance Operator can read Maintenance Order.
- Melissa can update Purchasing Order Documents.

- Kevin can create Planning Orders.
- Donald can create Production Orders.
- Richard can create Maintenance Orders.
- Business Actors assigned to the Training Course Business Process can read Training Courses Business Objects.
- Business Actors assigned to the Campaigns Business Process can create Advertising Business Objects.
- Business Actors assigned to the Payment Processing can create the Payment Documents.
- Business Actors assigned to Procurement can access Purchase Order Documents and Vendor Contracts.
- HR Director has the same privileges of Business Actors in its department.
- CFO has the same privileges of Business Actors in its department.
- Procurement Manager has the same privileges of Business Actors in its department.

The above policies are applied to the different modules within the ERP. Our ERP case study considers the following modules:

- Human Resources
- Sales & Marketing
- Purchasing
- Planning & Production
- Accounting & Finance
- Maintenance
- Inventory Management

For this case study, a new board and ACL views were created with the entities needed to analyze the accesses in ERP.

5.2.1 ERP Big Picture

Business Actors have the same accesses to Business Objects in the Business Layer and the corresponding Data Objects in the Application Layer. For each Business Object, there is a Data Object with the same name which realizes that Business Object. Figure 5.5 displays all accesses for Business Actors in terms of Business Objects and Data Objects. The remaining elements in the board view behind Business Actors, Business Objects, and Data Objects are entities used in policies definition which belong to Archimate Framework [8].

Figure 5.6 is a board view with filtered accesses, where it is possible to analyze isolated cases in terms of access control for one Business Actor and one Business Object/Data Object. The example of figure 5.6 displays all accesses allowed for Business Actor Scott and all Business Actors that have permissions for Payment Document Business Object/Data Object. This example allows to analyze the accesses from two different perspectives - active entity perspective, where is possible to analyze the accesses allowed to Business Actors, and passive entity perspective, where is possible to analyze all the Business Actors with access to a specific Business Object or Data Object.

Figure A.3 contains all graphical queries used to implement the policies in this case study. Queries that support the policies in this case study can be derived from subject's role, subject's business process, or the subject's identification (e.g. subject name). **This case study also presents complex queries which are queries using other queries. This kind of approach enables query reuse, abstraction, and facilitates the work of policy administrator.**

5.2.2 ERP Business Actors VS Applications

This view intends to help the access control administrator in the identification of Data Objects that can be access by a Business Actor in each Application Component. In the ERP case study, each Application Component represents a module or a sub module inside the ERP. Figure 5.7 reflects all accesses of Business Actor to Data Objects by Application Component. Each cell of the Atlas ACL executes the graphical queries which reflects the relevant policies for this view. In each non-blank cell of figure 5.7, we can click over to see in a pop-up the list of Data Objects that each Business Actors can access by Application Component. The *ERP Business Actors VS Application* ACL view was created using graphical queries relevant to identify the accesses of Business Actors. The queries can be check in figure A.5.

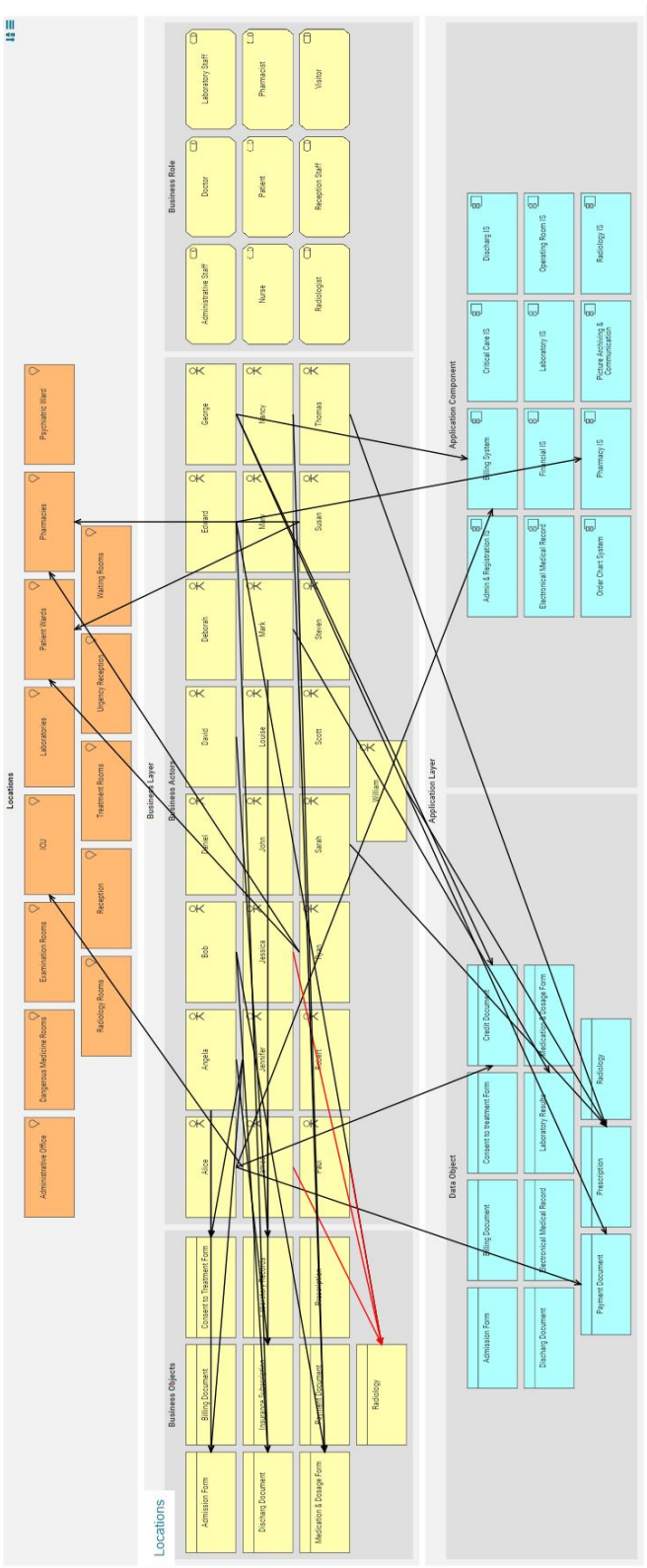

Figure 5.2: Board View with all accesses allowed for business actors in Hospital case study

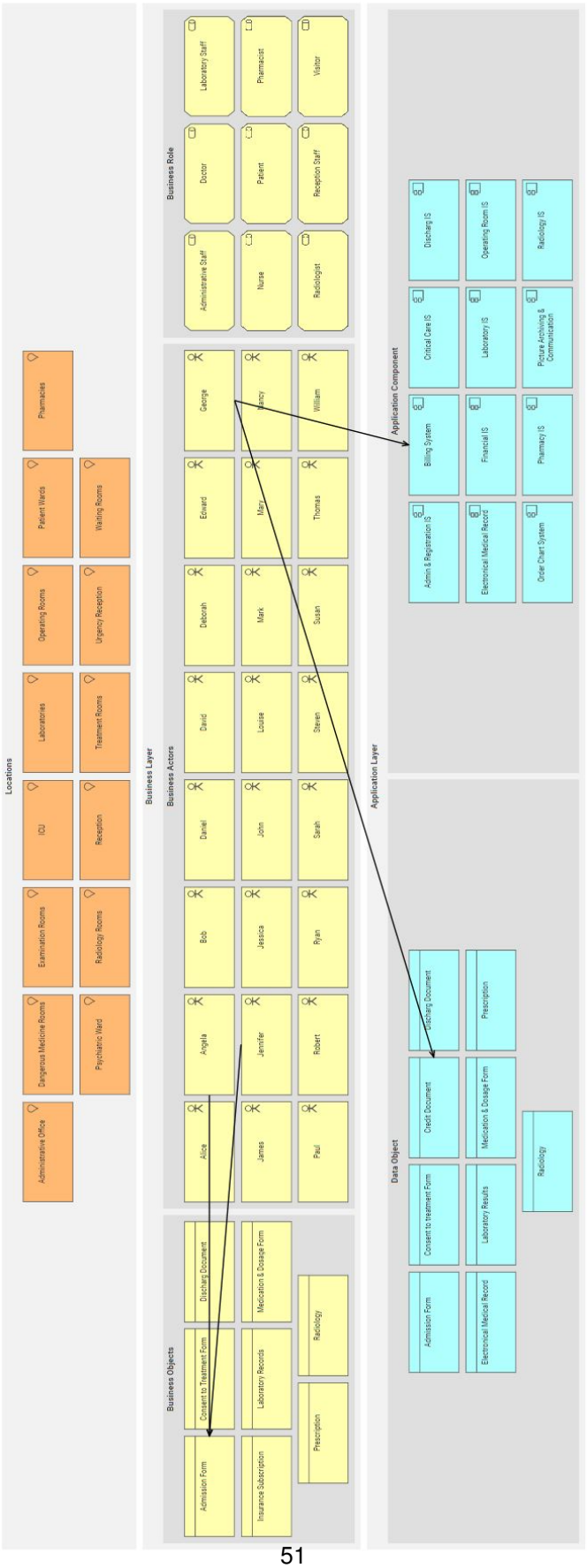

Figure 5.3: Board View with filtered accesses in Hospital case study

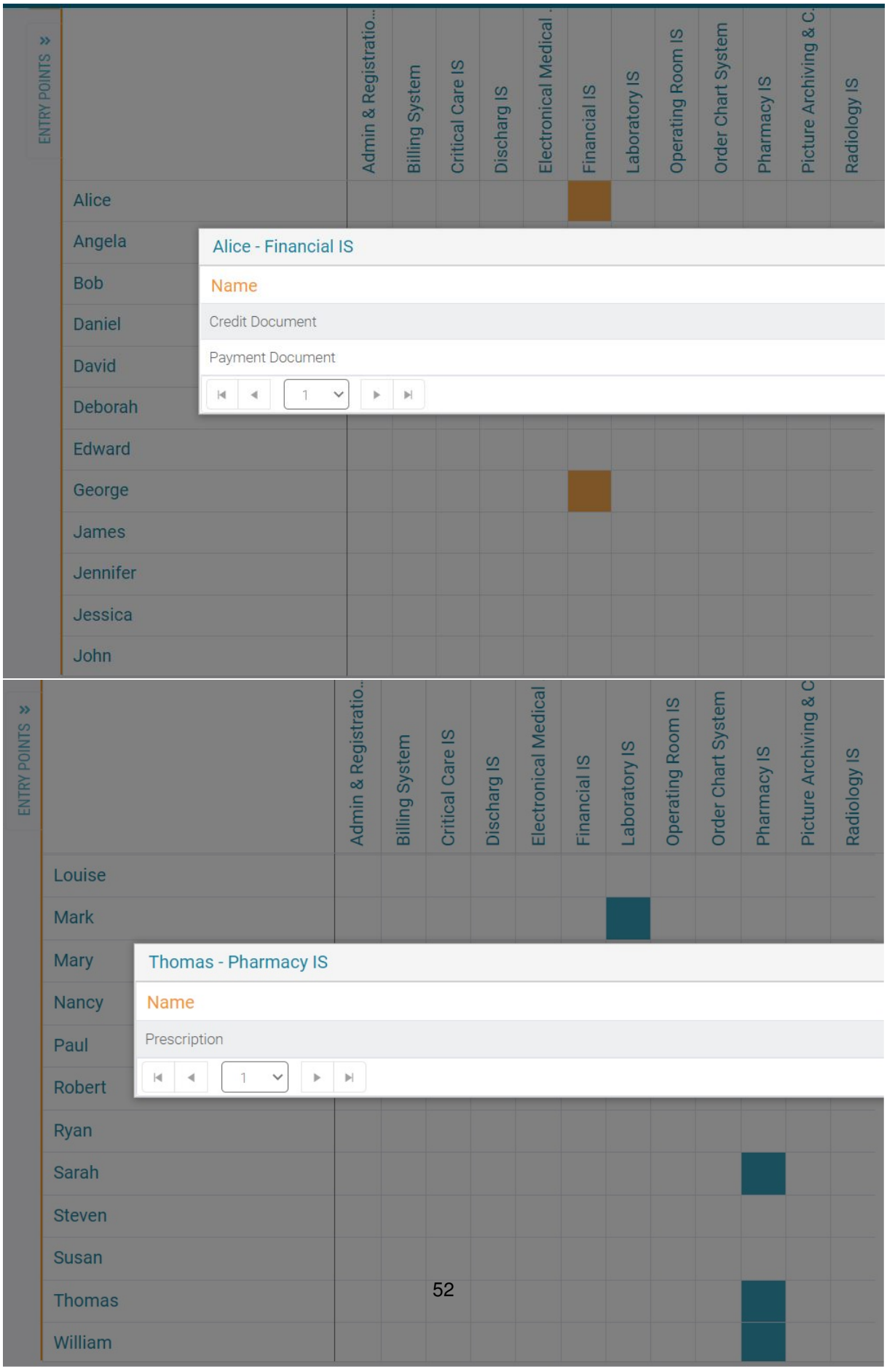

Figure 5.4: Hospital Business Actors VS Application View

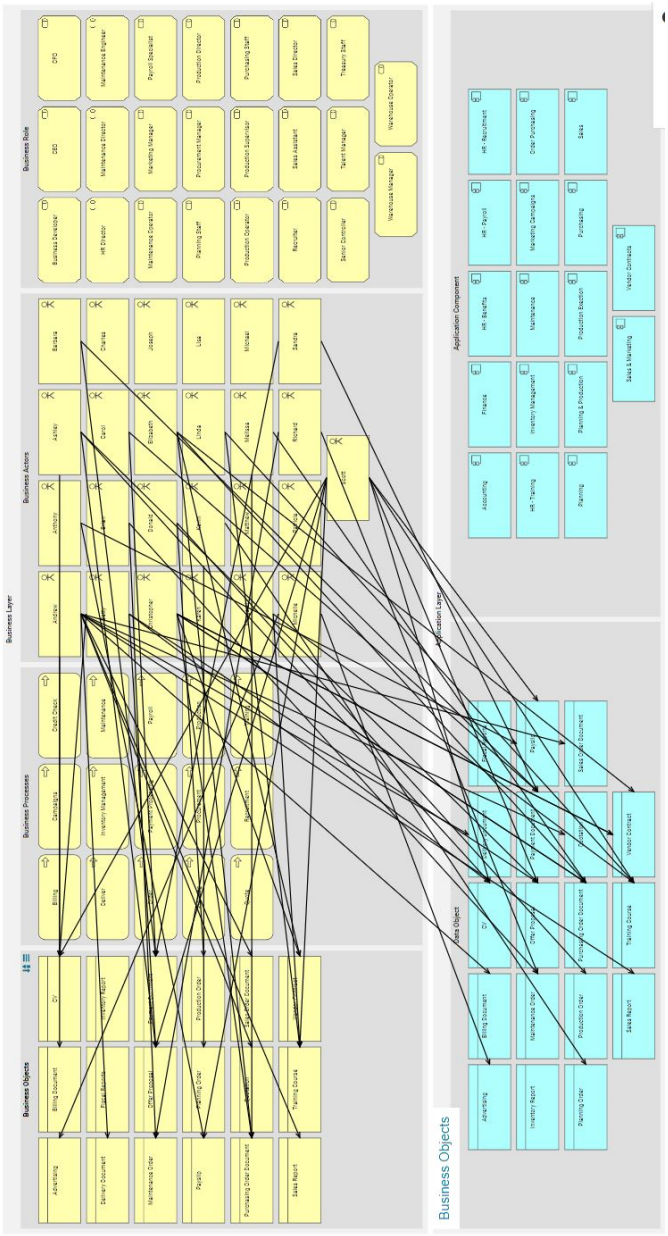

Figure 5.5: Board View with all accesses allowed for business actors in ERP case study

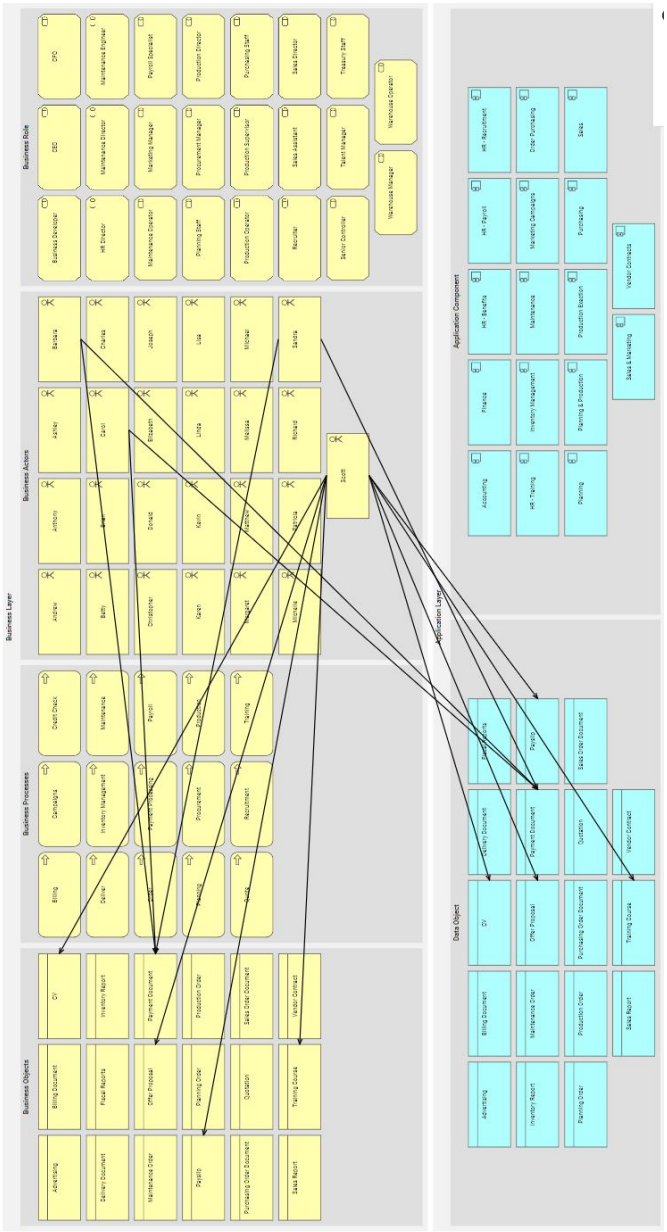

Figure 5.6: Board View with filtered accesses in ERP case study

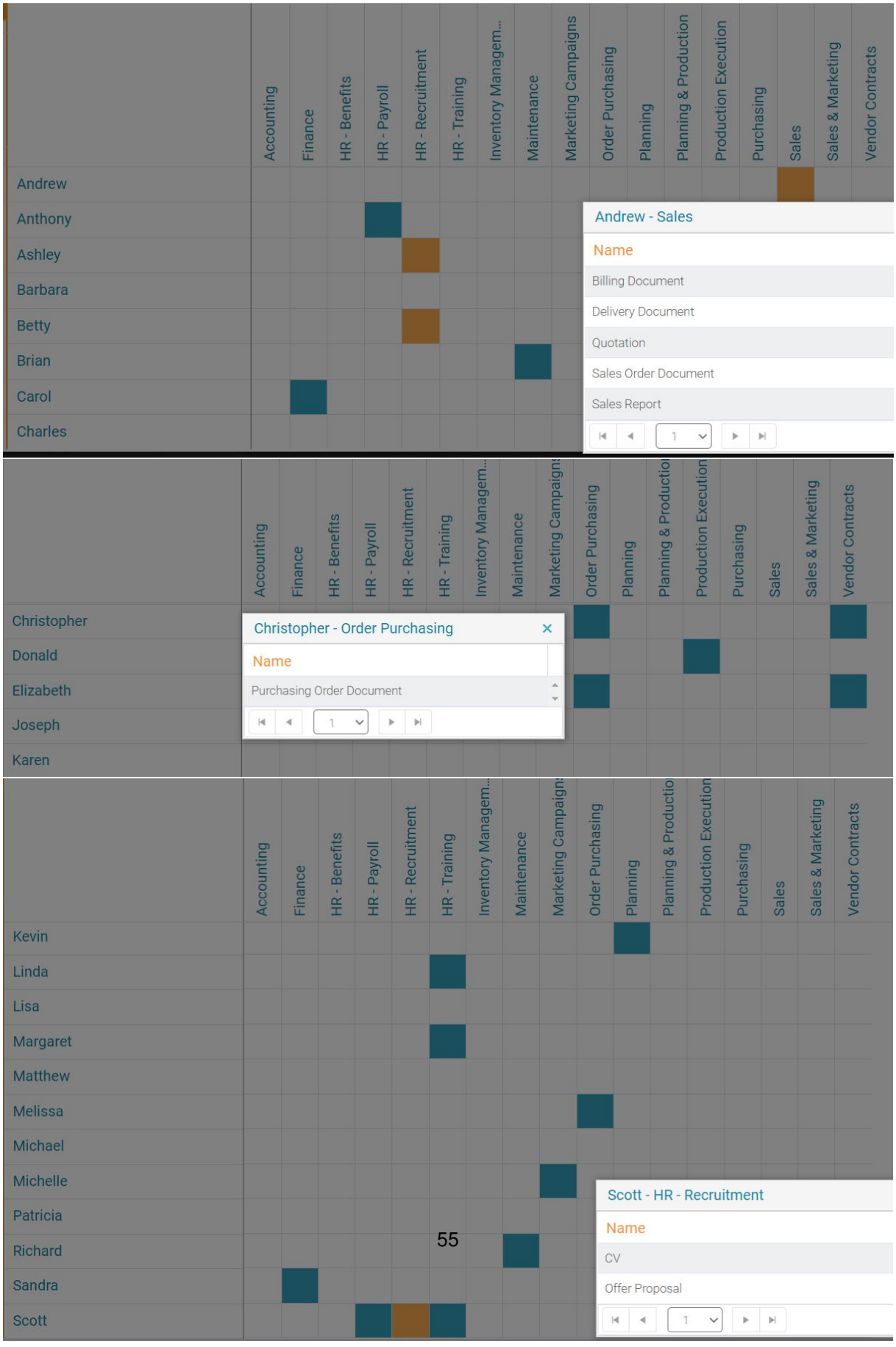

Figure 5.7: ERP Business Actors VS Application View

6

Evaluation

Contents

This chapter represents the 5th chapter of DSRM: Evaluation [6]. This chapter is responsible to analyze how successful the solution artifact proposed in chapter 4 is in solving the problem described in chapter 1.2. This chapter aims to discuss the demonstration results comparing them with the research goals presented.

6.1 Evaluation Methods

The solution proposed in this research work was demonstrated using board and ACL views in the Atlas tool [2] with the two different case studies (Hospital and ERP).

Our solution is evaluated in this chapter using the artifact evaluation proposed by Prat et al. [33] that helps Design Science Research (DSR) researchers, providing a holistic view of artifact evaluation. Prat et al. [33] 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"*.

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 research work evaluation.The definitions provided are present in Prat et al. work [34].

- **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.If the board and ACL views for access control analysis change with this approach, we will consider our work validated. 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.

- **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.

- **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.
- **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 [7] framework core elements. Each element represents an entity throughout enterprise structure. on the other hand, ABAC provides the necessaries flexibility and fine-grain approach enough for organizations' policy definition.

The solution artifact will be considered successful if the views generated can clearly display all users' access within the organization. Accesses in the views must change when the entity attributes change or when the policies change. Our solution must also reduce maintenance costs for the access control administrator.

6.2 Demonstration

In this section, we present the evaluation and validation of this research work. During the evaluation, we will use Hospital Case Study to apply the evaluation criteria and evaluation methods previously presented in this chapter.

Validity We start by changing the attribute values of entities in the Data Explorer of Atlas tool. We then select a query which is using the specific attributes changed. When the attributes relevant for the query are changed, this will be reflected in the board and ACL views of access control. The views are regenerated and the differences between the changes to the attributes before and after the change for a specific Business Actor are compared. The equivalent access control changes must be reflected in both views after its refresh.

Analysing the Data Explorer of Atlas tool [2], we can verify that Alice is located at *”Administrative Office”* like displayed in figure 6.2. Looking for the policies defined to Hospital case study, we can verify that policy *”Staff working in Administrative Office can change all documents in Financial IS”* gives access to Alice for Credit Document due the assignment between Financial IS as displayed in figure 6.1. Before the changes of Alice's Location attribute, she had access to Credit Documents, as we can confirm in Big Picture View (figure 6.3) and ACL View (figure 6.4)

We then changed the Alice's location from Administrative Office to Reception like displayed in figure 6.5. The changes were saved and the views were refreshed. After the refresh, we can check that Alice after location change lost the accesses to Credit Documents. The figures 6.6 and 6.7 display the Business Actors with access to Credit Documents. We can verify that now only George has access Credit Documents.

Using this practical example in Atlas, we verified that the artifact works correctly.

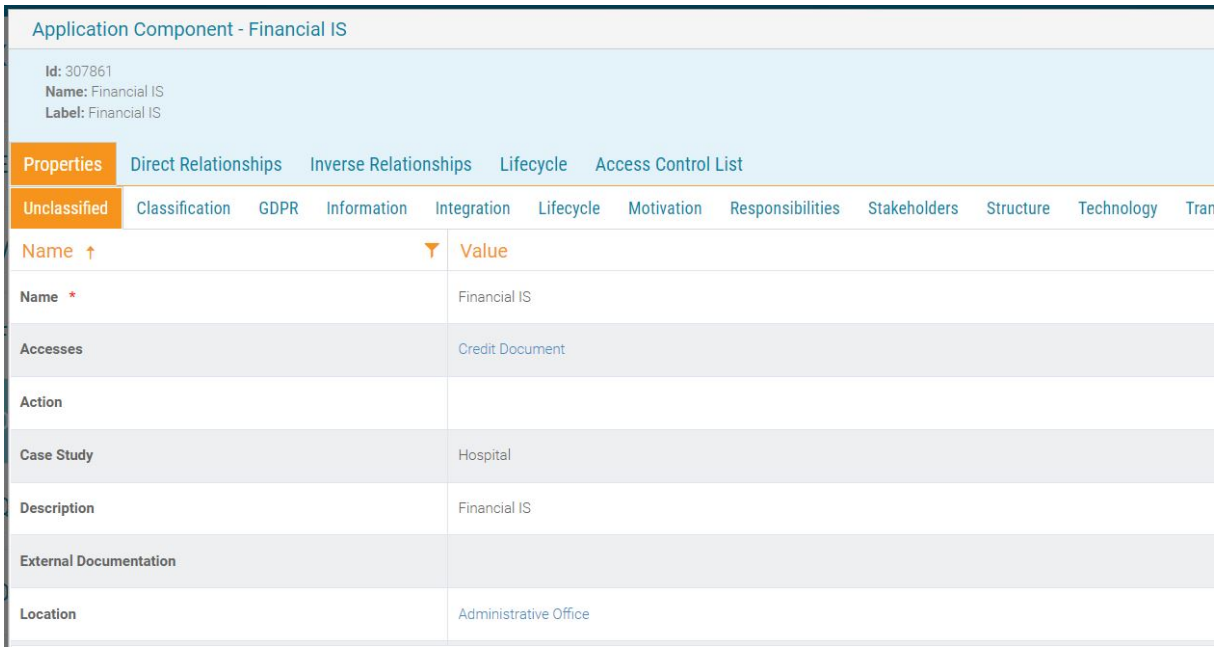

Figure 6.1: Financial IS attributes

Regarding **Technical Feasibility** we interviewed an expert in access control management of Information Systems to gathers its feedback about the practical implementation of our solution using the Atlas [2] tool. 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. Both cases studies were also presented and the specialist was encouraged to implement a query *"George(Administrative Staff) can access Administrative Office."* reflecting a new policy in the Hospital Case Study. Looking for the graphical queries already created with similar purpose, the interviewed internalized the main concepts and produced the desired outcome.

The feedback from the professionals 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 will be 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 (e.g. Subject has a new role assigned). The main data relevant for case studies was uploaded using files during our solution demonstration and evaluation. The screenshots of the Excel files with relevant data uploaded are available in the Appendix B.

Completeness as already mentioned during this chapter, is provided by Archimate [7] framework

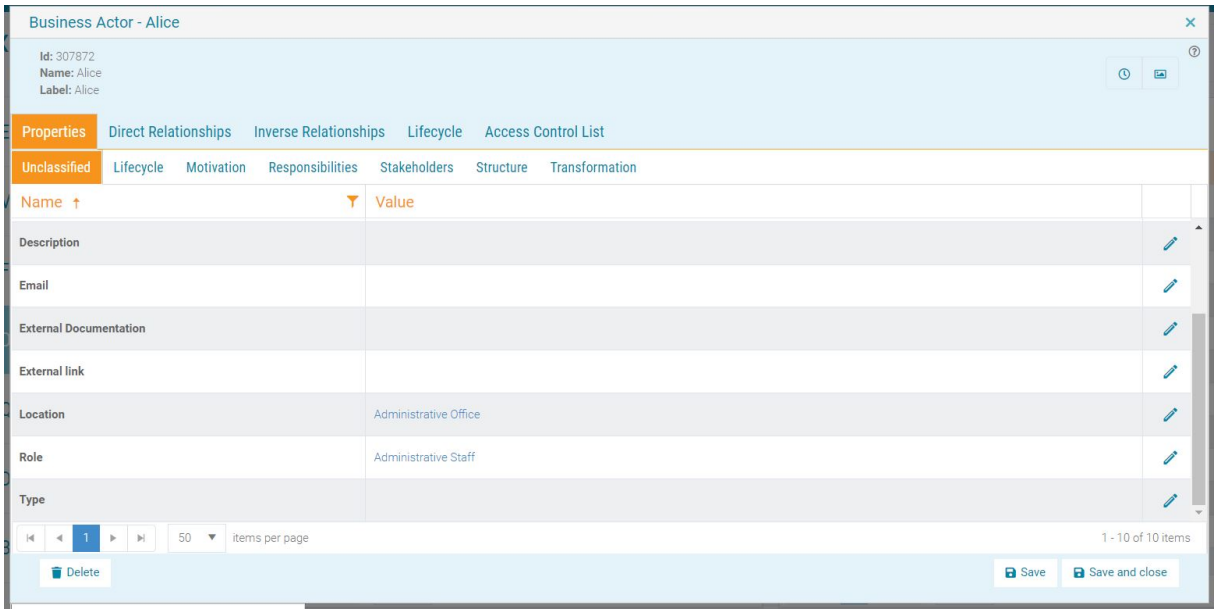

Figure 6.2: Alice's Attributes before change

and their elements and relationships. 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. **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.**

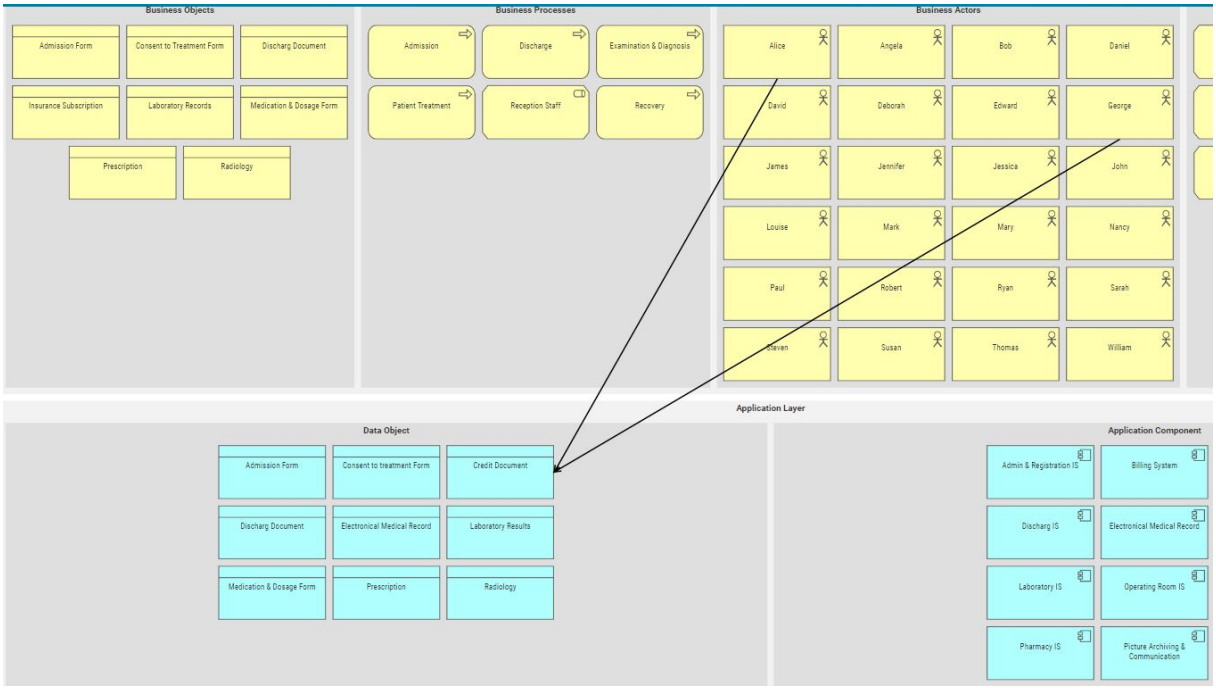

Figure 6.3: Alice Business Actor accesses before attributes' change

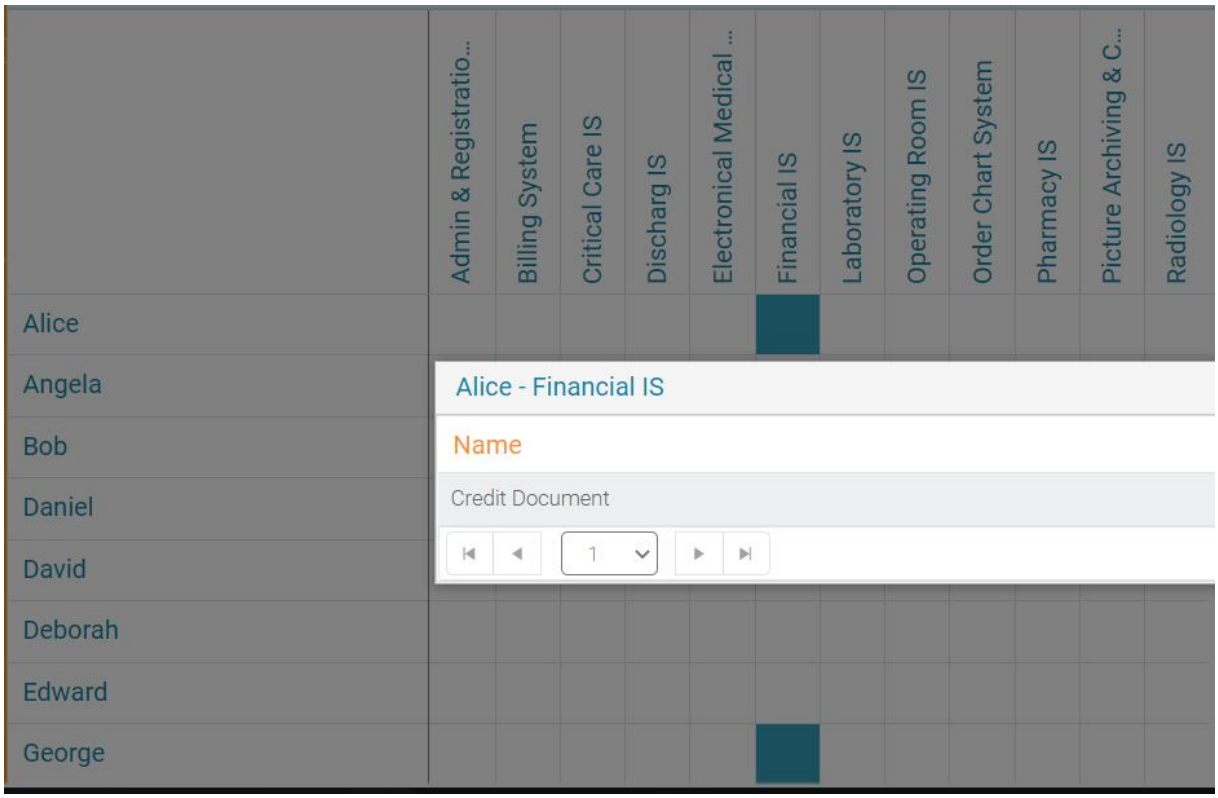

Figure 6.4: Alice Business Actor accesses before attributes' change

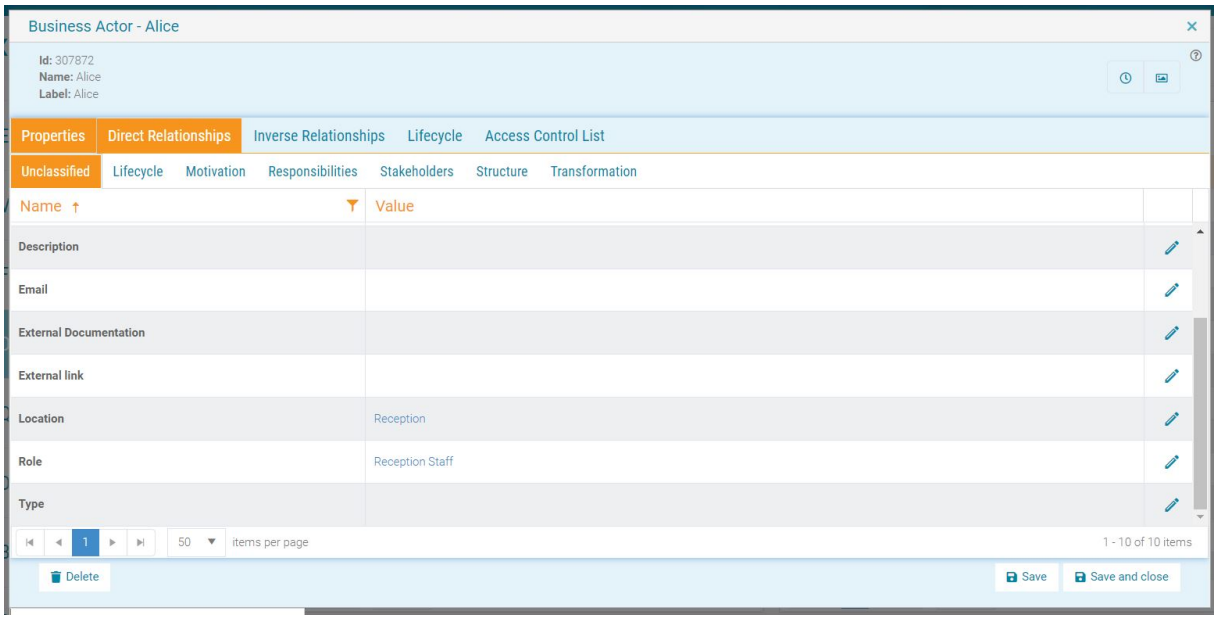

Figure 6.5: Alice's Attributes after change

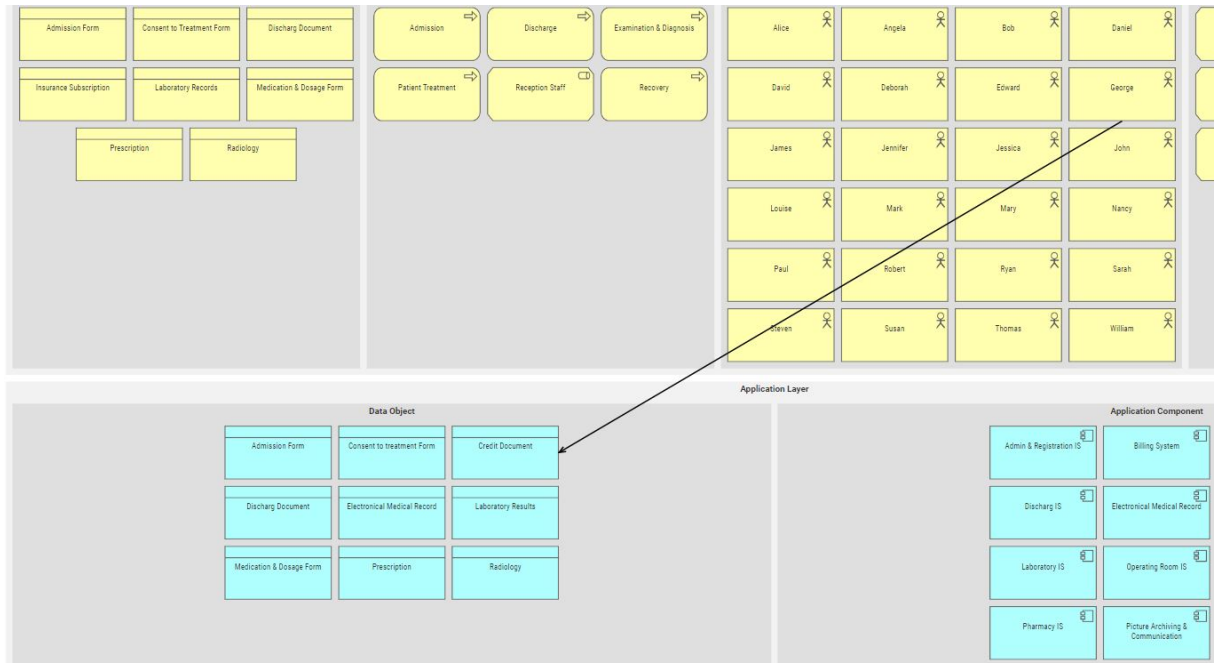

Figure 6.6: Alice Business Actor accesses after attributes' change

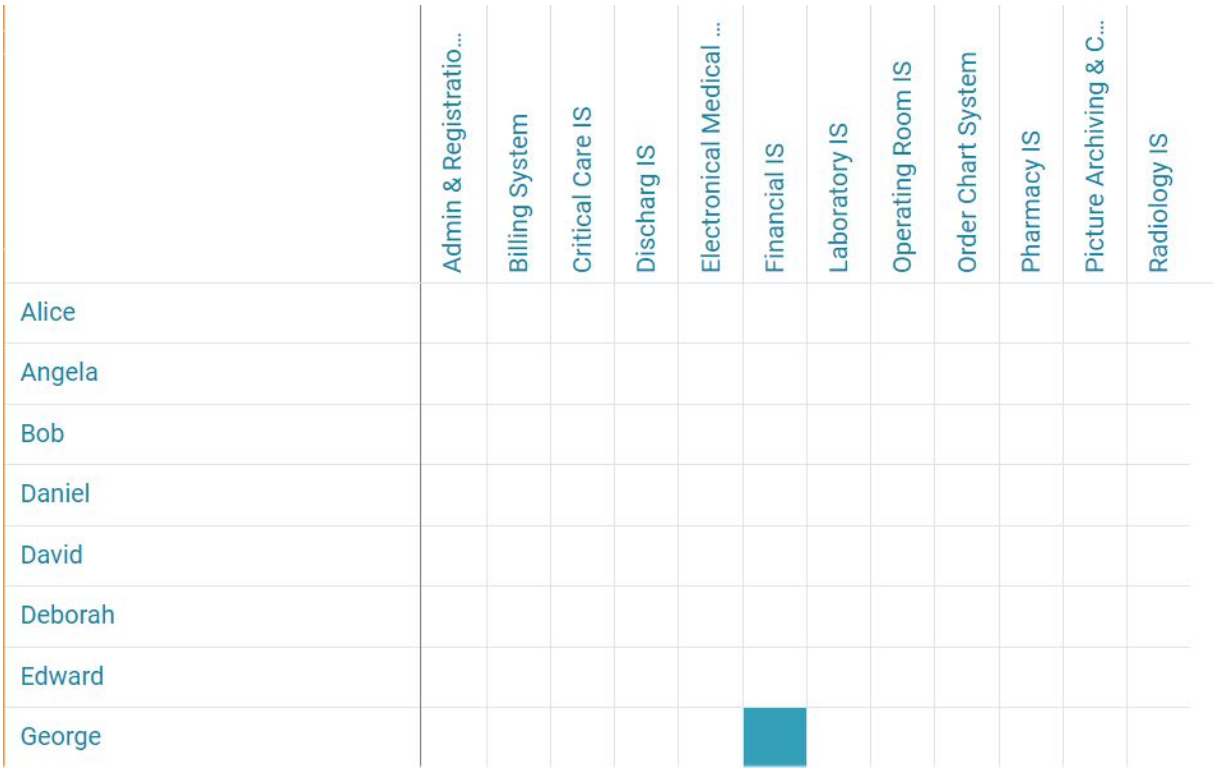

Figure 6.7: Alice Business Actor accesses after attributes' change

7

Conclusion

Contents

In this chapter, we present a final review of our work, mentioning contributions, limitations and what we identified as a possible next steps regarding the future work about this topic. Our work was developed using the DSRM [6] methodology.

7.1 Contributions

Security starts by the definition of suitable access control policies within an organization. The problem this work 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 orchestrated 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 orchestrated system, the relationship of enterprise elements in terms of access control.**

The views developed in this research work 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 work is a merge between the flexible access control model ABAC [22] and the enterprise architecture framework Archimate [7].

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
- Orchestrated 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 orchestrated 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 [8] 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 research work were accomplished, since in both illustrative scenarios were possible to demonstrate and evaluate the access control model proposed in our solution.

7.2 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.

7.3 Future Work

After the development this research work, 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.

Bibliography

[1] K. Gaaloul and H. Proper, "An access control model for organisational management in enterprise architecture," *2013 Ninth International Conference on Semantics, Knowledge and Grids*, pp. 37–43, 2013.

[2] P. Sousa, R. Leal, and A. Sampaio, "Atlas: the enterprise cartography tool," *Proceedings of 8th the Enterprise Engineering Working Conference Forum*, vol. 2229, 2018.

[3] D. Ferraiolo, R. Sandhu, and et al, "Proposed NIST Standard for Role-Based Access Control," *National Institute of Standards and Tecnhology*, vol. 4, no. 3, pp. 224–274, August 2001.

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

[5] D. Servos and S. Osborn, "Current research and open problems in attribute-based access control," *ACM Computing Surveys*, vol. 49, no. 4, January 2017.

[6] K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.

[7] "Archimate® 3.1 specification," *The Open Group Standard*, November 2019.

[8] A. Josey, M. Lankhorst, I. Band, H. Jonkers, and D. Quartel, "An introduction to the archimate® 3.0 specification," *The Open Group*, June 2016.

[9] "A guide to understanding discretionary access control in trusted systems," *National Computer Security Center, Fort George G. Meade, Maryland*, September 1987.

[10] B. Lampson, "Protection," *5th Princeton Conference on Information Sciences and Systems*, pp. 437–443, March 1971.

[11] G. Graham and J. Denning, “Protection - principles and practice,” *Proc. Spring Joint Computer Conference*, pp. 417–429, May 1972.

[12] M. Harrison, W. Ruzzo, and J. Ullman, “Protection in operating systems,” *Comm. Association for Computing Machinery*, vol. 29, no. 2, pp. 461–471, August 1976.

[13] D. Bell and L. LaPadula, “Secure computer systems: Unified exposition and multics interpretation,” *The Mitre Corporation*, March 1976.

[14] K. Biba, “Integrity considerations for secure computer systems,” *The Mitre Corporation*, April 1977.

[15] “Department of defense trusted computer systems evaluation criteria,” *Department of Defense (DoD) National Computer Security Center*, August 1983.

[16] D. Clark and D. Wilson, “A comparison of commercial and military computer security policies,” *IEEE Symposium of Security and Privacy*, pp. 184–194, April 1987.

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

[18] R. Sandhu and et al, “Role-based access control models,” *Computer*, vol. 29, no. 2, pp. 38–47, February 1996.

[19] E.Yuan and J. Tong, “Attributed based access control (abac) for web services,” *Proceedings of the IEEE International Conference on Web Services (ICWS’05)*, 2005.

[20] X.Jin, R. Krishnan, and R. Sandhu, “A unified attribute-based access control model covering dac, mac and rbac,” *DBSec: IFIP Annual Conference on Data and Applications Security and Privacy*, pp. 41–55, 2012.

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

[22] V. Hu, D. Kuhn, and D. Ferraiolo, “Access control for emerging distributed systems,” *National Institute of Standards and Technology*, vol. 51, pp. 100–103, October 2018.

[23] P.Biswas, R. Sandhu, and R. Krishnan, “Label-based access control: an abac model with enumerated authorization policy,” *Proc. 2016 Association for Computing Machinery International Workshop on Attribute Based Access Control*, pp. 1–12, 2016.

[24] B. Parducci and H. Lockhart, “extensible access control markup language (xacml) version 3.0,” OASIS, Tech. Rep., January 2013. [Online]. Available: https://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf

[25] D. Ferraiolo, R. Chandramouli, R. Kuhn, and V. Hu, “A comparison of attribute based access control (abac) standards for data serviceapplications. nist special publication 800-178,” *National Institute of Standards and Technology*, October 2016.

[26] M. Al-Kahtani and et al, “A model for attribute-based user role assignment,” *Annual Computer Security Applications Conference*, pp. 353–362, 2002.

[27] R. Kuhn, E. Coyne, and T. Weil, “Adding attributes to role-based access control,” *IEEE Computer*, vol. 43, no. 6, pp. 79–81, June 2010.

[28] A. Kern and C. Walhorn, “Rule support for role-based access control,” *Proc. 10th Association for Computing Machinery Symposium on Access Control Models and Technologies*, pp. 130–138, June 2005.

[29] R. Pilipchuck, S. Seifermann, and R. Heinrich, “Aligning business process access control policies with enterprise architecture,” *CEEC, Ljubljana, Slovenia*, November 2018.

[30] K. Gaaloul, S. Guerreiro, and H. Proper, “Modeling access control transactions in enterprise architecture,” *IEEE 16th Conference on Business Informatics*, 2014.

[31] M. Masrom and A. Rahimly, “Overview of data security issues in hospital information systems,” *Pacific Asia Journal of the Association for Information Systems*, vol. 7, no. 4, pp. 51–66, December 2015.

[32] M. Masrom and et al, “Activity-oriented access control to ubiquitous hospital information and services,” *Information Sciences*, pp. 2979–2990, 2010.

[33] N. Prat and et al, “Artifact evaluation in information systems design - science research - a holist view,” *Pacific Asia Conference on Information Systems*, 2014.

[34] ——, “A taxonomy of evaluation methods for information systems artifacts,” *Journal of Management Information Systems*, vol. 32, no. 3, pp. 229–267, 2015.

Blueprints & Queries

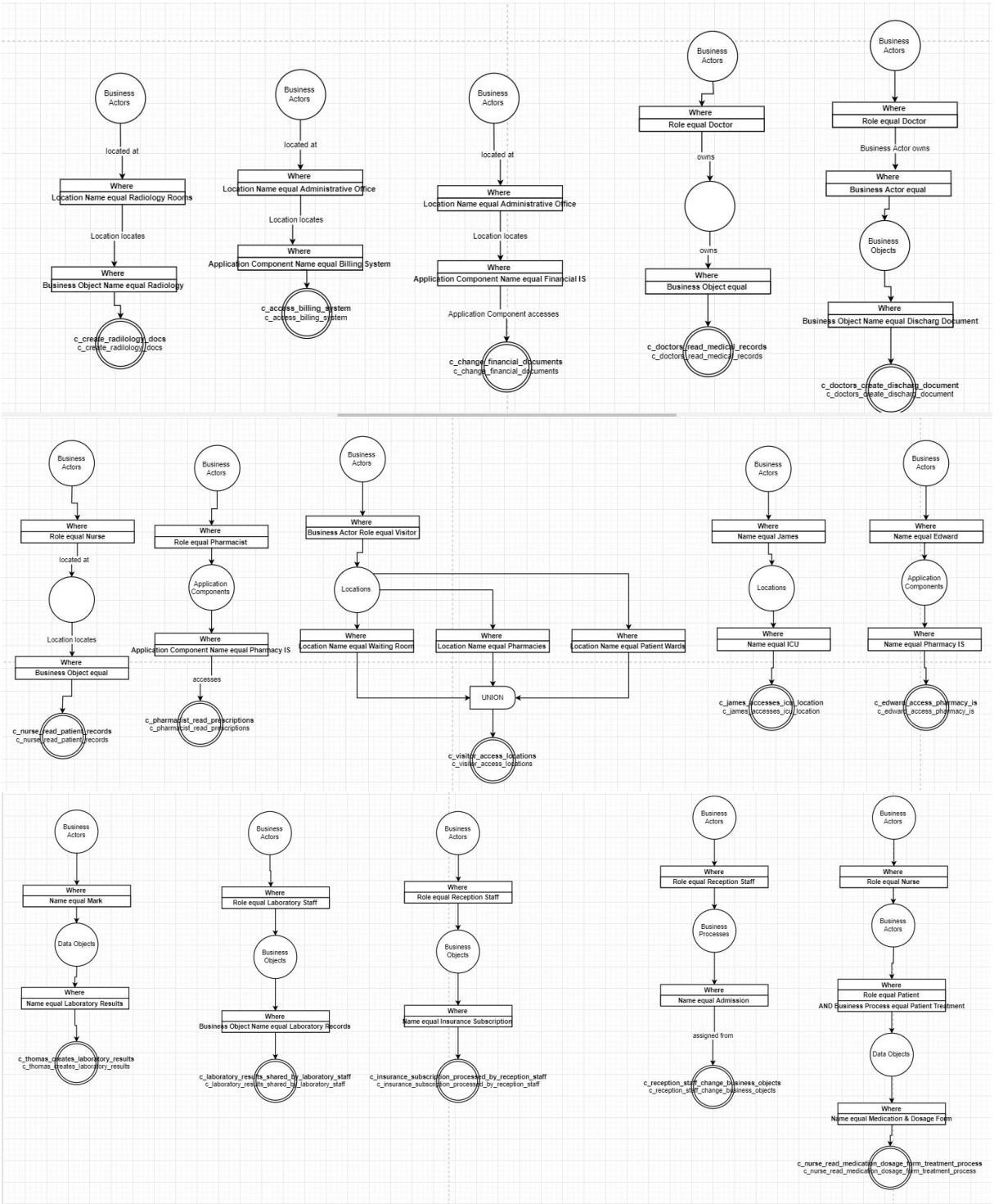

Figure A.1: Queries applying the Hospitals' policies case study in "Big Picture" View

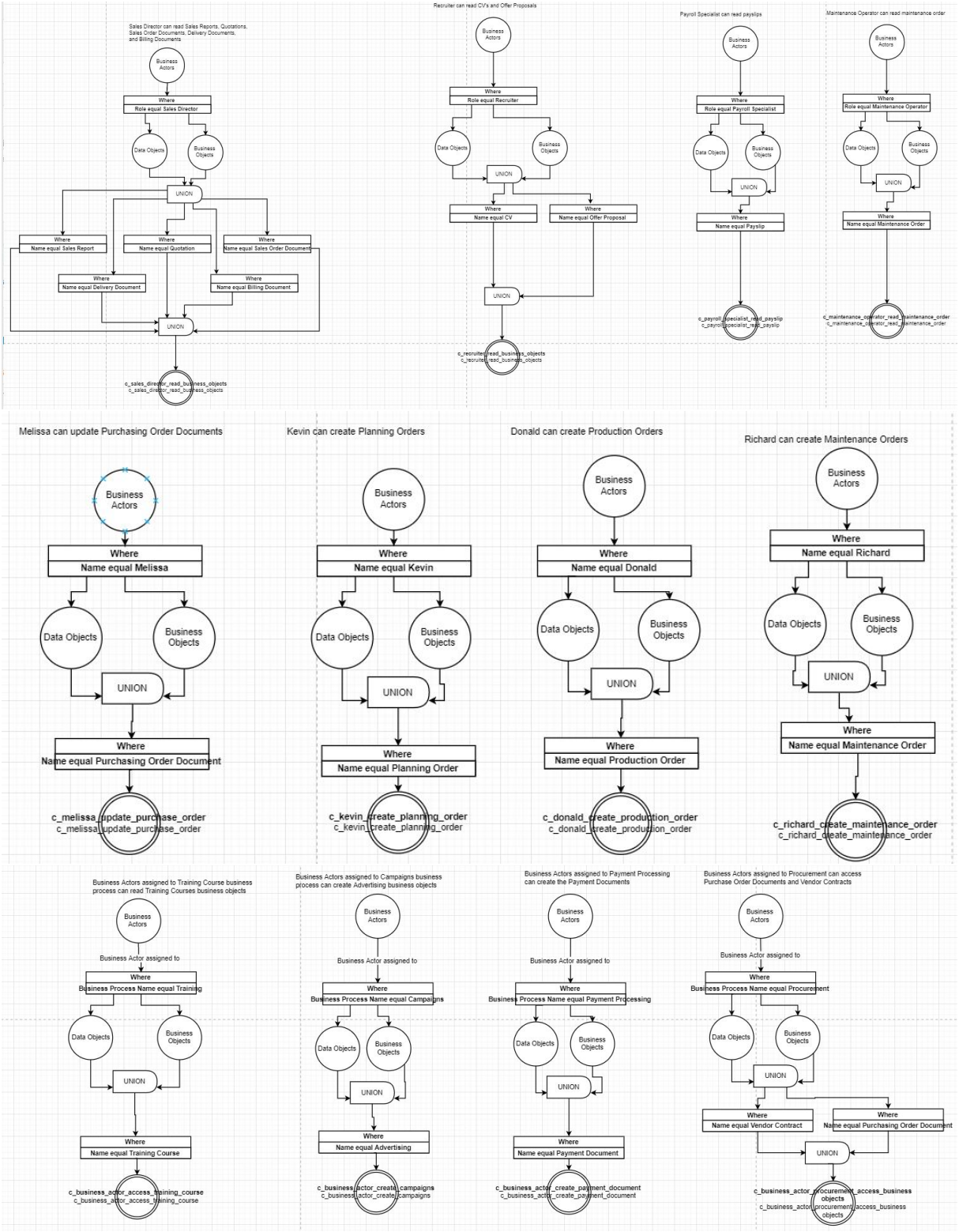

Figure A.2: Queries applying the ERPs' policies case study in "Big Picture" view

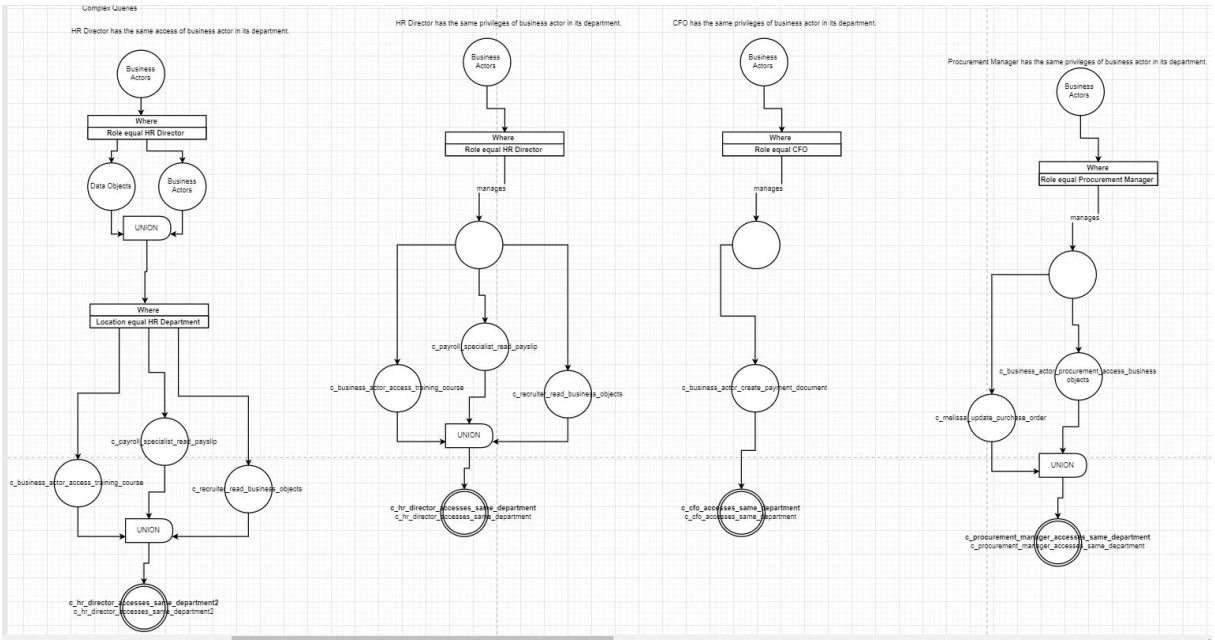

Figure A.3: Queries applying the ERPs' policies case study in "Big Picture" view

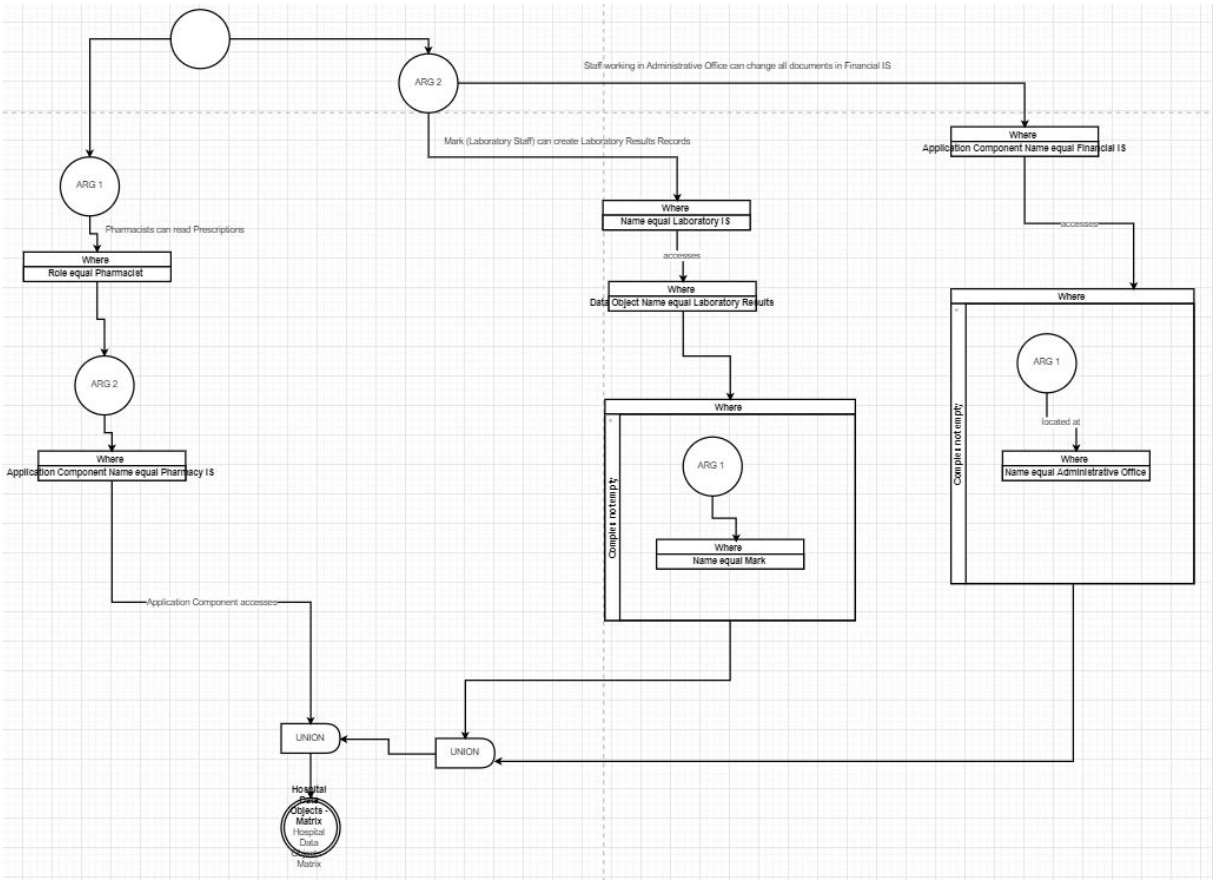

Figure A.4: Queries applying the Hospitals' policies case study in "Business Actors VS Application" view

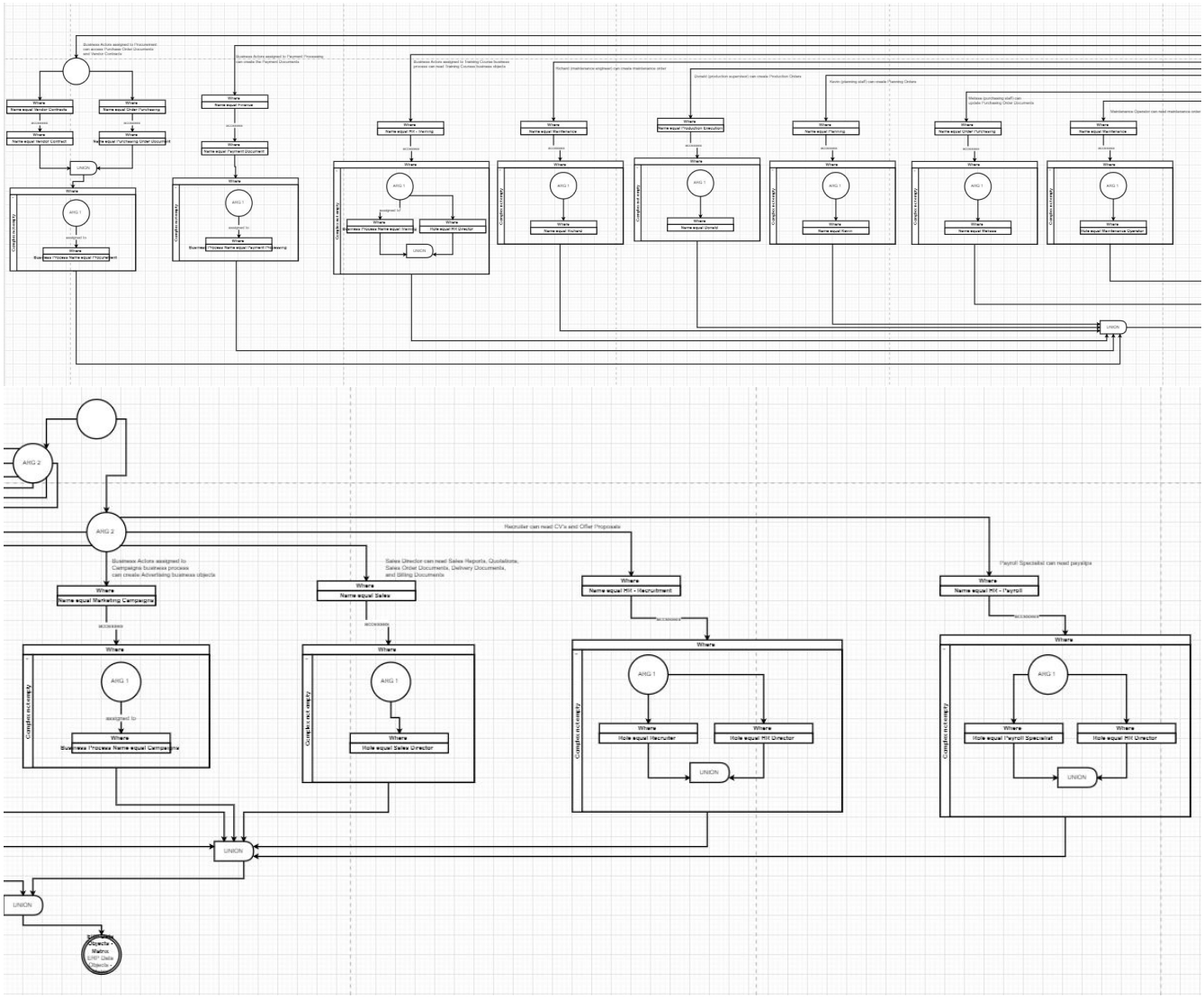

Figure A.5: Queries applying the ERPs' policies case study in "Business Actors VS Application" view

B

Data Uploaded

	A	C	E	S	T	V	Y
1	Name	Business Process	Case Stu	Location	Manage	Own	Role
2	Alice		Hospital	Administrative Office			Administrative Staff
3	Andrew		ERP	Sales Department			Sales Director
4	Angela		Hospital	Reception			Reception Staff
5	Anthony		ERP	HR Department	Scott		Payroll Specialist
6	Ashley		ERP	HR Department	Scott		Recruiter
7	Barbara		ERP	Finance Department			CFO
8	Betty		ERP	HR Department	Scott		Recruiter
9	Bob		Hospital	Operating Rooms			Doctor
10	Brian		ERP	Maintenance Department	Joseph		Maintenance Operator
11	Carol	Payment Processing	ERP	Finance Department	Barbara		Tresuary Staff
12	Charles		ERP	Board			CEO
13	Christopher	Procurement	ERP	Logistc Department			Procurement Manager
14	Daniel	Admission	Hospital	Urgency Reception		Edward	Patient
15	David		Hospital	Laboratories			Laboratory Staff
16	Deborah	Patient Treatment	Hospital	Patient Wards		Bob	Patient
17	Donald		ERP	Production Department	Lisa		Production Supervisor
18	Edward		Hospital	ICU			Doctor
19	Elizabeth	Procurement	ERP	Logistc Department	Christopher		Purchasing Staff
20	George		Hospital	Administrative Office			Administrative Staff
21	James		Hospital	Radiology Rooms			Radiologist
22	Jennifer		Hospital	Urgency Reception			Reception Staff
23	Jessica		Hospital	Radiology Rooms			Radiologist
24	John		Hospital	ICU			Doctor
25	Joseph		ERP	Maintenance Department			Maintenance Director
26	Karen		ERP	Sales Department	Andrew		Business Developer
27	Kevin		ERP	Production Department	Lisa		Planning Staff
28	Linda	Training	ERP	HR Department	Scott		Talent Manager
29	Lisa		ERP	Production Department			Production Director
30	Louise		Hospital				Doctor

	A	C	E	S	T	V	Y
1	Name	Business Process	Case Stu	Location	Manage	Own	Role
31	Margaret	Training	ERP	Logistic Department	Christopher		Purchasing Staff
32	Mark		Hospital	Laboratories			Laboratory Staff
33	Mary	Patient Treatment	Hospital	Patient Wards			Nurse
34	Matthew		ERP	Sales Department	Andrew		Sales Assistent
35	Melissa		ERP	Logistic Department	Christopher		Purchasing Staff
36	Michael		ERP	Finance Department	Barbara		Senior Controller
37	Michelle	Campaigns	ERP	Marketing Department			Marketing Manager
38	Nancy	Patient Treatment	Hospital	Patient Wards			Nurse
39	Patricia		ERP	Logistic Department	Christopher		Warehouse Manager
40	Paul		Hospital	Radiology Rooms			Radiologist
41	Richard		ERP	Maintenance Department	Joseph		Maintenance Engineer
42	Robert	Examination & Diagnos	Hospital	Laboratories			Patient
43	Ryan		Hospital	Waiting Rooms			Visitor
44	Sandra	Payment Processing	ERP	Finance Department	Barbara		Tresuary Staff
45	Sarah		Hospital	Pharmacies			Pharmacist
46	Scott		ERP	HR Department			HR Director
47	Steven	Patient Treatment	Hospital	Patient Wards			Patient
48	Susan		Hospital	Waiting Rooms			Visitor
49	Thomas		Hospital	Pharmacies			Pharmacist
50	William		Hospital	Pharmacies			Pharmacist

Figure B.1: Business Actors' File Data

	A	F
1	Name	Case Study C
2	Administrative Office	Hospital
3	Board	ERP
4	Dangerous Medicine Rooms	Hospital
5	Examination Rooms	Hospital
6	Finance Department	ERP
7	HR Department	ERP
8	ICU	Hospital
9	Laboratories	Hospital
10	Logistic Department	ERP
11	Maintenance Department	ERP
12	Marketing Department	ERP
13	Operating Rooms	Hospital
14	Patient Wards	Hospital
15	Pharmacies	Hospital
16	Production Department	ERP
17	Psychiatric Ward	Hospital
18	Radiology Rooms	Hospital
19	Reception	Hospital
20	Sales Department	ERP
21	Treatment Rooms	Hospital
22	Urgency Reception	Hospital
23	Waiting Rooms	Hospital

Figure B.2: Locations' File Data

	A	B	C	F	I	U	AD
1	Name	Accesses	Action	Case Study	Components	Description	Location
2	Accounting			ERP			
3	Admin & Registration IS	Admission FormConsent to treatment Form		Hospital		Admission & Registration IS	Reception
4	Billing System	Billing Document		Hospital		Billing System	Administrative Office
5	Critical Care IS	Electronical Medical RecordMedication & Dosage Form		Hospital		Critical Care IS	ICU
6	Discharg IS	Discharg Document		Hospital		Discharge Information System	Patient Wards
7	Electronical Medical Record	Electronical Medical RecordMedication & Dosage Form		Hospital		Electronical Medical Record	ICUExamination RoomsLaboratoriesPatient WardsPsychiatric WardRadiology RoomsTreatment Rooms
8	Finance			ERP			
9	Financial IS	Credit DocumentPayment Document		Hospital		Financial IS	Administrative Office
10	HR - Benefits			ERP			
11	HR - Payroll			ERP			
12	HR - Recruitment			ERP			
13	HR - Training			ERP			
14	Inventory Management			ERP			
15	Laboratory IS	Laboratory Results		Hospital		Laboratory IS	Laboratories
16	Maintenance			ERP			
17	Marketing Campaigns			ERP	Sales		
18	Operating Room IS	Electronical Medical Record		Hospital		Operating Room IS	Operating Rooms
19	Order Chart System	Medication & Dosage Form		Hospital		Order Chart System	ICUExamination RoomsLaboratoriesPatient WardsPsychiatric WardRadiology RoomsTreatment Rooms
20	Order Purchasing			ERP			
21	Pharmacy IS	Prescription	Access	Hospital		Pharmacy IS	Pharmacies
22	Picture Archiving & Communicatic	RadiologyLaboratory Results		Hospital		Picture Archiving and Communication Syste	ICUExamination RoomsLaboratoriesPatient WardsPsychiatric WardRadiology RoomsTreatment Rooms
23	Planning			ERP			
24	Planning & Production			ERP	PlanningProduction Execution		
25	Production Exection			ERP			
26	Production Execution						
27	Purchasing			ERP	Order PurchasingVendor Contracts		
28	Radiology IS	Radiology		Hospital		Radiology IS	Radiology Rooms
29	Sales			ERP			
30	Sales & Marketing			ERP			
31	Vendor Contracts			ERP			

Figure B.3: Application Components' File Data

	A	E	G	T	W
1	Name	Business Process	Case Study	Location	Owner
2	Admission Form	Admission	Hospital		
3	Advertising		ERP		
4	Billing Document		ERP	Administrative Office	
5	CV		ERP		
6	Consent to Treatment Form	Admission	Hospital		
7	Delivery Document		ERP		
8	Discharg Document		Hospital		Edward
9	Fiscal Reports		ERP		
10	Insurance Subscription		Hospital		
11	Inventory Report		ERP		
12	Laboratory Records		Hospital		
13	Maintenance Order		ERP		
14	Medication & Dosage Form		Hospital	Patient WardsPsychiatric Ward	Deborah
15	Offer Proposal		ERP		
16	Payment Document		ERP		
17	Payslip		ERP		
18	Planning Order		ERP		
19	Prescription		Hospital		
20	Production Order		ERP		
21	Purchasing Order Document		ERP		
22	Quotation		ERP		
23	Radiology		Hospital	Radiology Rooms	Daniel
24	Sales Order Document		ERP		
25	Sales Report		ERP		
26	Training Course		ERP		
27	Vendor Contract		ERP		

Figure B.4: Business Objects' File Data

	A	F	G
1	Name	Ca	Case Study
2	Admission		Hospital
3	Billing		ERP
4	Campaigns		ERP
5	Credit Check		ERP
6	Deliver		ERP
7	Discharge		Hospital
8	Examination & Diagnosis		Hospital
9	Inventory Management		ERP
10	Maintenance		ERP
11	Order		ERP
12	Patient Treatment		Hospital
13	Payment Processing		ERP
14	Payroll		ERP
15	Planning		ERP
16	Procurement		ERP
17	Production		ERP
18	Quote		ERP
19	Recovery		Hospital
20	Recruitment		ERP
21	Training		ERP

Figure B.5: Business Processes' File Data

	A	E
1	Name	Case Study
2	Administrative Staff	Hospital
3	Business Developer	ERP
4	CEO	ERP
5	CFO	ERP
6	Doctor	Hospital
7	HR Director	ERP
8	Laboratory Staff	Hospital
9	Maintenance Directo	ERP
10	Maintenance Engine	ERP
11	Maintenance Operat	ERP
12	Marketing Manager	ERP
13	Nurse	Hospital
14	Patient	Hospital
15	Payroll Specialist	ERP
16	Pharmacist	Hospital
17	Planning Staff	ERP
18	Procurement Manage	ERP
19	Production Director	ERP
20	Production Operator	ERP
21	Production Supervisc	ERP
22	Purchasing Staff	ERP
23	Radiologist	Hospital
24	Reception Staff	Hospital
25	Recruiter	ERP
26	Sales Assistant	ERP
27	Sales Director	ERP
28	Senior Controller	ERP
29	Talent Manager	ERP
30	Treasury Staff	ERP
31	Visitor	Hospital
32	Warehouse Manager	ERP
33	Warehouse Operator	ERP

Figure B.6: Business Roles' File Data

	A	D	G	V
1	Name	Application	Case Study	Realizes
2	Admission Form		Hospital	
3	Advertising		ERP	Advertising
4	Billing Document		ERP	Billing Document
5	CV		ERP	CV
6	Consent to treatment Form		Hospital	
7	Credit Document		Hospital	
8	Delivery Document		ERP	Delivery Document
9	Discharg Document		Hospital	
10	Electronical Medical Record		Hospital	
11	Fiscal Reports		ERP	Fiscal Reports
12	Inventory Report		ERP	Inventory Report
13	Laboratory Results		Hospital	
14	Maintenance Order		ERP	Maintenance Order
15	Medication & Dosage Form		Hospital	
16	Offer Proposal		ERP	Offer Proposal
17	Payment Document		ERP	Payment Document
18	Payslip		ERP	Payslip
19	Planning Order		ERP	Planning Order
20	Prescription	Pharmacy IS	Hospital	
21	Production Order		ERP	Production Order
22	Purchasing Order Document		ERP	Purchasing Order Documen
23	Quotation		ERP	Quotation
24	Radiology		Hospital	
25	Sales Order Document		ERP	Sales Order Document
26	Sales Report		ERP	Sales Report
27	Training Course		ERP	Training Course
28	Vendor Contract		ERP	Vendor Contract

Figure B.7: Data Objects' File Data