Enterprise Information Architecture Patterns for Government

Rute Sofia Caronho Lemos

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. André Ferreira Ferrão Couto e Vasconcelos

Examination Committee
Chairperson: Prof. José Carlos Martins Delgado
Supervisor: Prof. André Ferreira Ferrão Couto e Vasconcelos
Member of the Committee: Prof. José Luís Brinquete Borbinha

October 2016
I owe my deepest gratitude to Professor André Vasconcelos. This dissertation would not have been possible without his valuable guidance and expertise and it was an honor to have him as my advisor.

I am also thankful to Lisdália Sanches and Marco Pedro from AMA for the support given in my thesis and for their participation in reunions and revision of my results, thus adding value to this dissertation.

I am indebted to my family for supporting me throughout all my studies at university. I would like to deeply thank and dedicate this dissertation to my grandmother Maria de Jesus Caronho, my mother Ilda Caronho, my brother Hugo and my father António Lemos for all their love, support and dedication.

A special dedication to my grandfather Vitor Manuel da Fonseca Caronho, who passed away last year. He has always supported and guided me with much love and to who forever I will always be indebted.

I would also like to thank the scholarship given by INESC-ID.

Finally, I would like to show my gratitude to my colleagues and friends that directly or indirectly helped me during this dissertation.
In memory of my dear grandfather, Vitor Caronho. I owe everything to you and grandma. Thank you for always being there for me.
Abstract

Currently, there isn’t a specific method to find patterns for Enterprise Information Architecture in the area of Government. The goal of this work is to investigate the use of patterns and anti-patterns and propose a method to generate Enterprise Information Architecture patterns catalog for the Portuguese Government. Using patterns the organizations are expected to document proven practice solutions for recurring problems in a given context, in order to be used when creating new Enterprise Information Architecture in future projects.

To accomplish that, we propose a five-phased method to generate patterns which involves the selection of Enterprise Information Architecture documents with similar context, the comparison between them using an iterative process with weighted similarity rules, the generation of the pattern, classifying it as a pattern or anti-pattern, and its documentation.

For demonstration purposes, an example based on a practical case is shown. It is also demonstrated the use of the method in the specific case of a Portuguese Government Agency, AMA.

Evaluation was made using interviews and also by comparison with other Enterprise Information Architecture approaches.

**Keywords:** Enterprise Information Architecture, Patterns, Informational Entities, Pattern Documentation.
Atualmente, não existe um método específico de descoberta de padrões de Arquitetura de Informação Empresarial na área da Administração Pública. O objectivo deste trabalho é investigar o uso de padrões e anti-padrões e propor um método para gerar um catálogo de padrões de Arquitetura de Informações Empresariais para a Administração Pública Portuguesa. Usando padrões, é expectável que a organização documeente práticas de soluções comprovadas num dado contexto, de forma a serem usados em futuros projectos que envolvam a criação de uma nova Arquitetura de Informação Empresarial.

Para se atingir o objectivo mencionado, propomos um método de cinco fases que envolve a seleção de documentos de Arquitetura de Informações Empresariais com contexto semelhante entre si, a comparação entre essas arquiteturas usando um processo iterativo com pesos nas regras de similaridade, a geração dos padrões, a classificação como padrão ou anti-padrão e a documentação dos mesmos.

Para demonstrar o método, um exemplo baseado num caso prático é mostrado. Também é demonstrado o método numa agência da Administração Portuguesa, a AMA.

A avaliação foi feita recorrendo à comparação com outras frameworks de Arquiteturas de Informação Empresariais.

**Palavras-chave:** Arquitectura Informacional, Padrões, Entidades Informacionais, Documentação de Padrões.
## Contents

### List of Tables

vii

### List of Figures

viii

### Acronyms

ix

### 1 Introduction

1.1 Motivation ......................................................... 1
1.2 Problem Statement ............................................... 2
1.3 Contributions ..................................................... 3
1.4 Research Method .................................................. 3
1.5 Document Structure ............................................... 5

### 2 Related Work

2.1 Enterprise Architecture .......................................... 6
2.2 Information Architecture ......................................... 7
2.3 Enterprise Information Architecture ................................ 9
2.4 Data Modeling ..................................................... 11
   2.4.1 UML .......................................................... 11
   2.4.2 Entity-Relationship Modeling ................................. 12
   2.4.3 Comparison between ER Model and UML ...................... 12
2.5 Pattern ............................................................ 13
   2.5.1 Documentation of Patterns ................................. 13
2.6 Work developed by AMA ........................................... 14
2.7 Reverse Engineering ............................................... 15
   2.7.1 Removal of tables exclusive to DBs .......................... 15
   2.7.2 Removal of attributes exclusive to DBs .................... 15
   2.7.3 Removal of Foreign Keys .................................... 16
   2.7.4 Removal of Tables which contain only exclusive Foreign Keys 16
2.8 Other related work ............................................... 17
2.9 Discussion ......................................................... 20

### 3 Solution Proposal

3.1 1st Phase - Gathering of EIA documents .......................... 23
3.2 2nd Phase - Comparison of the EIA ............................... 23
   3.2.1 Representation of EIA in a Relational Databases Model .... 27
   3.2.2 Representation of EIA in an UML Model .................... 27
   3.2.3 Representation of EIA both in Relational Databases and UML Models 28
List of Tables

2.1 Data features by Inmon. ................................................. 9
2.2 Different types of diagrams in UML language. .................. 11
2.3 ER and UML comparative analysis. .................................. 12

3.1 Conditions when selecting EIA's to be compared. ............... 23
3.2 Process Iteration of the Similarity Rules. ......................... 26
3.3 Similarity Rules for Relational Databases. ......................... 27
3.4 Similarity Rules for EIA modeled in UML. ......................... 28
3.5 Classification of Patterns and Anti-patterns ....................... 34

5.1 Analysis of the number of people needed to create each catalog. .......................... 50
5.2 Comparison of the catalogs according to quality factors of Moody & Shanks. .............. 51
5.3 Differences between our work method and Brás' method. .............. 52
5.4 Comparison between our work method and Yesika's. .................... 53
List of Figures

1.1 The DSRM process. ........................................... 4

2.1 The Enterprise Architecture layers. .............................. 7
2.2 A conceptual approach to EIA Reference Architecture. .............. 10
2.3 Removal of tables exclusive do DBs. ............................. 15
2.4 Removal of attributes exclusive to BDs. ......................... 16
2.5 Removal of Foreign Keys. ....................................... 16
2.6 Removal of Tables which contain only exclusive Foreign Keys. ........ 17

3.1 Process model of the solution proposal. .......................... 21
3.2 Solution Proposal steps. ......................................... 22
3.3 Flowchart of the process to calculate the name similarity between IEs. ......................... 24
3.4 Application of the Similarity Rules according to meta-model. ........... 26
3.5 Reverse Engineering transformations. ............................. 29
3.6 Rule 1 of the Pattern Generation. .................................. 30
3.7 Rule 2 of the Pattern Generation. .................................. 31
3.8 Rule 3 of the Pattern Generation. .................................. 31
3.9 Rule 4 of the Pattern Generation. .................................. 32
3.10 Rule 4 of the Pattern Generation. .................................. 33
3.11 Structure of the documented patterns. .............................. 35

4.1 Database tables of Model A from the academic example. ................. 38
4.2 Model B from academic example. .................................. 39
4.3 Model A after reverse engineering techniques, from academic example. .......... 40
4.4 The resulting pattern from academic example. ...................... 41
4.5 Relational Database of AMA with tables that only have meaning in a DB model. .......... 43
4.6 Relational Database of AMA with tables containing exclusively and only foreign keys. .... 44
4.7 Relational Database of AMA with table with foreign keys. ............ 44
4.8 Relational Database of AMA with an example table named Group_. .................. 45
4.9 IE Group_ from Portal do Cidadão modeled to UML. .................... 45
4.10 Balcão do Empreendedor translated to English, due to phase 1 conditions. ............ 46
4.11 Example of an AMA’s generated pattern. .......................... 47
4.12 AMA example of a generated pattern documented in AMA’s EIA Pattern Catalog. .......... 48

5.1 Part of AMA’s Service Catalog. ................................... 49
5.2 The data model quality factors by Mood & Shanks. .................... 50
Acronyms

AMA Agência para a Modernização Administrativa. 2, 3, 5, 36, 42

DB Database. 7, 15, 16, 28

DSRM Design Science Research Methodology. 3, 26

EA Enterprise Architecture. 6, 17

EIA Enterprise Information Architecture. 1-3, 5, 21, 25, 27, 35, 42

ER Entity-Relationship. 11

FK Foreign Key. 37

IA Information Architecture. 7

IDEF1X Integration DEFinition for information modeling. 12

IE Informational Entity. 8, 18, 19, 21, 23, 25, 31, 32

IS Information Systems. 2, 3

IT Information Technology. 9

PA Public Administration. 1-3, 21

PK Primary Key. 37

UML Unified Modeling Language. 11, 27

URI Uniform Resource Identifier. 24
Chapter 1

Introduction

Enterprise Information Architecture (EIA) has become very important for business sustainability and competitive advantage today. According to Godinez et al. [1], firms of all dimensions search for practical ways to create business value by using information and correlating insights to be able to predict outcomes. This information era poses unique challenges, such as how to make information quick and easily accessible to people and processes that need it, and at the same time, how to protect and secure that information and mitigate risks inherent in business decision making [1].

As enterprises become increasingly information based, making improvements in their information activities is a top priority to assure their continuing competitiveness. A key to achieve these improvements is developing an Enterprise Information Architecture. An EIA can be viewed as a structured set of multidimensional interrelated elements that support all information processes [2].

1.1 Motivation

Information and systems for its management are critical elements for the efficient and effective operation of today’s knowledge dependent organizations [3]. From an organization’s perspective, information management ensures that valuable information is acquired and exploited to its fullest extent. Because of the critical dependency of organizations on information, improving its overall management can yield significant operational benefits to all areas of an organization and importantly its overall efficiency, competitiveness and responsiveness [3].

Therefore, improving the management of information has been a real concern for organizations, and for that reason, EIA in organizations is extremely important. Public Administration (PA) is no different, and, as a result, the development and improvement of an EIA as a reference for the PA has become a fundamental goal.

The discovery and use of patterns for EIA can help enterprises meet their future needs, reduce incompatibilities, missing elements and unnecessary duplication in information. A large number of papers and books published about patterns supports the view that the concept of the design pattern is valued by many experienced developers as a categorization of recurring issues (and solutions), as well as providing a widely used vocabulary for discussing design [4].

The experience of experts enables them to find solutions for many recurring problems, creating design patterns. Juniors can resolve problems as if they have many years of experience by reusing solutions for similar problems [5]. Despite the importance of patterns, there is not a process that allows its discovery, since the specification of patterns is realized by experts.

Defining patterns for any type of architecture in order to make use of the knowledge and experience
acquired by architects through their projects is necessary, since it can help reducing effort and time in the development of architectures and can be a guide for future works with similar contexts [6].

There are many patterns and anti-patterns yet to be discovered, because standard existing patterns, such as client-server [7], don’t cover all existing architectures in EIA [8]. Even though each system is unique, there are many systems with similar areas with similar architectures, and as such, there’s a probability that patterns can be discovered.

Also in the Portuguese Administration context we have faced the lack of specific methodologies for improving and updating the Enterprise Information Architecture, regarding future projects with similar context. As so, this thesis proposes a pattern-based approach for improving currently existing approaches for EIA with special attention to PA organizations, where we will make an EIA patterns catalog for Government, to be used as the basis in future projects with akin context.

This work will be focused on the Portuguese Government organization, more specifically, Agência para a Modernização Administrativa (AMA), the agency responsible for the execution of the project of the information architecture for the Portuguese public administration.

1.2 Problem Statement

This section describes the “Identify Problem & Motivate” step of the DSRM Process Model and has the objective to describe the research problem and to justify the value of a solution.

Information management systems that are not well aligned to the organization or the existing Information Systems (IS) infrastructure can have a significant detrimental effect on the organization and its performance [9].

Because of the critical dependency of organizations on information, improving its overall management can yield significant operational benefits to all areas of an organization as well as its overall efficiency, competitiveness and responsiveness [10].

Saiz et al. [11] confirm that in the literature, there are some frameworks that deal with performance management/measurement for inter-organizational contexts, and that each framework deals in detail certain aspects of performance measurement and present a specific structure. However, the authors say that practically all of these frameworks present a common gap: the low degree of consideration of information treatment (acquisition, analysis, storage). As such, they recognize the importance of developing information architectures, methodologies and systems to facilitate performance management’s success in the long term [11].

Ernst, in [12], states that methods for EA management (which includes EIA) are documented on a too generic level to solve problems occurring in the EA management context. He also states that existing methods are often too company-specific and can therefore not be used as a general solution to existing problems. He affirms that a few engineering oriented approaches to EA management are currently in development, but none of them can be considered to be widely accepted in the EA community. Typical frameworks for managing the EA also only partially contribute to the solution of the aforementioned problems as they may be helpful for an architect in his daily work, but knowing them does not make one an architect, because experience and knowledge is required to apply a framework.

In the Portuguese Public Administration context, as declared by EORG@AP Group and by Gomes [13], one of the main concerns that need to be regulated at the Public Administration’s Enterprise Architecture is the accessibility, reliability, ontology and security of the information entities. Furthermore, information entities are part of the independent components in the Enterprise Architecture and they represent all the human resources, material and immaterial involved in the activities performed.

Fowler [14] stated that “Frequently I find that many aspects of a project revisit problems I have faced
before”. Good architecture is a critical factor in the success of the system development. As mentioned in the previous chapters, a prominent approach to document knowledge in software engineering are patterns, which originally were introduced in architecture where patterns are used to document recurring solutions to common problems in a given context [12].

Regarding the development of the IS in some sectors of the PA, the non-conformity with a certain set of norms and requirements can lead to the lack of interoperability between those systems [15], and the increase in the difficulties in its maintenance [16].

Therefore, the importance of information in the Portuguese PA, the lack of specific methodologies for improving and developing the Enterprise Information Architectures and the nonexistence of a method to create an EIA patterns catalog in the Portuguese Public Administration regarding future projects with similar context, lead us to need for the Portuguese Public Administration to have a catalog of EIA patterns to be used as the basis for future projects that have akin context with former PA projects.

The questions that this research will address are:

• Is it possible to create a catalog of Information Architectures patterns to be used by the Portuguese PA specifically AMA ?

• How do we compare different Enterprise Information Architectures?

• How can we extract patterns and anti-patterns?

• How do we document Enterprise Information Architecture patterns and anti-patterns?

1.3 Contributions

Based on the problem questions described above, this thesis contributes to the development of both Enterprise Information Architecture and pattern themes, especially among the Portuguese Public Administration. As such, we aim such contributions by:

• Stressing the importance of using patterns in Enterprise Information Architecture;

• Defining the conditions for generating Enterprise Information Architecture patterns;

• Proposing a method to compare different Enterprise Information Architectures, based on different methodologies;

• Propose a method to extract Enterprise Information Architectures patterns;

• Define how to document those patterns;

• Creation of an EIA patterns catalog to be used by AMA

In addition, to disseminate this work, the paper [17] was accepted and presented at 8º INForum[1]

1.4 Research Method

The methodology applied is Design Science Research Methodology (DSRM) where a proposal is developed and validated to solve our problem [18]. It is a system of principles, practices and procedures needed to execute a study. Its objective is to overcome research paradigms, for instance the traditional

descriptive and interpretative research, in which the outputs are mostly explanatory and, one could dispute, are often not applicable in practice [19].

It consists in an iterative process composed by 6 phases. In Figure 1.1 the DSRM steps are visually presented, which are the following:

- **Problem identification and Motivation**: Definition of problem’s importance and the necessity of a solution. This phase is in the chapter 1, namely section 1.1 and 1.2;

- **Definition of the objectives for a solution**: Presentation of requirements that should be fulfilled by the solution to implement. This is carried out in chapter 2, “Related Work”, and in chapter 3, “Solution’s Objective”, which covers aims and objectives as the awareness and recognition of a problem from a state of the art review giving us the issues that must be addressed;

- **Design and Development**: Key element of the DSR methodology where artifacts will be implemented to address requirements. In the context of this work, the designed artifact was the catalog of enterprise information architecture patterns, and chapter 3 gives the details on how it was accomplished;

- **Demonstration**: Confirmation of application of artifact to the problem’s requirements. This phase corresponds to the chapter 4 of this work;

- **Evaluation**: Measurement of the level in which the artifacts produced fulfill the initial problem. It is shown in chapter 5;

- **Communication**: Documentation and spreading of the artifacts as the problem’s solution. This will be archived by using this thesis and publish articles to communicate the artifact, its value and utility. This work was accepted and presented in the Inforum 2016 Conference, which is addressed in section 6.2.

As mentioned before, the design science research method follows an iterative approach, which means that the phases solution goals, design and development, demonstration, evaluation and communication, are revisited throughout the execution of different iterations for the identified problem and motivation.
It finishes when the answer to the stated problem is achieved, having as evidence the outcome of the results of applying the proposed solution.

If there is no positive answer to the stated problem, goals of the solution should be revised and a different approach should be addressed.

The starting point was the evaluation of a previous work by Brás [20]. For the problem stated in section 1.2, various iterative actions where overseen. On the first iteration, a evaluation of a previous work by Brás [20] was carried out, since part of his work was going to be used for this thesis. During the next iterations, some improvements were being made, such as the need to add a new similarity rule for comparing different EIAs after initially established in the design and development phase of the DSRM. There were also some modifications on the pattern generation rules. A basic condition was also added in our solution proposal after an evaluation phase of an DSRM iteration. Due to the several iterations and experiments, we were also able to choose the best value for the cut-off in the matching process of our solution proposal.

1.5 Document Structure

This dissertation is composed by six chapters. To be coherent with our research work, this dissertation will follow the same structure as DSRM which phases are easily mapped to the structure of this document.

First, in chapter 1, the motivation, problem statement, contributions and research method are presented.

In chapter 2, the related work is detailed, namely Information architecture, Enterprise Information Architecture, the definition of patterns and its documentation, among other related works.

The Problem Statement section and the Related Work chapter identify the problem and the motivation behind the research work.

In chapter 3, the solution hypothesis and proposal are presented, where the objectives of the solution are detailed as well as the proposed solution.

In chapter 4, we demonstrate the application of our artifact with an academic example and also with a sample of [AMAs EIA] namely the comparison between Portal do Cidadão with Balcão do Empreendedor.

In chapter 5, we show the evaluation and analysis results.

Finally in chapter 6, some conclusions are presented, some lessons learned are detailed and suggestions for future work are given.

The communication step of the DSRM is addressed in section 6.2.
Chapter 2

Related Work

In this chapter, it is presented a literature review of the topics related and most relevant to this work. This chapter starts by introducing the concepts of the different types of architectures, followed by the definition of a graphical notation for representing software systems - and the information entities by Inmon. Later on, it’ll be introduced the concept of patterns and methods to document them. The work developed by AMA will be described as well as other related works concerning this thesis’ theme.

2.1 Enterprise Architecture

Aier [21] states that the basic idea about Enterprise Architecture (EA) is to map the most important artifacts of an enterprise together with their relationships to a model. The modeling seeks to describe the reciprocal dependencies of the subjects of organization of an enterprise on an aggregated level. This is done for documentation and for analysis reasons, but also to plan the future development [12].

Ersnt [12] noted that there are various definitions of EA which lead to the problem that there is no general accepted definition of the term, ending up in confusion in discussions about this topic. He pursued a similar approach like TOGAF and resorts to ISO/IEC 42010 standard as a basis for the definition of EA which is the following: “the enterprise architecture is the fundamental conception of the enterprise in its environment embodied in its elements, their relationships to each other and to its environment, and the principles guiding its design and evolution” [12].

The Zachman Framework says that an organization doesn’t have just one architecture, but a set of them, arranged as layers. Each of these layers produce artifacts that answer six organizational questions (what, where, when, why, who and how) [22]. The alignment between business and IT and the coordinated steering based on information about the EA are the most important aspects of EA management.

The complexity of the Enterprise Architecture can be represented based on a collection of architectural layers: each layer supports the needs of the one above it, with the top one precisely supporting the capabilities required by the business strategy [1]. These layers are shown in Figure 2.1.

The layers mentioned above are the described in more detail as it follows [1]:

- **Enterprise Strategy Layer**: this layer mainly describes an enterprise strategy regarding product portfolio and appropriate customer segments, appropriate delivery and distribution channels in the given market environment, competitors and core competencies, and capabilities of the company;

- **Business Layer**: in order to develop an Enterprise Architecture, it is necessary to provide the process and integrated tools to identify the as-is state of the organization (the business and IT
ecosystem) and the desired, to-be state. Enterprise Architecture promotes the formation of enterprise blueprints that show how business processes are in the present state and how they can be implemented, exploiting the full range of capability of underlying IT architectural building blocks;

- **Application Layer**: this layer supports the business and describes the required business functions in the underlying IT application systems, which is very important, since the rapid increase of applications, systems and platforms and their inter-dependencies turns the procedure of adding and improving IT capabilities a possible risk for the business without a EA strategy;

- **Information Layer**: information is the fuel that drives business artifacts; its flow achieves value to the user. One key objective of the EA is to convert raw data into relevant information that gives additional insight and value to the business;

- **Infrastructure Layer**: the infrastructure layer consists of the server, network and storage infrastructure which supports higher-level functions, for example, a Database (DB) applications or e-mail servers. There is a high demand for more flexibility and agility also from this layer, due to cost pressure, leading to higher degrees of virtualization and systems consuming less energy, reducing electricity costs.

![Figure 2.1: The Enterprise Architecture layers.](image)

Regarding the enterprise architecture’s layers explained before, the scope of this thesis is the information layer.

### 2.2 Information Architecture

An Information Architecture (IA) is a high-level map of the information requirements of an organization [23]. IA, or information systems architecture, is becoming one of the most important issues in IS development [24].

IA gives information-relevant concepts and frameworks, in order to deal in a steady and integrated way with the technology to assure the responsiveness and trusted information insight that a business needs from its information layer.

The IA delineates the principles and guidelines that allow consistent implementation of information technology solutions, the way data and information are managed and shared across the enterprise, and how to gain business-relevant trusted information insight[1]. Following are some examples of the core principles that guide an Information Architecture, as defined by Godinez et al.[1]:

...
- **Access and exchange of information**: information services should give unconstrained access to the appropriate users at the right time;

- **Service re-use**: promote the discovery, selection and re-usability of services and reassure the use of uniform interfaces when possible;

- **Information Governance**: proper information technology should support the execution of an Information Governance strategy in an efficient way;

- **Standards**: a group of coherent standards for data and technology should be defined to facilitate simplification across the Information Infrastructure.

When an Information Architecture is well designed and implemented, the consistent use of information by all significant services and business applications is facilitated, as well as the access and transaction of information with services. The discovery and reuse of services is also promoted. For all the reasons mentioned above, an IA can deliver a stable, responsive, and consistent information-centric system behavior [24].

Regarding the composition of an IA, it’s possible to distinguish three fundamental concepts[25]:

- **Informational Entity (IE)**: any concept that has a meaning in the business context and in which is possible and relevant to store information.
  This concept can be, for example, a person, a place or anything physical and is usually characterized by having a name, an unique identifier, a description and its structural relationship with other informational entities (and the derived relations with processes and applications);

- **Attribute**: any characteristic that defines an informational entity;

- **Relationship**: any pair of attributes related between themselves that add detail in the business context.

To conclude, we can say that an IA is composed by informational entities, characterized by its attributes and by the relationships that the different informational entities establish between themselves.

Another concept of Information Architecture is the Inmon taxonomy, introduced by W. Inmon, which is a methodology to classify data. According to [26], data can be classified according to three dimensions: primitive vs. derived; historic vs. projected; publics vs. private. As so, there’s the following data types:

- **Primitive Data**: are specified with a single fact;

- **Derived Data**: calculated data aggregated or summarized; based on one or more primitive data or other data derived from;

- **Historical Data**: record unambiguous and irrefutable facts; the values recorded are accurate and there is agreement about the means to obtain or calculate them;

- **Projected Data**: are estimates, predictions or inferences from facts that might happen and there may not be unanimity about the means to obtain or calculate;

- **Public Data**: data that have multiple stakeholders and may be visible outside the organization. Its integrity is maintained by the organization;

- **Private Data**: are owned by a single individual or group of individuals. They reflect specific needs and may not be relevant or not to make sense out of a limited context. Its integrity is managed locally.
Considering only primitive and derived data, they might be analyzed according to some aspects of their features and use which are: detailing level, performance, use patterns, availability, users, updates, definition, access level, and modification which is expressed in Table 2.1.

<table>
<thead>
<tr>
<th>Name</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail Level</td>
<td>High</td>
</tr>
<tr>
<td>Performance</td>
<td>High</td>
</tr>
<tr>
<td>Use Patterns</td>
<td>High</td>
</tr>
<tr>
<td>Availability</td>
<td>High</td>
</tr>
<tr>
<td>Users</td>
<td>Several</td>
</tr>
<tr>
<td>Updates</td>
<td>Immediate</td>
</tr>
<tr>
<td>Definition</td>
<td>Static</td>
</tr>
<tr>
<td>Amount of Access</td>
<td>Many and Short</td>
</tr>
<tr>
<td>Modification</td>
<td>High</td>
</tr>
</tbody>
</table>

Table 2.1: Data features by Inmon.

The concepts defended by Inmon, at the level of information architecture, are not supported specifically in a modeling language. Thus, Inmon’s work is essentially a good starting point regarding the identification of concepts, attributes and relations for the information architecture [27].

2.3 Enterprise Information Architecture

The Enterprise Information Architecture is basically the framework that defines the information principles, architecture models, standards, and processes which facilitates information technology decisions across the enterprise.

The EIA converts the business requirements into informational strategies. In the information supply chain, it also defines which data components are needed by whom and when. Furthermore, every business has the need to generate and maintain trusted information that is derived by significant data parts [1], and the EIA addresses these concerns. The “enterprise” in the definition of EIA adds the enterprise-wide business context to the definition of IA described in the previous section [1], as shown in Figure 2.2.

A Reference Architecture is a template that encapsulates at an abstract level the results and best practices derived from multiple deployments of solutions to a given business problem. Thus, they are a valuable tool for Information Technology (IT) Architects to help identify and assess gaps and reduce risks in the solution development cycle.
Making consistent decisions across multiple business units, departments and individual projects is a challenge that most organizations have to deal with. In this context, the EIA is a key component of the needed framework for making effective decisions, since it defines the guiding principles for the strategy on how to deal with the needs of the organization. It also defines the information-centric technology infrastructure that supports them. The EIA defines what is necessary for the organization to manage data and information over its lifetime in terms of technological skills and processes. It also helps to optimize content-based operational processes and business performance, including managing and delivering trusted information.

By aligning the needs of business with the technology and the flow of information in the supply chain, EIA delivers flexibility, agility and responsiveness to the business process and the organization. The main objective of the EIA is to reduce complexity, so it helps to eliminate the factors that act as inhibitors to change and, at the same time, it contributes to new business paradigms [1].

Primary characteristics that can be used to distinguish a well-defined EIA implementation include the following [1]:

- **Gaining transparency**: the information remains independent from applications and user interfaces. It provides a transparency layer between the information and application domains;

- **Considering enterprise business requirements**: the architecture takes into account the overall information needs of the entire enterprise;

- **Avoiding inconsistencies**: it should be useful to identify gaps, inconsistencies, conflicts, and overlaps in the data and information, and should be helpful when selecting proper solutions by offering a concept, framework, and methods to resolve this;

- **Managing Service Level Agreements (SLA)**: it should provide mechanisms for the definition and administration of information-centric SLAs which can be monitored and enforced;

- **Enabling decision making**: if the architecture is flexible and extensible, it can enable more consistent and efficient IT decision making that is linked to business needs;

- **Addressing reusability aspects**: Enforcing an EIA means that information assets are shared and reused, avoiding data duplication and thus reducing development, service, and support costs;
• **Addressing data scope**: The Information Reference Model (see Chapter 3) used by the enterprise describes the scope of the used data and information supported by the EIA;

• **Defining a technology strategy**: It establishes the framework upon which the technology strategies adopted by the enterprise depend. In addition, it defines the set of principles that guide how an organization’s information systems and technology infrastructure are engineered.

### 2.4 Data Modeling

We address two different approaches for data modeling: Entity-Relationship (ER) modeling and UML modeling. Those are the ones that are mostly used among industry, and we found them as the ones that should be studied in order to help us developing our work. Data modeling using UML is addressed first, followed by a resume of Entity-relationship modeling.

#### 2.4.1 UML

The [Unified Modeling Language (UML)](https://www.uml.org) is an open standard controlled by the Object Management Group (OMG) which is a family of graphical notations, defined over a single meta-model. It is useful in the design and description of software systems, especially those which use an object-oriented approach [28].

To date, UML is mainly used in industry for designing object-oriented program code. Although it can be used for designing databases, it had little success in displacing other approaches such as ER for this purpose [29]. However, Halpin [29] states that UML is a very important language that could well become popular for database design in the future. As such, we address UML notation in the data-modeling context, especially because of the background information regarding our problem and solution.

UML 2.0 defines thirteen types of diagrams, divided into three categories: six diagram types that represent static application structure; three represent general types of behavior; and four represent different aspects of interactions[30].

**Table 2.2** depicts the different types of diagrams in the UML language.

<table>
<thead>
<tr>
<th>Diagrams included</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Structure Diagrams</strong></td>
<td>Class Diagram</td>
</tr>
<tr>
<td></td>
<td>Object Diagram</td>
</tr>
<tr>
<td></td>
<td>Component Diagram</td>
</tr>
<tr>
<td></td>
<td>Composite Structure Diagram</td>
</tr>
<tr>
<td></td>
<td>Package Diagram</td>
</tr>
<tr>
<td><strong>Behavior Diagrams</strong></td>
<td>Use Case Diagram</td>
</tr>
<tr>
<td></td>
<td>State Chart</td>
</tr>
<tr>
<td></td>
<td>State Machine Diagram</td>
</tr>
<tr>
<td></td>
<td>Activity Diagram</td>
</tr>
<tr>
<td><strong>Interaction Diagrams</strong></td>
<td>Sequence Diagram</td>
</tr>
<tr>
<td></td>
<td>Communication Diagram</td>
</tr>
<tr>
<td></td>
<td>Timing Diagram</td>
</tr>
<tr>
<td></td>
<td>Interaction Overview Diagram</td>
</tr>
</tbody>
</table>

Table 2.2: Different types of diagrams in UML language.
The one which is used in this thesis is the class diagram represents the static view of an application and describes the attributes and operations of a class and also the constraints imposed on the system. The class diagram is the only UML diagram that can be mapped directly with object oriented languages and therefore, it is widely used in the modeling of object oriented systems [28].

Furthermore, the class diagram shows a collection of classes, interfaces, associations, collaborations among the elements of the static view and constraints, while describing the functions performed by the system.

2.4.2 Entity-Relationship Modeling

The ER approach was originally introduced by Chen, in 1976 [31]. Its original notation uses rectangles for entity types and diamonds for relationships. Attributes may be defined, but are excluded from the ER diagram.

Although the ER diagrams do not display attributes, relationships may have attributes, but they can’t play roles in other relationships, according to [29]. Relationships are formalized in terms of ordered tuples of entities, allowing the order to be dropped if role names are used [31].

Citing [29], one problem with the ER approach is that there is an abounding number of versions of the ER modeling, with no single standard. Nonetheless, in industry, the most popular versions of pure ER are the Barker and Information Engineering notations [32]. Another popular notation is Integration DEFinition for information modeling (IDEF1X) [33], which is a hybrid of ER and relational notation.

2.4.3 Comparison between ER Model and UML

With the two data modeling approaches described, we are going to analyze them taking in consideration both their key features and this work’s context. As such, we focused this comparative analysis in the following vectors, based on [13]:

- **Entities representation**: defines how well an entity is represented in the modeling language;
- **Attributes representation**: how well the relationships are depicted in the modeling language;
- **Relationships representation**: how well the relationships are depicted in the modeling language;
- **Alignment with enterprise context**: aims to frame the modeling language in this work’s context, especially the Enterprise Information Architecture context. This vector raises with the restrictions of the EIA patterns that we aim in our research.

<table>
<thead>
<tr>
<th></th>
<th>Entities</th>
<th>Attributes</th>
<th>Relationships</th>
<th>This work’s context</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UML Modeling</strong></td>
<td>++</td>
<td>++</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td><strong>ER Modeling</strong></td>
<td>++</td>
<td>+</td>
<td>++</td>
<td>+</td>
</tr>
</tbody>
</table>

Table 2.3: ER and UML comparative analysis.

This analysis allows us to understand why the UML modeling language was the one that we picked in the solution proposed. These languages are quite similar, but the UML modeling is actually stronger in depicting the attributes than the ER modeling. Even though the ER modeling is better at depicting
the relationships between entities, it lacks when considering our context, since the work at present
developed in the IA area in the Portuguese Public Administration already uses UML modeling as basis.

2.5 Pattern

Pattern is a concept well-known and used in various disciplines. One of these disciplines is architec-
ture, which also established one of the earliest definitions for patterns. Patterns evolved from several
initiatives. Kent Beck and Ward Cunningham, two of the pioneers of Smalltalk, came across the ideas
of Christopher Alexander, who had developed a theory and collection of patterns in architecture [14].
Alexander developed a range of theories about patterns in architecture and published these in a series
of books [34].

Alexander et al. [34] declared that “each pattern describes a problem which occurs over and over
again in our environment, and then describes the core of the solution to that problem, in such a way that
you can use this solution a million times over, without ever doing it the same way twice. Each pattern is
a three-part rule, which expresses a relation between a certain context, a problem and a solution”.

Fowler [14] stated that as a result of the diversity of patterns community, he had difficulty in defining
the term pattern. He further affirms that we all think we can recognize a pattern when we see it, we think
most of us would agree in most cases, but we cannot come up with a single definition. He concluded
that “a pattern is an idea that has been useful in one practical context and will probably be useful in
others”.

The fourth volume of the Pattern-Oriented Software Architecture series [35] defines a pattern as
follows: “a pattern describes a particular recurring design problem that arises in specific design contexts
and presents a well-proven solution for the problem. The solution is specified by describing the roles of its
constituent participants, their responsibilities and relationships, and the ways in which they collaborate”.

Douglass explained in his book [36] that over the years, experienced developers have encountered
the same problems over and over, and even if they are not exactly the same problems, they have very
things in common. He believes that the very best developers abstract these problems and their solutions
into generalized approaches (patterns) that have proved consistently effective.

Patterns have become accepted in many different areas and described as very useful because they
reuse knowledge already acquired from experienced users and they capture and document proven
practices [12].

Anti-pattern is also a concept worth mentioning. It describes a commonly occurring solution to a
problem that generates decidedly negative consequences [37]. The anti-pattern may be the result of
a manager or a developer not having sufficient knowledge or experience in solving a particular type of
problem, or having applied a perfectly good pattern in the wrong context.

2.5.1 Documentation of Patterns

One of the most popular pattern forms is the Alexandrian Form [34], which consists of:

- **Name**: Should be memorable, usable and distinct;
- **Prologue**: One sentence per pattern that can be expected to precede this pattern;
- **Epilogue**: One sentence per pattern that can be expected to follow this pattern;
- **Problem statement**: One or two sentences that summarize the problem solved by the pattern;
- **Solution**: One or two sentences that tell what to do to solve the problem;
• **Diagram:** A picture or two, hand sketched or photographed, that illustrate the pattern (and sometimes the lack of the pattern);

• **Discussion:** Anywhere from 4 to 40 paragraphs that illuminate the system of forces resolved by the pattern.

One of the criticisms for this method of documenting patterns is that it breaks the flow of the pattern and makes you worry too much about whether issues should be discussed in the problem half or the solution half.

Another pattern form mentioned in several books is the Portland form \[12\], which consists of the following parts:

• **Name**;

• **Context:** The context and a brief description of the problem;

• **Problem:** Cause of the problem followed by the forces that must be resolved in order, roughly, from strongest to weakest and with conflicts between the forces highlighted;

• **Solution:** Describe a solution that resolves the strongest forces in this context;

• **Resulting context:** What has been resolved and what needs to be addressed next? What new possibilities are available at this point and what new problems have arisen? What possibilities are no longer available, among others;

• **Summary:** Discussion about the greater context in which this pattern belongs, related patterns, and the specific relationships between those patterns and this one.

Buschmann et al. \[35\] concluded that there is no ideal pattern form and creating a new pattern or selecting an existing one should always have in consideration the experience of the author, the intent that the pattern author has in mind when documenting a pattern, and the target audience.

### 2.6 Work developed by AMA

This work is developed in collaboration with *Agência para a Modernização Administrativa* (AMA), the agency responsible for the execution of the project of the information architecture for the Portuguese public administration. In order to be able to develop the intended catalog with enterprise information architecture patterns, a initial phase of the solution’s architecture will consist on the analysis of two specific contexts of the Portuguese public administration modernization, and it's important to describe them as it constitutes the state of the art in the subject in Portugal and will provide us a initial basis to work on.

Being this work developed in collaboration with AMA, we will have access to the projects currently under its orientation, like the *Citizen Portal* and the *Entrepreneur Counter*, that already integrate a fair set of information regarding the citizen and the entrepreneur. We will also have access to AMA’s Information Architecture.

A brief description of two of the projects of AMA that we will have access to and which will be used to compare Enterprise Information Architectures is as it follows:

• **Citizen Portal** (*Portal do Cidadão*, in Portuguese): Its main goal is to simplify the relationship between the citizens and the public organisms, being a privileged channel to the services provided by the public administration, where the citizens can also consult information online. Its data is in a MySQL and SQL database;
• **Entrepreneur Counter** (*Balcão do Empreendedor*, in Portuguese): It's the point of access to the services related to business activity, where the entrepreneurs can have access to various types of contents, like how to create a company, register a trademark, get certificates or how to obtain activities licensing. Its data is viewed in documents who use the UML standard.

• **AMA’s Information Architecture**: the document containing the model of AMA’s entities, its relationships and specializations. It's in a high abstraction level of representation.

### 2.7 Reverse Engineering

An abstraction for a software artifact is a succinct description that suppresses the details that are unimportant to software developer and emphasizes the information that is important. According to [38], reverse engineering is the process of analyzing a subject system to identify the system’s components and their interrelationships and create representations of the system in another form or at a higher level of abstraction.

Reverse engineering in and of itself does not involve changing the subject system or creating a new system. It is a process of examination, not a process of change or replication. Reverse engineering is useful to abstract models of DBs in order to create representations of those models in a higher abstraction level [20].

In this thesis, the set of transformations in reverse engineering will be used in the solution proposal (section 3.1). Those transformations are the following, adapted from [39] and [20]:

#### 2.7.1 Removal of tables exclusive to DBs

Tables that only have a meaning and utility at the DB level must be removed in a higher abstraction level.

Figure 2.3 depicts that table *Act_HI_ACTINST* has initially three attributes, *proc_def_ID, proc_inst_ID* and *call_proc_inst_ID*, that only have utility at the DB representation level. Thus, in the final result, this table is removed.

![Figure 2.3: Removal of tables exclusive do DBs](image)

#### 2.7.2 Removal of attributes exclusive to DBs

Attributes that only have a meaning and utility at the DB level must be removed in a higher abstraction level. Figure 2.4 shows that table *T* has initially two attributes, *db_createdBy* and *db_lastUpdatedBy*, that only have utility at the DB level. As such, in the final result, these two attributes are removed.

2.7.3 Removal of Foreign Keys

Attributes that represent foreign keys must be removed from the table, and a relationship between that table and the table where the attribute is the primary key must be formed.

Figure 2.5 illustrates the removal of foreign keys. Table T has an attribute attrib_d that is a foreign key. In table Q that attribute is the primary key. The final result removes the attribute attrib_d from table T and creates a relationship between the two tables.

2.7.4 Removal of Tables which contain only exclusive Foreign Keys

Tables that contain exclusively attributes that represent foreign keys must be removed, as well as the relationships that are formed between those tables with other ones. That type of table can be useful in DB level, but it is not in a higher abstraction level.

Figure 2.6 illustrates that transformation. Table R initially contains only two attributes, attrib_a and attrib_d, which represent the foreign keys of table T and Q, respectively. The final result includes the
removal of table R and the relationship of table R with the tables T and Q.

![Diagram of tables](image)

**Figure 2.6:** Removal of Tables which contain only exclusive Foreign Keys.

### 2.8 Other related work

In this section, we present two works by different authors that are related to this thesis problem and solution.

The first author is Yesika Reinolds. Regarding the subject of patterns in Enterprise Architecture, Reinolds [8] developed a method called to discover patterns and anti-patterns in EA. Her method was created to be used on all the layers of Enterprise Architecture, including the enterprise strategy layer, business layer, application layer, information layer and infrastructure layer, and also considers the relationships between these layers. Her work is based on a holistic view of the enterprise architecture, while this thesis focus on the information layer of an EA.

The second author is Tiago Brás, in his dissertation to obtain the Master in Information Systems and Computer Engineering, named *Arquitetura Informacional de Referência para o Setor da Saúde Portuguesa* [20], used the interoperability concept and refined the integration of schemas/models to suggest a method that, applying a bottom-up approach and starting with a group of information systems, allows us to get to an Information Reference Architecture (IRA) that can assure the maintenance efficiency and the semantic interoperability. It is very important to refer this work because this project will extend it and use it to compare different enterprise information architectures, not to get an IRA like his work’s main objective was, but to extract patterns, which is the goal of this project.

According to [40], schema integration is about constructing a global view, having by starting point a set of schema developed independently. It’s common for organizations to have many information systems, which are developed independently and which causes interoperability problems between them.

According to [20], the schema/model integration has the following phases:

1. **Preintegration:** An analysis of models is carried out before integration to decide upon some integration policy. This governs the choice of models to be integrated, the order of integration, and a possible assignment of preferences to entire models or portions of schemas. Note that it will only be used binary comparisons, i.e., only two different architectures at a time will be compared.
Global strategies for integration, namely, the amount of designer interaction and the number of schemas to be integrated at one time, are also decided in this phase. Collection of additional information relevant to integration, such as assertions or constraints among views, is also considered to be a part of this phase.

2. **Comparison of the Models**: Models are analyzed and compared to determine the mapping among concepts and detect possible conflicts. Inter-schema properties may be discovered while comparing models.

Given two models S1 and S2, having a match means that for each concept in S1, we try to find a concept in S2 that will be semantically similar. In this model matching phase, we may find the following heterogeneity:

- **Syntactic Heterogeneity**: differences at the level of the formats used in the representation of the elements;
- **Structural Heterogeneity**: differences at the level of the types and structures used to represent elements;
- **Semantic Heterogeneity**: differences in the interpretations that various people make of one element;
- **System Heterogeneity**: differences at the level of the system architectures, operation systems, hardware types, among others.

2.1 **Similarity between IE**

According to [20] and [41], two IE are similar if they share a set of characteristics between them. Formally, we can quantify this similarity in the following way:

- \( \text{Similarity}(A, B) \in [0...1] \);
- \( \text{Similarity}(A, B) = 1 \): the two IE are equal;
- \( \text{Similarity}(A, B) = 0 \): the two IE don’t have common characteristics between themselves;
- \( 0 < \text{Similarity}(A, B) < 1 \): the two IE are not 100% equal, but they have some common characteristics.

2.2 **Similarity Rules**

According to [42], there is a set of rules to verify the existence of similarity between IE of different models. Note that these rules are grouped by order of importance and that none of them can, by itself, take any definitive conclusion of the similarity between the IE. The relevant rules for this work are:

**Group 1 - Assured Equality**

- Rule 1 - If two IE have the same Uniform Resource Identifier (URI), they are equal;
- Rule 2 - If two IE have the same instances, they are equal.

**Group 2 - Terminological Similarity** - compares the IA’s labels.

- Rule 3 - If the name of two IE is the same or similar, it is likely that the two IE are equal or similar.

**Group 3 - Internal Structure Similarity** - compares the internal structure of the IE.

- Rule 4 - If the attributes of two IE are equal, it is likely that the two IE are also equal;
• Rule 5 - If the description of two IE is similar, it is likely that the two IE are also similar.

Group 4 - Structure and Extension Similarity - Compares the external structure of the IE (rules 6-9) and compares the instances of the IE (rules 10-12).

• Rule 6 - If the hierarchical path to the IE is equal, the compared IE are similar;
• Rule 7 - If the super entities are the same, the compared IE are similar;
• Rule 8 - If the sub entities are the same, the compared IE are similar;
• Rule 9 - If two IE have equal "sisters IE", the compared IE are similar;
• Rule 10 - If two instances have the same "mother IE", they are similar;
• Rule 11 - IE that have equal instance quantity are similar;
• Rule 12 - If two instances are connected to another instance through the same property, they are similar among themselves.

2.3 Model Matching Process To get to a correct matching we shouldn’t use one of the rules, but a process that uses various of the rules mentioned above.

In [20] and [42], it is suggested a matching process between models, which has the following steps:

(a) Given two models, we want to calculate the similarities between any pair of IE;
(b) Choose a pair of IE to be compared;
(c) Iterate, in order, for all the similarity rule groups defined above and apply, at maximum, one rule of each group;
(d) Sum weights of the rules applied;
(e) When sum of weights is higher than the cut-off (0.75), IE are considered correspondent and the process ends;
(f) If at the end of the process, the value obtained is lower than the cut-off (0.75), IE are not considered correspondent.

To each similarity rule, it is assigned a weight between 0 and 1.

At the end of this process, a matrix of similarity is obtained with the values of similarity for the IE that were compared. This process has a minimum value to conclude that two IE are similar named cut-off. By analyzing this matrix, all the pairs that have values below this cut-off are discarded. There is no consensual value in the scientific community, since it depends on the context this value needs to be applied on.

3. Conforming the Models: Once conflicts are detected, an effort is made to resolve them so that the merging of various models is possible. Automatic conflict resolution is generally not feasible; close interaction with designers and users is required before compromises can be achieved in any real-life integration activity.

4. Merging of the Models: Now the models are ready to be unified, giving rise to some intermediate integrated model(s). We have the following rules in the unification of models:

• Rule 1 - If two IE in two different models are a match, the result of the unification is a unique IE that makes reference to the two original concepts.
• Rule 2 - If an IE in a "A" model is a specialization of a B IE in a "B" model, the result of the unification is the establishment of a relation of specialization between A and B in a new "E" model.
2.9 Discussion

This chapter presented the state-of-the-art for Enterprise Information Architecture and patterns in general, as well as a practical approach to abstract concepts to a higher level, to be used in the method proposed. It also explained the UML language that will be used in this work for representing Enterprise Information Architectures, since it is a standard notation for software design and flexible.

The idea was to provide a knowledge background of the concepts used in the sections 3, 4 and 5 of this work. It hopes to serve as a guideline of thoughts on understanding the use and the importance of patterns in Enterprise Information Architectures.
Chapter 3

Solution Proposal

The main goal of this thesis is the:

*Development of a method to create an Enterprise Information Architecture Patterns and Anti-Patterns Catalog as a categorization of recurring issues and its proposed solutions.*

Thus, our work is a pattern-based approach to improve currently existing methods for future EIA projects of the PA based on previous successful projects (that have similar context). The objectives of this work are listed below:

- Identification of correspondences between IE of different information architectures and identification of IE attributes between different Enterprise Information Architectures from AMA;
- Identify common problems in this specific industry concerning Enterprise Information Architecture and extract the patterns;
- Creation of an AMA’s EIA Patterns Catalog;
- Implement solutions to this specific industry based on historical data analyzed and information necessary for optimized architecture design decisions;
- Reduce the costs for creating a Patterns Catalog, by reducing the number of human resources necessary to develop it.

![Figure 3.1: Process model of the solution proposal.](image-url)
Figure 3.1 depicts the process model of our solution proposal. First phase of the solution is to gather EIAs that verify certain conditions. They need to have similar contexts and problems, need to be designed in the same meta-model, and also need to be in the same language. The second phase is to make binary comparisons of the EIAs. Given two models with similar context, we intend to calculate the similarities between any pair of IE. We iterate, in order, for all the similarity rule groups and apply, the maximum, one rule of each group. To each similarity rule, it is assigned a weight between 0 and 1. When the sum of weights is higher than the cut-off (0.70) or equal, the IEs are considered correspondent and the process ends. In the third phase, if two IEs are similar, generation pattern rules explained in section 3.4 should be iterated and applied when the conditions are verified. In the forth phase, we classify patterns and anti-patterns. Finally, in the fifth phase, we document those patterns according to a defined structure. Figure 3.2 shows a resume of the various steps of the solution proposal.

Figure 3.2: Solution Proposal steps.
3.1 1st Phase - Gathering of EIA documents

First, we need to gather all the Enterprise Information Architecture documents that we will later analyze. In this phase, it is important to analyze different EIA documents so to find architectures that have similar contexts and functions.

The architecture needs to be designed in the same meta-model and also needs to have a similar context and a similar problem. They cannot be compared to find patterns if they don’t meet these conditions.

The architectures should preferentially be represented in UML, as explained in section 2.4.3 of this work. If not, the modeling of the information systems may be necessary in the next phase to conceptualize and to have a formal representation of the IS. The architectures can only proceed to comparison if they have the same meta-model (for this thesis, it is UML, explained in section 2.4), as well as the same context and problems. It is also relevant to mention that the two EIA that will be compared, need to be in the same language. If they are not, they should be translated to English.

In order to understand the architectures that are analyzed, it is necessary to gather information about them. In this phase, one must note that the architectures have a considerable amount of information, which requires some considerable effort at the beginning [43].

Once we have defined which Enterprise Information Architectures will be used, and once we have confirmed that they have the same context and problem, we can proceed to the second phase. Table 3.1 resumes the conditions that EIAs must have in order to be selected for comparison in this first step.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Problem</strong></td>
</tr>
<tr>
<td>The causes of the problem and the forces that must be resolved should be identified and similar between different EIAs.</td>
</tr>
<tr>
<td><strong>Context</strong></td>
</tr>
<tr>
<td>A list of the situations where the problem occurs as well as the set of circumstances or facts that surround the chosen EIAs should be similar.</td>
</tr>
<tr>
<td><strong>Language</strong></td>
</tr>
<tr>
<td>The EIAs should be in the same language. If they are not, they should be translated to English.</td>
</tr>
<tr>
<td><strong>Meta-model</strong></td>
</tr>
<tr>
<td>The EIAs, in order to be compared, need to be in the same meta-model, usually UML. If one of them is in a relational database, it must be converted to a higher level abstraction such as UML in the next phase.</td>
</tr>
</tbody>
</table>

Table 3.1: Conditions when selecting EIAs to be compared.

3.2 2nd Phase - Comparison of the EIA

In order to find patterns, we need to compare the Enterprise Information Architectures, namely the informational entities and its attributes. For the scope of this study, comparison of the informational entities of the enterprise informational architectures of AMA was carried out.

The Enterprise Information Architecture will only be compared if they have the same meta-model, and a similar context and problem. The section before explained these conditions in more detail.

Note that in the context of this work, and in order to achieve the objective of identifying IE corresponding to different models, only binary comparisons will be made, that is, only two models are compared at a time. The method to compare them is the Model Integration method by Tiago Brás [20] explained in the section 2.9 of the related work and which was adapted for the purpose of this work.

As mentioned in the section 1.4 named "Research Method", it was important to first evaluate the Model Integration method by Brás [20], described in section 2.8, and to improve it, according to our
needs. As such, we used one of the phases of Brás' method, called "Comparison of the Models" for this second phase of our method. The improvements were the addition of three more rules that were considered relevant, with a total of 15 different similarity rules. We also made a division of the similarity rules that should be used according to the model representation of the EIAs.

In this second phase of our method to compare EIA patterns, first it's necessary to define the concept of two matching IEs in this work, which is the following:

**Matching IEs:** Giving two IEs, IE1 and IE2, having a match means that for each concept in IE1, we try to find a concept in IE2 that it will be semantically similar.

Two names are correspondent if they have the same or synonymous labels, or if they have the same Uniform Resource Identifier (URI), or if the name correspondence techniques allow us to establish which are the same or similar. To verify if two IE have the same name or are similar, it's necessary to use an algorithm to calculate it, such as Levenshtein Metric, Smith Waterman, JaroMetric, among others [44].

However, before calculating any type of name similarity, it is necessary to verify if some words are synonyms or antonyms. If both words are synonyms, then these words have 100% similarity and if they are antonyms, their similarity is 0%. Also, in the case that some words prepositions or conjunctions, the similarity of these words is not calculated.

The flowchart in Figure 3.3 explains the process to discover the similarity between words in different IE, to determine if their name is similar or equal, adapted from [8]. After selecting two IE, the start of the flowchart begins by the selection of one word each of two IE, followed by the verification if a selected work is a preposition or a conjunction.

If some of the words is a conjunction or preposition, the word is discarded, selecting another combination. If they are both, then again another combination of words is selected. If it is checked that it is not a preposition or conjunction, it is verified if the words are antonyms. If they are, then it is selected another combination of word once again.

If they are not antonyms nor synonyms, the calculation of the similarity using an algorithm happens. If they are more combination of words to be analyzed, the flowchart is initialized, if not, the similarity of the IE is calculated based on the similarity of the calculated words.

![Flowchart](image)

Figure 3.3: Flowchart of the process to calculate the name similarity between IEs.

The algorithm chosen is the Jaro–Winkler distance, with the following formula: [45]
The Jaro distance $d_j$ of two given strings $s_1$ and $s_2$ is

$$d_j = \begin{cases} 
0 & \text{if } m = 0 \\
\frac{1}{3} \left( \frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m} \right) & \text{otherwise} 
\end{cases}$$

(3.1)

Where:

- $m$ is the number of matching characters;
- $t$ is half the number of transpositions.

If a name or description of the IE has more than one word, first it is calculated the similarity between each word individually, and then the global similarity. Now that the calculation of the similarity between word of IEs has been explained, the global process of the similarity calculation between IEs will be detailed.

As mentioned in the first phase, when comparing different EIAs, they must have similar context and problems. They also need to be in the same metamodel.

For this thesis, most of the architecture were in a higher level of abstraction, such as UML. But some of them are in a lower level, such as relational databases.

When choosing two architectures to compare, if they are both in a relational database, the approach to compare them is different than if they are in UML. If the two architectures chosen are in different metamodels, then the information of the enterprise that is in a database must be mapped to a higher level of abstraction, such as UML.

In the next subsections, we cover each different case. However, regardless of the metamodel used, to verify the similarity between different IEs, a set of rules has been created and adapted from the Model Integration Method by Brás, as mentioned before. These similarity rules are different according to the metamodel of the two architectures, but the process is the same.

To each similarity rule, it is assigned a weight between 0 and 1. These weights can be assigned manually or learned by programs. One of the manual techniques consists in assigning weight 1 to the first rule selected, 1/2 to the second rule selected, 1/3 to the third, and so on. This way, we can guarantee that the most important rules have a considerable bigger weight than less important rules.

As such, each rule is assigned to a group number according to the rule’s importance, where the biggest group number is the least significant. Group 1 has the most important rules, which have the value of 1.0. The similarity rules of Group 2 have the value of 1/2. Group 3 rules’ value is 1/3, Group 4 rules’ value is 1/4, and finally, Group 5 rules’ value is 1/5 = 0.2.

The similarity is quantified formally as:

- $\text{Similarity}(A, B) \in [0...1]$;
- $\text{Similarity}(A, B) = 1$ : the two IEs are equal;
- $\text{Similarity}(A, B) = 0$ : the two IEs don’t have common characteristics between them;
- $0 < \text{Similarity}(A, B) < 1$ : the two IEs are not 100% equal, but they have some common characteristics.

Affirming that one IE has a match with another using just one rule it may be proved to be incorrect. As so, to get a correct match, a process must be followed where the similarity rules are used. The process that will be followed is presented on Table 3.2.
<table>
<thead>
<tr>
<th>Order</th>
<th>Process Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Given two models with similar context, we intend to calculate the similarities between any pair of IE.</td>
</tr>
<tr>
<td>2</td>
<td>Choose a pair of IE to be compared.</td>
</tr>
<tr>
<td>3</td>
<td>Iterate, in order, for all the similarity rule groups and apply, the maximum, one rule of each group.</td>
</tr>
<tr>
<td>4</td>
<td>Sum weights of the rules applied.</td>
</tr>
<tr>
<td>5</td>
<td>When sum of weights is higher than the cut-off (0.70) or equal, the IEs are considered correspondent and the process ends.</td>
</tr>
<tr>
<td>6</td>
<td>If at the end of the process, the value obtained is lower than the cut-off (0.70), IEs are not considered a match.</td>
</tr>
</tbody>
</table>

Table 3.2: Process Iteration of the Similarity Rules.

Regarding the attributes correspondence, we consider that two IE are correspondent if at least 70% of the attributes of an IE have correspondence in the attributes of another IE.

The cut-off value considered in [20] was 75% since it was considered to be high enough to not get false positives. This value is dependent of the context, since there is not a consensual value among the scientific community.

Initially, our value was also considered to be 75%. However, due to our several iterations of the DSRM, our experiments found that the best value for our work context is 70% since it was high enough to not get false positives, but not as high as 75% and therefore, it allows to find more similarity between different informational entities.

At the end of the iteration, if the two IE are concluded to be a match, we need to generate the pattern, with will be explained in the next phase.

This iteration process has different similarity rules according to the meta-model of the two selected architectures. In the next subsections, we will cover for each case.

Figure 3.4 shows the flowchart for applying the similarity rules according to the EIAs that where gathered. If they are in the same meta-model, they can either be in a relation database or in a high abstraction level such as UML. If only one of the two EIAs to be compared is in a relational database, then reverse engineering must be used.

![Figure 3.4: Application of the Similarity Rules according to meta-model.](image-url)
3.2.1 Representation of EIA in a Relational Databases Model

As explained before, we made a division of the similarity rules that should be used according to the model representation of the EIAs.

If both the selected EIA have its informational entities and its relationships represented in a relational database model, the similarity rules that will be used to compare them will be the one shown in Table 3.3.

Note that when we refer to the same instance, we mean a relation instance, i.e., a table, with all its data, at a given time. In this context, it’s at the time the two IEs are being compared. A primary key is a field in a table which uniquely identifies each row/record in a database table. A table can have only one primary key, which may consist of single or multiple fields.

<table>
<thead>
<tr>
<th>Description</th>
<th>Group</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 1 If two IE have the same Uniform Resource Identifier, they are equal.</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>Rule 2 If two IE have the same instances, they are equal.</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>Rule 3 If the name of two IE is the same or similar, it is likely that the two IE are equal or similar.</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>Rule 4 If two IE have the same primary key, it’s likely they are equal.</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>Rule 5 If the attributes of two IE are equal, it is likely that the two IE are also equal.</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>Rule 6 If the description of two IE is the same or similar, it is likely that the two IE are equal or similar.</td>
<td>3</td>
<td>0.33(3)</td>
</tr>
<tr>
<td>Rule 7 If the hierarchical path to the IE is equal, the compared IE are similar.</td>
<td>4</td>
<td>0.25</td>
</tr>
<tr>
<td>Rule 8 If the super entities are the same, the compared IE are similar.</td>
<td>4</td>
<td>0.25</td>
</tr>
<tr>
<td>Rule 9 If the sub entities are the same, the compared IE are similar.</td>
<td>4</td>
<td>0.25</td>
</tr>
<tr>
<td>Rule 10 If two IE have equal “sisters IE”, the compared IE are similar.</td>
<td>4</td>
<td>0.25</td>
</tr>
<tr>
<td>Rule 11 If two instances have the same “mother IE”, they are similar.</td>
<td>4</td>
<td>0.25</td>
</tr>
<tr>
<td>Rule 12 IE that have equal instance quantity, they are similar.</td>
<td>4</td>
<td>0.25</td>
</tr>
<tr>
<td>Rule 13 If two instances are connected to another instance through the same property, they are similar among themselves.</td>
<td>4</td>
<td>0.25</td>
</tr>
<tr>
<td>Rule 14 If two IE have similar functions, they may be equal.</td>
<td>5</td>
<td>0.2</td>
</tr>
<tr>
<td>Rule 15 If the type of the attributes being compared are equal, the attributes may be equal.</td>
<td>5</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Table 3.3: Similarity Rules for Relational Databases.

3.2.2 Representation of EIA in an UML Model

If the EIAs are not represented in a relational database model, there is the case in which both of two EIA that will be compared could be represented in a high abstraction level model.

As such, if both EIAs have its informational entities as well as its relationships represented in a higher abstraction level model, such as UML, the similarity rules are presented in Table 3.4.
<table>
<thead>
<tr>
<th>Description</th>
<th>Group</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule 1 If two IE have the same Uniform Resource Identifier, they are equal.</td>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>Rule 2 If the name of two IE is the same or similar, it is likely that the</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>two IE are equal or similar.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule 3 If the attributes of two IE are equal, it is likely that the two IE</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>are also equal.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule 4 If the description of two IE is the same or similar, it is likely</td>
<td>3</td>
<td>0.33(3)</td>
</tr>
<tr>
<td>that the two IE are equal or similar.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule 5 If the hierarchical path to the IE is equal, the compared IE are</td>
<td>4</td>
<td>0.25</td>
</tr>
<tr>
<td>similar.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rule 6 If the super entities are the same, the compared IE are similar.</td>
<td>4</td>
<td>0.25</td>
</tr>
<tr>
<td>Rule 7 If the sub entities are the same, the compared IE are similar.</td>
<td>4</td>
<td>0.25</td>
</tr>
<tr>
<td>Rule 8 If two IE have equal “sisters IE”, the compared IE are similar.</td>
<td>4</td>
<td>0.25</td>
</tr>
<tr>
<td>Rule 9 If two IE have similar functions, they may be equal.</td>
<td>5</td>
<td>0.2</td>
</tr>
<tr>
<td>Rule 10 If the type of the attributes being compared are equal, the</td>
<td>5</td>
<td>0.2</td>
</tr>
<tr>
<td>attributes may be equal.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.4: Similarity Rules for EIA modeled in UML.

3.2.3 Representation of EIA both in Relational Databases and UML Models

In the previous sections, we covered the case in which the two EIAs to be compared are both represented in a relational databases model (section 3.2.1), or the case in which they are both represented in a high abstraction level such as UML (section 3.2.2).

We now cover the case in which the two Enterprise Information Architectures to be compared are represented in different meta-models.

If one of the EIAs is in a relational database, and the other is in a high abstraction level such as UML, we need to abstract the DBs to a higher level, where reverse engineering should be used. This includes the removal of attributes exclusive to relational databases, removal of tables which only have a meaning in relational databases, removal of foreign keys and removal of tables which contain only exclusive foreign keys. This part was explained in more detail in section 2.7.

Figure 3.5 summarizes the reverse engineering transformations from section 2.7 to be used in this phase in case all the architectures aren’t in the same meta-model, more specifically in lower levels, like relational databases.

After using reverse engineering transformations, we can proceed to compare the two different architectures, using the similarity rules described in the previous section, namely, section 3.2.2.

Note that a foreign key is a key used to link two tables together, in the context of relational databases, more specifically, SQL. A foreign key is a column or a combination of columns whose values match a Primary Key in a different table.

Following the process iteration for similarity rules as explained in Table 3.2, we can determinate if two IEs are equal. If they are, then we can proceed to the next phase, where we will generate the pattern.
Figure 3.5: Reverse Engineering transformations.
3.3 3rd Phase - Generation of the pattern

If two IE are similar, in order to generate the pattern, the following rules should be iterated and applied when the conditions are verified.

3.3.1 Pattern Generation Rules

3.3.1.1 Pattern Generation Rule 1

If two IE in different models are a match, we generate a pattern that is a unique IE based on the two original IE. If the names are the same, the pattern will have that same name. If the names of the two IE aren’t 100% equal, the name chosen will be the most generic one. This applies to the names of the IEs, and also its attributes. This is shown in Figure 3.6, where we have the IE *GenericName* in ModelA, representing an IE with a more generic title than the IE *SimilarName* in ModelB. If the names of the two IE are a match, then the generated pattern will have the most generic name between the two. However, as an exception to this rule, we can use the most specific name instead of the most generic name when it’s more relevant and useful to use the specific name instead of the generic name, according to their EIAs context.

![Diagram](image)

Figure 3.6: Rule 1 of the Pattern Generation.

3.3.1.2 Pattern Generation Rule 2

All the attributes of an IE *A* in a ModelA that have a match to other attributes of an IE *B* in a ModelB, will result in a pattern with a new IE with the attributes that had a match, as shown in Figure 3.7.
3.3.1.3 Pattern Generation Rule 3

The generated pattern will have the same relationships that are common in both IE, including the attributes that are also a match. If the names are synonyms, we choose the most generic one, as shown in Figure 3.8 where $A'$, $B'$ and $C'$ are synonyms of $A$, $B$ and $C$, respectively.
3.3.1.4 Pattern Generation Rule 4

If an IE $B$ in a $ModelB$ is a specialization of an IE $A$ in a $ModelA$, the result of the unification is the establishment of a relation of specialization between $A$ and $B$ in the generated pattern, as shown in Figure [3.9] where the IE $B$ has a relation of specialization of the IE $A$ in model $A$, and there’s a match to IE $B$ in $ModelB$.

Note that the most inclusive class in a generalization/specialization is called the superclass (or in the context of this work, super-entity) and is generally located at the top of the diagram. The more specific classes are called subclasses (in this work, we call them sub-entities) and are generally placed below the superclass/super-entity.

![Diagram](image)

Figure 3.9: Rule 4 of the Pattern Generation.

3.3.1.5 Pattern Generation Rule 5

If IE $A$ in $ModelA$ doesn’t have a match with any IE in $ModelB$, but there is a $ModelC$ where an IE $A$ is a match with IE $A$ in $ModelA$, and IE $A$ is a generalization of IE $B$, then the resulting pattern can be a new model where IE $B$ can have the same association relationships as IE $A$.

This is possible because all statements that are made about a super-entity also apply to all sub-entities. We say that sub-entities "inherit" attributes, associations, and operations from the super-entity.

Note that a generalization is the process of extracting shared characteristics from two or more entities, and combining them into a generalized super-entity. Shared characteristics can be attributes,
relationships/associations, or methods.

This rule is summarized in Figure 3.10.

![Diagram of models and pattern generation](image)

Figure 3.10: Rule 4 of the Pattern Generation.

3.4 4th Phase - Classification of the Pattern

We also need to reflect about the discovered patterns and realize if the patterns provide us relevant or additional information.

Patterns have become accepted in many different disciplines, because it is better to use patterns then to reinvent the wheel every time a problem has to be solved. That is exactly what patterns are all about, they document knowledge about solutions to recurring problems in a way that it can be reused by others having the same problem. Documenting patterns is not simply following an instruction but has to be learned and trained.

We need to verify practical aspects about the usage of these patterns and to list the consequences to use them in the architecture. To list the consequences we can use the documentation or/and the people’s experience. After that, we need to classify the consequences of the patterns. The classifications consist in two levels: positive or negative. To do the classification we must use the information about the consequences previously collected.

Here we defined patterns and anti-patterns. In this phase, it is important that collaborators with expertise in the Enterprise Information Architectures analyzed provide their validation and their conclusion regarding if the patterns are positive or negative. If the patterns are considered positive then they are classified as a pattern. If they are considered to have negative consequences, we classify them as anti-patterns.

How exactly do we distinguish if a pattern is positive or negative? We need to analyze their implications.
**Description**

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Anti-pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>is a piece of literature that describes a design problem and a general solution for the problem in a particular context.</td>
<td>describes a commonly occurring solution to a problem that generates decidedly negative consequences.</td>
</tr>
<tr>
<td>describes a particular recurring design problem that arises in specific design contexts and presents a well-proven solution for the problem.</td>
<td>is a negative solution that presents more problems than they address.</td>
</tr>
<tr>
<td>is a general, reusable solution to a common problem in a given context.</td>
<td>documents a solution to a recurring problem in a specific context, which has proven not to work in practice.</td>
</tr>
</tbody>
</table>

**Consequences**

<table>
<thead>
<tr>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
</table>

Table 3.5: Classification of Patterns and Anti-patterns

Anti-Patterns highlight the most common problems that face the software industry and provide the tools to enable you to recognize these problems and to determine their underlying causes, while a pattern describes a recurring design problem that arises in specific design contexts and presents a well-proven solution for the problem.

Table 3.5 shows the differences between a pattern and anti-pattern, using definitions by different authors. Brown et al. affirm that “an anti-pattern is a literary form that describes a commonly occurring solution to a problem that generates decidedly negative consequences. The anti-pattern may be the result of a manager or a developer not knowing any better, not having sufficient knowledge or experience in solving a particular type of problem, or having applied a perfectly good pattern in the wrong context”. He further affirms that when properly documented, an anti-pattern describes a general form, the primary causes which led to the general form; symptoms describing how to recognize the general form; the consequences of the general form; and a refactored solution describing how to change the AntiPattern into a healthier situation [37].

Collaborators of the enterprise who are more aware of the information architectures should give their perspective if they classify it as a pattern or an anti-pattern.

### 3.5 5th Phase - Documentation of the pattern in EIA Catalog

The pattern form selected for documenting patterns is important for the applicability of the patterns. Although, various pattern forms are known (some of the most popular were explained in the related work section), there are five essential elements included in all of them, which are: identification, context, problem, solution and consequences. Buschmann et al. [5] give an overview about these elements and their purpose. We adapted their documentation form. The way we organized our Information Architecture Patterns and Anti-Patterns catalog is as it follows:

1. **Pattern/Anti-Pattern**: Define if it is classified as a pattern or as an anti-pattern.
2. **Pattern’s ID**: An unique alphanumeric identifier.
3. **Identification**: Name and classification for identifying pattern. The name should be memorable, usable, and distinct. Additionally, it is important for creating a vocabulary for the domain under consideration.
4. **Context**: Situation giving rise to a problem. No problem exists in a vacuum, there is always a surrounding context. The context helps to determine where to use the pattern, and provides...
evidence that it is of general application.

5. **Problem:** Set of forces repeatedly arising in the context. They constitute the obstacle, which are in the way of producing an efficient/elegant/powerful design. Usually there are some forces in conflict. That is why a pattern is required to solve them. For example, try optimizing code for speed, memory usage, and maintainability. This results in three conflicting forces. Some pattern forms reduce the problem to a single question or a summarizing formulation of the problem.

6. **Solution:** Configuration to balance the forces. The solution is a description of the elements of the solution design, their responsibilities, relationships, and collaborations. It is not a concrete design or implementation, but a general, tailorable solution.

7. **Consequences:** Consequences arising from application of the pattern. These are the results and trade-offs of applying the pattern, including the advantages and disadvantages.

Additionally, the documentation can also have the following optional items:

- **Alias:** Names the EIA pattern is also known as;
- **Summary:** A short summary of the EIA pattern;
- **Version:** Version number of the EIA pattern.

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Context Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>The context helps to determine where to use the pattern, and provides evidence that it is of general application.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set of forces repeatedly arising in the context.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solution Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detailed description of the EIA pattern</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Consequences Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consequences resulting from the usage of the EIA pattern</td>
</tr>
</tbody>
</table>

Figure 3.11: Structure of the documented patterns.

**Figure 3.11** shows the EIA patterns form. Adapting the different methods mentioned in the related work section to specifically compare information entities and Enterprise Information Architectures is a way to create a catalog with multiple patterns and anti-patterns for information architecture design and development that can be used by specific industries such as AMA to reduce costs and avoid common problems that have been documented and experimented.

### 3.6 Discussion

This chapter presented the solution for the problem description detailed in section 1.2, which is a method to generate Enterprise Information Architecture patterns and create an EIA patterns catalog.
The first phase of the solution is to gather EIA documents that verify certain conditions. They need to have similar contexts and problem. The architecture also needs to be designed in the same meta-model and also needs to be in the same language, preferable English.

The second phase is to make binary comparisons of the EIA. Given two models with similar context, we intend to calculate the similarities between any pair of IE. If one of the EIA is in a relational database, and the other is in a high abstraction level such as UML, we need to abstract the DBs to a higher level, where reverse engineering is used. If two IE are similar, in order to generate the pattern, generation pattern rules explained in section 3.3 should be iterated and applied when the conditions are verified. In the forth phase, we classify patterns and anti-patterns. Finally, in the fifth phase, we document those patterns according to a defined structure presented in section 3.5.

By applying these steps in EIA with similar context, we proposed to create an EIA patterns catalog for AMA.
Chapter 4

Demonstration

This chapter corresponds to the demonstration activity of DSRM process model.

The demonstration activities aim to demonstrate the use of artifacts to solve the one or more instances of the research problem [19]. Now that the solution proposal was addressed in the previous chapter, a demonstration is now going to be made to clarify it.

In this demonstration, an instantiation of the solution proposal of this work is made to the AMA sector of the Portuguese government, with the final result being a catalog of patterns. However, due to the documents given by AMA being very specific and also with high complexity, an instantiation will be first made using an academic example.

4.1 Demonstration with an Academic Example

In this demonstration, an academic example is used to show the several steps of the solution proposal (chapter 3).

For the 1st phase described in the solution’s proposal section, we needed to select and analyze Enterprise Information Architectures with similar contexts. In this case, let’s suppose that Figure 4.1 is part of a Model A, representing an EIA in relational database tables, and Figure 4.2 is part of a Model B, representing another EIA where both of them have similar problem and context, verifying the conditions described in section 3.1 (first phase of the solution proposal). It is also verified that they are in the same language, in this case, English.

Next, we proceed for the second phase of the solution proposal. Given the five database tables in Figure 4.1, we want to obtain a model that represents an abstract concept of these tables and its relationships. In the tables, there are four columns, where the first one is the field number and the second is the name of the field. The third column means that if a certain field is a Primary Key (PK) or not. The last column indicates if a certain field is or not a Foreign Key (FK). The table named entity is a generalization concept of Organization and Person.

From the tables given in Figure 4.1, we can conclude that they are represented in a different model than Figure 4.2, so it is necessary to use reverse engineering to abstract this model to a higher concept model, such as UML. To do it so, we need to find the correlations between them, as well as the fields that represent primary or foreign keys, which attributes should be removed in a higher abstraction level, and so on. The reverse engineering techniques are described in section 3.2.3 of this work.

Each table is going to be represented as an IE. The following transformations are made:

- Foreign key attributes are removed (for example, attribute addressID of table Entity and attribute contactID of table Organization);
Figure 4.1: Database tables of Model A from the academic example.

- Attributes that only have a meaning at the level of databases are removed (for example, added-ByLDAPImport attribute from table Person;

- If there are tables containing only foreign keys, they are removed;

- Tables who only have a meaning at databases level are also removed.

After the identification and realization of these transformations, we are going to use the UML language to abstract these tables, where the resulting model A is presented in Figure 4.3.

Now that we have both of these EIAs are in the same meta-model, we proceed to the comparison between the two models. Continuing the second phase of our solution proposal, we could say that, by simply analyzing the models in Figure 4.2 and Figure 4.3, it is possible to estimate that the IE pairs A.Entity/B.Entity and A.Address/B.Location are the same, and that the IE pair A.Person/B.Individual and A.Organization/B.Group are very similar. However, we must demonstrate that this idealization is true.

- **For pair A.Entity/B.Entity:** Iterating through the groups of the similarity rules defined in 3.2.2, the first rule that can be applied is Rule 1 - *If two IE have the same Uniform Resource Identifier (URI), they are equal.* We count the weight of this rule (1.0) in a variable starting at zero. Since the variable is now 1.0 which is superior to the cut-off (0.70), the iteration stops and we conclude that the two EI are correspondent.

- **For pair A.Address/B.Location:** Iterating through the groups of the similarity rules defined in 3.2.2, the first rule that can be applied is Rule 2 - *If the name of two IE is the same or similar, it is likely that the two IE are equal or similar.*, since they are synonyms. We count the weight of
this rule (0.5) in a variable starting at zero. Since the value accumulated in this variable does not exceed the cut-off established of 0.70, we must continue the iteration. So, the second rule to apply is the Rule 3 - If the attributes of two IE are equal, it is likely that the two IE are also equal. - since more than 70% of the attributes are either synonyms or have more than 70% string similarity using Jaro Winkler formula. By adding the weight of this rule (0.5) to the variable initialized earlier, we come to the value 1.0. Since this value already exceeds the cut-off of 0.70, we consider that the two IE are a match and the correspondence process comes to an end.

• For pair A.Person/B.Individual: Iterating through the groups of the similarity rules defined in 3.2.2, the first rule that can be applied is Rule 2 - If the name of two IE is the same or similar, it is likely that the two IE are equal or similar., since they are synonyms. We count the weight of this rule (0.5) in a variable starting at zero. Since the value accumulated in this variable does not exceed the cut-off established of 0.70, we must continue the iteration. So, the second rule to apply is the Rule 3 - If the attributes of two IE are equal, it is likely that the two IE are also equal. - since more than 70% of the attributes are either synonyms or have a value superior to 0.70 value of string similarity using Jaro Winkler formula. By adding the weight of this rule (0.5) to the variable initialized earlier, we come to the value 1.0. Since this value already exceeds the cut-off of 0.70, we consider that the two IE are correspondent and the correspondence process comes to an end. Another option instead of rule 3, would be Rule 9 - If two IE have similar functions, they may be equal., since both A.Person and B.Individual represent the same concept of a single human with the same functionality. So, adding the weight of this rule (0.2) to the previous rule number 2, the total would be 0.70, which equals the cut-off, and they are considered a match and the process comes to an end.

• For pair A.Organization/B.Group: Iterating through the groups of the similarity rules defined in 3.2.2, the first rule that can be applied is Rule 2 - If the name of two IE is the same or similar, it is likely that the two IE are equal or similar., since they are synonyms. We count the weight of this rule (0.5) in a variable starting at zero. Since the value accumulated in this variable does not
exceed the cut-off established of 0.70, we must continue the iteration. So, the second rule to apply is the rule 3 for the same reasons as mentioned previously. Other rules could also apply, but since the cut-off value is reached, the two IE are considered a match and the process ends.

- **For pair A.Contact/B.LicenseRequests:** There are no rules that apply to these IE pair, so we conclude that they are not a match.

Finally, we can now generate the pattern, which is the third phase of our solution proposal. We apply the pattern generation rules defined in the section 3.3.1, such as Rule 1 - *if two IE in different models are a match, we generate a pattern that is a unique IE based on the two original IE*. If the names are the same, the pattern will have that same name. If the names of the two IE aren’t 100% equal, the name chosen will be the most generic one. This applies to the names of the IEs, and also its attributes.

We need also to apply Pattern Generation Rule 2 - *All the attributes of an IE A in a ModelA that have a match to other attributes of an IE B in a ModelB, will result in a pattern with a new IE with the attributes that had a match*. - and Rule 3 *The generated pattern will have the same relationships that are common in both IE, including the attributes that are also a match*. The attributes who had a match are the ones that in the previous phase were concluded to be synonyms or to have more than 0.70 in Jaro Winkler similarity.

The other Pattern Generation Rules presented in section 3.3.1 don’t apply to this case.

The resulting pattern is represented in **Figure 4.4**.

Next, we proceed to the forth phase. We need to analyze if this pattern has a positive or negative consequences. Analyzing table 3.5 from section 3.4, we conclude that this pattern should be classified
has a positive pattern and not an anti-pattern. In this phase, it is important that collaborators with expertise in the Enterprise Information Architectures analyzed provide their validation and their conclusion regarding if the patterns are positive or negative. After that, we proceed to the final phase, which is the documentation of this pattern in our catalog.

The pattern form selected for documenting patterns was described in section 3.5 of this work. As such, the documentation of the generated pattern in the patterns catalog is the following:

**Enterprise Information Architectural pattern 1**

- **Pattern/Anti-Pattern**: Pattern.
- **Pattern’s ID**: P-1
- **Pattern’s name**: Entity-Address Pattern.
- **Context**: The entity can be a organizational unit that has an address for the company’s location, or an individual with a specific address.
- **Problem**: An entity, either be a person or an organization, needs to keep and update the data of its locations.
- **Consequences**: Keeps record of the entity’s data regarding its address in the system. However, if this system fails, there isn’t a system back-up.
- **Solution**: The pattern’s objects are in Figure 4.4, which are:
  - Object – Entity: can be a person or a group.
  - Object – Address: An address is the information regarding the location of a group or an individual, which is kept in the information system of an organization.
  - Entity has Address: The Entity unit uses Address to support its activities.

Additionally, the patterns catalog can also have a summary for each pattern, as well as its version and alias. All of them are optional.
4.2 AMA Demonstration

In this section, the solution proposal is instantiated to the Portuguese Government sector, namely AMA, which culminates in the AMA EIA Patterns Catalog. The logic and the reasoning is the same as the previous demonstration (section 4.1). However, the instantiation described in this section doesn’t intent to be as extensive, due to the fact that it is very unpractical to exhaustively detail all the specifics.

In order to facilitate the understanding, this instantiation will be structured as it follows:

- 1st Phase: Gathering of AMA’s EIA documents;
- 2nd Phase: Comparison of AMA’s EIAs;
  - Similarity Rules to be applied according to the model representation of EIAs;
  - Use of Reverse Engineering when applied;
  - Process Iteration of the Similarity Rules;
  - Verification if the words are synonyms or antonyms during name similarity comparison;
- 3rd Phase: Generation of AMA patterns;
- 4th Phase: Classification of the AMA patterns;
- 5th Phase: Documentation of the patterns in the AMA EIA Patterns Catalog.

4.2.1 Gathering of AMA’s EIA documents

In this first step, the strategy of section 3.1 will be followed. The three documents that were gathered and represent EIA of different sections of AMA, were explained in 2.6, in the related work. They are the following:

- **Citizen Portal** (Portal do Cidadão, in Portuguese): It’s a channel to the services provided by the public administration, where the citizens can also consult information online. Its Enterprise Information Architecture is represented in a SQL database model. Annex B presents the Liferay database used by AMA for Portal do Cidadão. However, it is not possible to show its whole relational database, since it is very extensive;

- **Entrepreneur Counter** (Balcão do Empreendedor, in Portuguese): It’s the point of access to the services related to business activity, where the entrepreneurs can have access to various types of contents. Its EIA is represented by a data model which uses the UML standard. It’s not possible to present a model of the EIA of Balcão do Empreendedor due to its extensiveness;

- **AMA’s Information Architecture**: It’s the document where the model of AMA’s entities, its relationships and specializations is. It’s in a high abstraction level of representation. Annex A depicts a model of AMA’s Informational Architecture. However, the attributes for each entity are not present due to being very comprehensive.

Since there are 3 different EIAs, and since the generation of EIA patterns are made using binary comparisons, our solution proposal was tested in a combination value of \( ^3C_1 = 3 \) different cases. That means that we tested and demonstrated our solution in three distinct cases: comparison of Portal do Cidadão with Balcão do Empreendedor; AMA’s Information Architecture with Portal do Cidadão and Balcão do Empreendedor with AMA’s Information Architecture.

We needed to see if they verify the conditions described in section 3.1. They have similar contexts and problems. However, they are not in the same language: Portal do Cidadão is in English, and Balcão...
do Empreendedor and AMA’s Information Architecture are both in Portuguese. As such, according to table 3.1 of section 3.1, we needed to translate them to English, so that we can compare them in the next phase. After the translation, we proceeded to the second phase. The other condition is that all EIA need to be in the same meta-model. Since the EIA of Portal do Cidadão is represented in a relational database model, more specifically in SQL language, and both Balcão do Empreendedor and AMA’s Information Architecture are represented in a high abstraction level, such as UML. So we need to use reverse engineering in the next phase in order to abstract the concepts of Portal do Cidadão.

4.2.2 Comparison of AMA’s EIA documents

In this phase, we follow the strategy described in section 3.2. Each table in the database of Portal do Cidadão was conceptually represented as an IE. This action doesn’t apply to Balcão do Empreendedor or to AMA’s Informational Architecture, because only Portal do Cidadão is modeled in a relational database, as mentioned before.

According to the flowchart in Figure 3.3 of section 3.2 of this work, the EIAs gathered aren’t in the same meta-model, so we need to use reverse engineering that was described in section 3.2.3. We proceed to reverse engineering. First we removed the tables that only had a meaning in DB’s language. An example is in Figure 4.5 where tables ACT_GE_BYTEARRAY, ACT_GE_PROPERTY and ACT_HI_ACTINST of Portal do Cidadão of AMA verify this condition. This is only a short example, since there were many tables such as these.

We also removed database tables who exclusively contained only foreign keys. An example of this case can be seen in Figure 4.6 where table SCLicensesCPProductEntries only has attributes named licenseId and productEntryId, which are foreign keys. Since the table doesn’t have other attributes besides foreign keys, this table was removed during the reverse engineering process.
We also applied the strategy of the removal of foreign key attributes in tables. Figure 4.7 depicts an example of this case. UserGroup table had the attributes named companyId and parentUserGroupId which were foreign keys. These attributes were removed.

Finally, we reached the final step of the reverse engineering process, which is to model these tables, attributes and relationships to UML language.

This was an extensive work, so we show only some examples. This step can be easily understood by reading the academic example from section 4.1.

Figure 4.8 depicts an example of a table, named Group, in the relational database of Portal do Cidadão. Figure 4.9 then shows the transformation after being modeled to UML language.

Only the attributes of table Group are shown in the example, since the table Company, creatorUser, className, liveGroup and parentGroup are not relevant in this example.

This demonstrates that the final result of the reverse engineering steps are the UML’s models of AMA’s relational databases.
After the reverse engineering steps, we were able to compare the different EIA’s. We followed the strategy in section 3.2, where the similarity rules were applied using an iteration process. For example, when comparing the IE Group_ in Figure 4.9 from Portal do Cidadão with Figure 4.10 from Balcão do Empreendedor, we needed to iterate through the groups of the similarity rules defined in 3.2.2, where the first rule that could be applied was Rule 2 - If the name of two IE is the same or similar, it is likely that the two IE are equal or similar, since they are synonyms.

The Jaro-Winkler distance formula also confirms the string similarity, since the value between the names Groups_ and Group_ is 0.93. We count the weight of this rule (0.5) in a variable starting at zero.
Since the value accumulated in this variable does not exceed the cut-off established of 0.70, we must continue the iteration.

As so, the second rule to apply is the Rule 9 - If two IE have similar functions, they may be equal. - since both of the IE represent the concept of an organization of multiple entities or people in AMA.

By adding the weight of this rule (0.2) to the variable initialized earlier, we come to the value 0.70. Since this value already reached the cut-off of 0.70, we consider that the two IE are a match and the correspondence process comes to an end.

![Figure 4.10: IE Grupos from Balcão do Empreendedor translated to English, because of the conditions necessary in phase 1 of the solution proposal.](image)

After comparing the IEs from the different EIA’s gathered from AMA’s documents, we proceeded to the third phase of our solution proposal, in which the IEs who were considered a match, were generated into an AMA pattern.

### 4.2.3 Generation of AMA’s patterns

We followed the strategy explained in section 3.3.

When comparing each IE individually between different AMA’s EIA’s, we applied the generation rules described in section 3.3.1.1, 3.3.1.2, 3.3.1.3, 3.3.1.4 and 3.3.1.5.

Following the example of the IEs Group_ and Group from the previous section, we applied the Pattern Generation Rule 1 of section 3.3.1.1, which says that If two IE in different models are a match, we generate a pattern that is a unique IE based on the two original IE. If the names are the same, the pattern will have that same name. If the names of the two IE aren’t 100% equal, the name chosen will be the most generic one. This applies to the names of the IEs, and also its attributes.

We also applied the Pattern Generation Rule 2: All the attributes of an IE A in a ModelA that have a match to other attributes of an IE B in a ModelB, will result in a pattern with a new IE with the attributes that had a match. The other pattern generation rules don’t apply to this example. The resulting pattern from this example is in Figure 4.11.

Note that this is a simple example of the process for generating patterns of AMA, due to the fact that this process was very broad and it is difficult to show them all. After each pattern that was generated, we proceeded to phase 4 of the solution proposal.

Some examples of patterns of AMA that were generated in this phase are the Entity Generalization Pattern, Email-Contact Pattern, Entity-Licenses Pattern, UserGroups-Message Pattern, FormRequests Anti-Pattern, Address Anti-Pattern, and so on.
4.2.4 Classification of AMA’s patterns

In this fourth phase, we followed the strategy detailed in section 3.4 of this work. We needed to verify practical aspects about the usage of these patterns and to list the consequences to use them in the Enterprise Information Architecture of AMA.

Following table 3.5 of section 3.4, we decided for each pattern if they were an anti-pattern or a pattern, according to their positive or negative consequences. We also had reunions with two AMA’s collaborators to analyze if these patterns were a general, reusable solution to a common problem in a given context or if they were a negative solution that presented more problems than they address and classified those patterns according to the conditions previously mentioned.

Following the Group Pattern example from the previous section, this was considered a positive pattern, and therefore, classified as a Pattern, since AMA’s collaborators agreed that this was a common concept used in the Enterprise Information Architecture and that it is considered a solution that works well, as experienced in the previous projects. After the classification of AMA’s patterns, we proceed to the fifth and final phase of the solution proposal.

4.2.5 Documentation of AMA’s patterns

Finally, for this final phase of the demonstration, we followed the strategy in section 3.5. The AMA’s EIA Patterns Catalog was structured based on the fundamental items explained in section 3.5.

The patterns in the catalog have an overview section, where the Id, name, alias, a short summary and the version of each pattern is presented. Then, there is the context section which helps to determine the use of the pattern in which situations and environment.

It is followed by the problem section, which described the set of forces repeatedly arising in the context. After that, there is the solution section, where a detailed description of the EIA pattern is shown. And the last section are the consequences, which result from the use of that same pattern.

Regarding the Group Pattern example from the previous section, for demonstration purposes, the documentation of this pattern is shown in Figure 4.12, in which the ID of the pattern is P-1 and its name is Group Pattern. The alias, summary and version (1.0) is also indicated. The context section explains that this pattern was generated when comparing the Portal do Cidadão EIA model and Balcão do Empreendedor EIA model. The solution section contains the pattern modeled in UML. There is also a problems section and the consequences of using this pattern.

This final phase concludes the solution proposal presented in this thesis. Using the method proposed, the final result is AMA’s EIA Patterns catalog to be used in future projects that have similar contexts.
Some improvements had to be made to the method of the solution proposal due to small issues encountered and in order to optimize the discovery of patterns and anti-patterns. This was explained in more detail in section 1.4, named Research Method.

The final result, AMA’s EIA Patterns Catalog, has a collection of more than 40 pages of patterns and anti-patterns.

4.3 Discussion

This chapter had the purpose of helping with the understanding of the solution proposal of this work, in two different manners. The first demonstration was an instantiation with an academic example to help understanding, in a not very complex approach, all the steps of the solution proposal.

The second demonstration was an instantiation of the solution proposal in the Portuguese Government, more specifically, AMA, where the method to find patterns was applied to a set of IS of AMA, such as the Citizen Portal and the Entrepreneur Counter and some examples of the solution steps were shown and explained.

Figure 4.12: AMA example of a generated pattern documented in AMA’s EIA Pattern Catalog.
Chapter 5

Evaluation

This section corresponds to the evaluation activity of the DSRM process model. The evaluation activities aim to observe and measure how well the artifacts support a solution to the research problem [19]. This activity intends to compare the objectives of a solution to actual observed results from use of the artifacts presented in section 4.

A very convenient and practical way to evaluate an IA or EIA catalog model obtained (and consequently to validate our hypothesis) is to compare it with a prior EIA catalog model previously developed for the same practical case, but using another methodology/approach. This should allow us to make a direct comparative analysis, and understand the quality of our EIA Patterns Catalog, which can be considered a model.

AMA recently created a Services Catalog to serve as a directory of information about the different services that AMA provides. We used it to compare with our AMA’s EIA Patterns Catalog. Figure 5.1 shows part of AMA’s Service Catalog. This Service Catalog was created without using any specific methodology; it was verbally defined in meetings with different AMA stakeholders.

![Figure 5.1: Part of AMA’s Service Catalog.](image)

The metric variables used for comparing these two different methodologies are:
• Number of people needed to create the catalog;
• Optimal number of people needed to create the catalog.

We surveyed some AMA’s key stakeholders about how many people did it take to create the Services Catalog. It took them five meetings with technical teams of 5 or 6 elements, and more than six meetings with functional teams consisting of more than 6 elements. The total is approximately 12 elements to create AMA’s Service Catalog. This thesis solution proposal only needs one person to generate the patterns. However, during the fourth phase of the solution proposal, section 3.4 of this work, named “Classification of the Patterns”, the optimal number of people would be 3, in order to validate the patterns generated and to use their expertise in the analysis of the pattern. The conclusion is that the necessary number of people to create a patterns catalog is significantly smaller using our work methodology, which means it needs less human resources. Table 5.1 resumes this analysis.

<table>
<thead>
<tr>
<th>Metric Variables</th>
<th>AMA’s Service Catalog</th>
<th>Our AMA’s EIA Patterns Catalog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimal number of people needed to create the catalog</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>Optimal Number of people needed to create the catalog</td>
<td>12</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 5.1: Analysis of the number of people needed to create each catalog.

Another evaluation process used in our work is inspired in the Moody & Shanks Framework [46], which defines quality factors for a data model. In the data model quality management framework, the authors defined the quality factors that are decisive factors for a model evaluation which are the ones in Figure 5.2 The assessment of these quality factors is made for our AMA’s EIA Patterns Catalog.

Figure 5.2: The data model quality factors by Mood & Shanks.

The objective or purpose of each quality is the following:
1. **Completeness**: refers to whether the data model contains all user requirements.
2. **Simplicity**: means that the data model contains the minimum possible entities and relationships.
3. **Flexibility**: is defined as the ease with which the data model can cope with business and/or regulatory change.
4. **Integration**: is defined as the consistency of the data model with the rest of the organization’s data.
5. **Understandability**: is defined as the ease with which the concepts and structures in the data model can be understood.
6. **Implementability**: is defined as the ease with which the data model can be implemented within the time, budget and technology constraints of the project.

These analysis factors were defined taking in consideration our work’s context, and the concrete issues that we need to address in order to validate it. These factors intend to discuss how the final results are according to those we excepted.

Analyzing each different methodology to create the patterns catalogs considering each factor of Moody & Shanks Framework, we can conclude that:

1. **Completeness**: Comparing AMA’s Service Catalog and this thesis EIA’s Patterns Catalog, we can conclude that AMA’S Service Catalog only has informational entities regarding services and focus only on the type of services, it’s taxonomies, point of cares, documents and images. Our thesis Patterns Catalog is considered to be slightly more complete since it has informational entities concerning a holistic view of AMA’s EIA. Besides informational entities regarding AMA’s services, it also has informational entities regarding the different types of entities and it’s specializations, it’s attributes such as contact, addresses, different types of contacts, and so on. It also contains anti-patterns, which are not present in AMA’s Service Catalog.

2. **Simplicity**: Due to the structure of this thesis’ catalog, the EIAs models are divided into smaller patterns, and documented according to the informational entities types. So this thesis’ is simpler than AMA’s Service Catalog, since they have more relationships between informational entities.

3. **Flexibility**: Both catalogs are similar regarding the flexibility factor.

4. **Integration**: Both catalogs are similar regarding the integration factor.

5. **Understandability**: This thesis’ catalog has a structured documented form, in which each pattern is document according to the informational entity types and descriptions, as well as its context, problems, solutions and consequences. AMA’s Service Catalog isn’t documented, so this thesis catalog is easier to understand.

6. **Implementability**: Both catalogs are similar regarding the implementability factor.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>AMA’s Service Catalog</th>
<th>This thesis AMA’s EIA Patterns Catalog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completeness</td>
<td>-</td>
<td>Slightly more complete</td>
</tr>
<tr>
<td>Simplicity</td>
<td>-</td>
<td>Simpler</td>
</tr>
<tr>
<td>Flexibility</td>
<td>Equal</td>
<td>Equal</td>
</tr>
<tr>
<td>Integration</td>
<td>Equal</td>
<td>Equal</td>
</tr>
<tr>
<td>Understandability</td>
<td>-</td>
<td>Easier to understand</td>
</tr>
<tr>
<td>Implementability</td>
<td>Equal</td>
<td>Equal</td>
</tr>
</tbody>
</table>

Table 5.2: Comparison of the catalogs according to quality factors of Moody & Shanks.

After this analyses, we can conclude that this thesis’ patterns catalog is more complete, simple and easier to understand than AMA’s Service Catalog, for the reasons mentioned above. Table 5.2 resumes this analyses.
5.1 Discussion

In this chapter we presented the case studies evaluation. The evaluation process was based on the combination of the Moody and Shanks Framework with the comparison of a similar Catalog model made without a specific methodology.

The results were positive, showing that this thesis solution proposal needed less human resources to generate patterns and create a catalog to be used in future projects, and is also better documented, easier to understand and more complete.

Table 5.3 presents a summary of what was adapted from the Model Integration method of Tiago Brás for this thesis (as mentioned in section 2.8), showing the changes made, like the addition of three new rules that we considered to be lacking, and the improvements, so we can distinguish the main differences.

Note that Brás's work isn't about patterns, but about creating a Reference Architecture for PA.

<table>
<thead>
<tr>
<th>Model Integration Method</th>
<th>This thesis’ method</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total Number of Similarity Rules</strong></td>
<td>12</td>
</tr>
<tr>
<td><strong>Added Rules</strong></td>
<td>-</td>
</tr>
<tr>
<td><strong>Phases used</strong></td>
<td>all</td>
</tr>
<tr>
<td><strong>Phases removed</strong></td>
<td>-</td>
</tr>
<tr>
<td><strong>Cut-off Value</strong></td>
<td>75%</td>
</tr>
<tr>
<td><strong>Other improvements</strong></td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.3: Differences between our work method and Brás' method.

The Model Integration Method by Brás has four phases total, which are the following: PreIntegration, Comparison, Conforming and Merging of the models, which were explained in more detail in the section 2.8 of this work.

This Thesis only used the Comparison phase of the Model Integration Method and improved it. It had a total of 12 similarity rules, where this thesis has a total of 15 different similarity rules.

The three added rules where rules number 4, 14 and 15 of table 3.3. Another improvement was the division of the table according to the model representation of the EIA’s.

Another mentioned work in section 2.8 was Yesika’s method to create patterns [8]. Her work is focused in patterns for all the layers of Enterprise Architecture, such as the technology layer, the business layer, the applications layer, and so on.

Our work focus specifically on Enterprise Information Architecture.

One of the main differences is that Yesika’s work proposed a for generating patterns in EA, and didn’t have a final catalog with the generated patterns for a specific area. Also, this work has specific generation rules for creating patterns.

Table 5.4 resumes the main differences between the works.
<table>
<thead>
<tr>
<th>Yesika’s Work</th>
<th>This thesis</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Context</strong></td>
<td>Hollistic view of Enterprise Architecture</td>
</tr>
<tr>
<td><strong>Number of Phases</strong></td>
<td>4</td>
</tr>
<tr>
<td><strong>Proposition of a method to create patterns</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Catalog with patterns</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Comparison between different EAs</strong></td>
<td>Lexical comparison of EA</td>
</tr>
<tr>
<td><strong>Generation Rules</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Other differences</strong></td>
<td>Doesn’t include a method to transform two different EA into the same meta-model</td>
</tr>
</tbody>
</table>

Table 5.4: Comparison between our work method and Yesika’s.
Chapter 6

Conclusion

6.1 Results Achieved

We believe that the solution proposal described in this work adds value to its surrounding context: the development of a method to generate EIA patterns and the creation of a catalog with those patterns, for the Portuguese Public Administration.

The main contributions of this work are directly related to the main questions in the section 1.2 of this work and to the objectives in section 3. As such, it is important to relate these two components.

• To the main question "Is it possible to create a catalog of Information Architectures patterns to be used by the Portuguese PA, specifically AMA?" is related with the objective Creation of a AMA's EIA Patterns Catalog and implement solutions to this specific industry based on historical data analyzed and information necessary for optimized architecture design decisions:

• The question "How do we compare different Enterprise Information Architectures?" is related with the following objectives: Identification of correspondences between IE of different information architectures and identification of IE attributes between different Enterprise Information Architectures from AMA;

• The question "How can we extract patterns and anti-patterns?" has a correspondence with the following objective: development of a method to create an Enterprise Information Architecture Patterns and Anti-Patterns Catalog as a categorization of recurring issues and its proposed solutions.

• The question "How do we document Enterprise Information Architecture patterns and anti-patterns?" is related with the objective Creation of a AMA's EIA Patterns Catalog.

We describe next the way each of the defined objectives were accomplished, by giving, at the same time, the answer to the questions matched with them:

• Development of a method to create an Enterprise Information Architecture Patterns and Anti-Patterns Catalog as a categorization of recurring issues and its proposed solutions: The 5-phased method proposed in chapter 3 allows us to create an EIA Patterns Catalog with the generated patterns, in which the first phase of the solution is to gather EIA documents that have similar contexts, problems, and are in the same meta-model and language. The second phase is to make binary comparisons of the EIA, in which given two models with similar context, we calculate the similarities between any pair of IE. We Iterate, in order, for all the similarity rule groups and apply one rule of each group. To each similarity rule, it is assigned a weight between
0 and 1. When sum of weights is higher than the cut-off (0.70) or equal, the IEs are considered a match and the process ends. In the third phase, if two IE are similar, generation pattern rules explained in section 3.4 should be iterated and applied when the conditions are verified. In the forth phase, we classify patterns and anti-patterns. Finally, in the fifth phase, we document those patterns according to a defined structure. Evaluation in section 5 shows that this method requires less human resources, with the minimum been 1 (while AMA's Service Catalog made without a specific methodology required 12 human resources) and it is better documented. The final result was an EIA Patterns Catalog for AMA;

- Identification of correspondences between IE of different information architectures and Identification of IE attributes between different Enterprise Information Architectures from AMA: With the second phase of the solution proposal, in section 3.2, named Comparison of the EIA, these objectives were accomplished with the matching process that was defined using a total of 15 similarity rules (8 regarding structure and extension similarity between IEs, 4 about internal structure, 2 about assured equality and 1 regarding terminological similarity) to compare the different informational entities and its attributes and relationships;

- Identify common problems in this specific industry concerning Enterprise Information Architecture and extract the patterns: This objective has also been accomplished, with the third phase of our solution proposal, in section 3.3, which has 5 generation rules, in which we define, for example, that the pattern will have the same name, attributes, relationships that were considered a match between the IEs during the comparison phase, and with the forth phase, in section 3.4, where we propose a method to extract the patterns and classify them as a pattern if they have positive consequences or anti-pattern, if it’s a bad a solution;

- Implement solutions to this specific industry based on historical data analyzed and information necessary for optimized architecture design decisions: It has been possible to achieve this objective with the final artifact of this research, which is the AMA's EIA Patterns Catalog, which has a total of 41 patterns, including 3 anti-patterns, and where each pattern is documented according to their name, type, context, problem to solve, consequences, among other optional sections;

- Reduce the costs for creating a Patterns Catalog, by reducing the number of human resources necessary to develop it: This final objective has also been achieved, where the chapter 5 of our paper concluded that our methodology only needs, at minimum, 1 person to develop a Patterns Catalog, while the current Service's Catalog of AMA created without a specific methodology, needed 12 people. Our EIA Patterns Catalog also serves as a guideline to avoid common mistakes in future projects of AMA (with similar contexts), by not using the anti-patterns documented in the catalog.

6.2 Communication

This section corresponds to activity 6 of DSRM methodology - Communication.

Considering this investigation’s scope, a paper has been accepted and presented in 8º INForum - Simpósio de Informática, realized in Instituto Superior Técnico, Universidade de Lisboa, in September 9th, 2016, in Lisbon, where this work's solution proposal regarding comparing different Enterprise Information Architectures and extracting patterns was addressed [17].

Furthermore, this document contributes to the communication of the investigation that has been made, since it presents, to the scientific community, a new study component for work to be developed in the future.
6.3 Lessons Learned and Final Thoughts

Patterns have become accepted in many different areas and described as very useful because they reuse knowledge already acquired from experienced users and they capture and document proven practices. The importance of information in the Portuguese PA and Government, the lack of specific methodologies for improving and developing the Enterprise Information Architectures and the nonexistence of a methodology for creating an EIA patterns catalog in the Portuguese Public Administration regarding future projects with similar context, lead us to need for the Portuguese Public Administration to have a catalog of EIA patterns to be used as the basis for future projects that have akin context with former PA projects.

A solution presented is the development of a method to create patterns and anti-patterns for Enterprise Information Architecture and the creation of a catalog to be used by the Portuguese Government, more specifically AMA, since patterns are a way to document proven practice solutions for recurring problems in a given context and proved to be a valuable approach for documenting knowledge.

Chapter 1 defined and described the Design Science Research Methodology that was applied on to support our work on scientific basis, and therefore guarantee its validity and continuity. We also clarified the problem and established the motivation towards the solution. The fundamental concepts and related work were explained on chapter 2.

The solution objective’s, mentioned above, and solution architecture were described in chapter 3, where the Model Integration method developed by Brás [20] was adapted. We explained that our process has five phases: gathering of the EIA documents (where we analyze and select IAs with similar metamodel and context), comparison of the EIA (we use an iteration process with 15 rules of similarity, with different weights, to conclude if each pair of IE is correspondent or not), generation of the pattern (we use 5 rules to generate the pattern and classify it as a pattern or anti-pattern), classification as a pattern or anti-pattern, and finally, the documentation of the pattern, where we write our pattern form for our AMA’s EIA Patterns Catalog.

On chapter 5, a demonstration was performed. An academic example was used and the process of creating AMA’s EIA patterns catalog was explained.

Finally, on chapter 6 we showed how this work was evaluated, by using assessed criteria to make a comparison with other service catalog by AMA. In Appendix A, we present AMA’s Information Architecture. Appendix B contains the Liferay Database that was used in Portal do Cidadão’s database, and appendix C contains the final result: AMA’s EIA Patterns Catalog.

By using the knowledge gathered under the thesis development, and by applying it into the development of the work during various iterations, we propose a methodology for generating EIA patterns for the Portuguese Government, that will serve as a guide for the PA in order to achieve a consistent and optimized Enterprise Information Architecture in future projects with similar context.

6.4 Limitations

Along with the contributions that our work was able to give, we also need to stress its limitations. These limitations are related both to decisions that we have taken during our work, and also due to the time limitations associated with this thesis.

One of the limitations is the lack of adequate comparative term for the final results. Since Enterprise Information Architecture patterns are practically nonexistent in multiple areas, it's difficult to chose the right metrics and compare our method and our final catalog with already existing methods to create patterns for Enterprise Architectures.
Another limitation is that this method is not automated. This is due to the fact that the Enterprise Information Architectures are usually represented in relational databases models or in higher abstraction level models, such as UML or EA model, which is difficult to make a automated comparison between them since they are in different meta-models. Initially, this thesis’ solution proposal was to automate the pattern generation process for EIA represented in ArchiMate Model, with the help of the Archi Modelling tool who has the functionality of exporting architectures to XML. However, the Portuguese PA, more specifically, AMA, the agency we worked with, didn’t have any of their EIA represented in ArchiMate model, so it wasn’t possible to do it, and our solution proposal was adapted.

6.5 Future Work

Firstly, we think that our proposal has potential, in order to be considered a good methodology to develop Enterprise Information Architecture patterns. However, we believe that this methodology can still be improved.

There is still plenty of research that can be done having this thesis as a basis.

Based on the outcomes of this work, we may point some following opportunities for related future work, such as applying it to more study cases. We believe that the application of this methodology to more cases where is possible to compare the results with defined Enterprise Information Architectures is quite relevant. This could improve the feedback received from such a comparative analysis, and would definitely improve the quality of the methodology as well as extend it.

This, it is important to conduct further DSRM work, integrating more aspects to compare different Enterprise Information Architectures.

It is also possible to extend the scope and/or depth of the thesis work, e.g. by developing solution’s proposal that addresses the inconsistencies detected and by demonstrating and evaluating the method in more government owned companies.

Another relevant improvement is to automate the process of our solution proposal.

In short, as enterprises become increasingly information based, making improvements in their information activities is a top priority to assure their continuing competitiveness and a key to achieving these improvements is developing an Enterprise Information Architecture. Therefore, this research work can help creating value by capturing patterns that can be useful in future projects with similar context, regarding the Portuguese Government and PA.
References


Appendix A

Informational Architecture of AMA
Appendix B

Liferay DB used on Portal do Cidadão
Appendix C

AMA’s EIA Patterns Catalog
Enterprise Information Architecture Patterns Catalog for AMA

Release 1.0

October 2016
1. Introduction

The objective of this document is to document patterns as a recurring solution to common problems in a given context of AMA. These patterns can be used for future projects of AMA which similar context in order to save time and avoid mistakes made in previous projects.

The EIA Pattern Catalog utilizes a consistent terminology and information organization to simplify the selection, adaption and integration of patterns.

Patterns of all types are described uniformly using the notation described in the following table.

<table>
<thead>
<tr>
<th>Overview Section</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>A unique alphanumerical identifier.</td>
</tr>
<tr>
<td>Name</td>
<td>A short and expressive name for the EIA pattern.</td>
</tr>
<tr>
<td>Alias</td>
<td>Names this EIA pattern is also known as (optional).</td>
</tr>
<tr>
<td>Summary</td>
<td>A short summary of the EIA pattern (optional).</td>
</tr>
<tr>
<td>Version</td>
<td>Version number of the EAM pattern.</td>
</tr>
<tr>
<td>Context Section</td>
<td>The context helps to determine where to use the pattern, and provides evidence that it is of general application.</td>
</tr>
<tr>
<td>Problem Section</td>
<td>Set of forces repeatedly arising in the context.</td>
</tr>
<tr>
<td>Solution Section</td>
<td>Detailed description of the EIA pattern.</td>
</tr>
<tr>
<td>Consequences Section</td>
<td>Consequences resulting from the usage of the EIA pattern.</td>
</tr>
<tr>
<td></td>
<td>(optional)</td>
</tr>
</tbody>
</table>
2. EIA Patterns for Government

2.1 Balcão do Empreendedor – AMA Informational Architecture

2.1.1 Entity-Generalization Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

Context Section

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

Problem Section

The IE named Entity is an abstract concept and needs to have specializations that specifies the two different types of Entities.

Solution Section

```
Entidade
- Nome char
- Estado char
- Dítrito char

Grupo
- Nome char
- Código char

Pessoa
- Nome char
- NomeProprio char
- Estado char
- Código char
```

Consequences Section

The Informational Entity (IE) Entidade is the superclass of both Grupo and Pessoa.

2.1.2 Entity-Contact Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

Context Section

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

Problem Section

The IEs Grupo and Pessoa need to have contact addresses in order to AMA contact them for exchange of information or for notifications.

Solution Section

```
Entidade
- Nome char
- Estado char
- Dítrito char

Grupo
- Nome char
- Código char

Pessoa
- Nome char
- NomeProprio char
- Estado char
- Código char

Contato
- TipoContato char
- ContatoPreferencial char
```

Consequences Section

Entidade, Grupo and Pessoa will have an association with the IE Contato, with attributes TipoContato and ContatoPreferencial.
### 2.1.3 Email-Contact Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Id</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Alias</strong></td>
</tr>
<tr>
<td><strong>Summary</strong></td>
</tr>
<tr>
<td><strong>Version</strong></td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Balcão do Empreendedor and AMA's Informational Architecture.

**Problem Section**

There isn’t just one contact, but many different contacts and they are all different types of contacts.

**Solution Section**

![Email Contact Pattern Diagram]

**Consequences Section**

*Contato Correio Electronico* is a specialization of *Contato*, where its attribute is the e-mail address.

### 2.1.4 Fax-Contact Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Id</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Alias</strong></td>
</tr>
<tr>
<td><strong>Summary</strong></td>
</tr>
<tr>
<td><strong>Version</strong></td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

**Problem Section**

There isn’t just one contact, but many different contacts and they are all different types of contacts.

**Solution Section**

![Fax Contact Pattern Diagram]

**Consequences Section**

*Contato Fax* is a specialization of *Contato*, where its attribute is a Fax number.
2.1.5 Telephone-Contact Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

Context Section

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

Problem Section

There isn’t just one contact, but many and they are all different types of contacts.

Solution Section

![Diagram of Telephone-Contact Pattern]

Consequences Section

Telephone is a specialization of Contato, with the number of the telephone as an attribute.

2.1.6 Entity-Address Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

Context Section

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

Problem Section

The IE Entidade needs to have a physical address.

Solution Section

![Diagram of Entity-Address Pattern]

Consequences Section

Entidade has an association with Concelho and Distrito as part of its address.
### 2.1.7 Entity-Operations Pattern

**Overview Section**

<table>
<thead>
<tr>
<th>Id</th>
<th>P-7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Entity-Operations Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Entidade has an association with Operacao.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

**Problem Section**

The IE Entidade needs to have an association to an action that can be made.

**Solution Section**

```
class Pattern 7

Operacao
  - OperacaoEstado: int
  - OperacaoTipo: int

Entidade
  - Nome: char
  - Estado: char
  - Distrito: char
```

**Consequences Section**

Entidade has an association with Operacao, with its type and state as attributes.

### 2.1.8 Entity-Activities Pattern

**Overview Section**

<table>
<thead>
<tr>
<th>Id</th>
<th>P-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Entity-Activities Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Entidade has an association with Atividade.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

**Problem Section**

The IE Entidade needs to have categories for its activities.

**Solution Section**

```
class Pattern 8

Entidade
  - Nome: char
  - Estado: char
  - Distrito: char

Atividade
  - AtividadeEstado: char

CategoriaAtividade
  - CategoriaAtividadeEstado: char
  - CategoriaAtividadeNome: char

SubCategoriaAtividade
  - SubCategoriaAtividadeNome: char
  - SubCategoriaAtividadeEstado: char
```

**Consequences Section**

Entidade has an association with Atividade. Atividade has an association with Categoria Atividade, which has an association with SubCategoriaAtividade.
### 2.1.9 Activity-Forms Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

**Problem Section**

The IE Atividade needs to have a form to fill out when requesting services.

**Solution Section**

- Atividade has an association with Formulario., with the attributes FormularioNome and FormularioTipo.

### 2.1.10 Entity-FormRequests Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

**Problem Section**

The IE Formulario needs to have form requests associated with Entity.

**Solution Section**

- Entity has an association with Formulario Pedido, which has an association with Formulario Tipo Estado Pedido.
### 2.1.11 Entity-Licenses Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

**Problem Section**

*Entidade* needs to have licenses for its activities.

**Solution Section**

*Entidade* has an association with *Licenca Atividade*, with its level, order and status as attributes.

### 2.1.12 Entity-RequestTypes Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

**Problem Section**

*Entidade* has requests, but there are different types of requests.

**Solution Section**

*Entidade* has an association with *Pedido*. *Pedido* has an association with *Tipo Pedido*. 
### 2.1.13 GroupLicenses Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

**Problem Section**

The specialization Grupo of the IE Entidade has different types of services requests of IE Pessoa, such as licensees for groups/companies.

**Solution Section**

**Consequences Section**

IE Grupo is associated to Pedido Licenca Entidade Grupo Contato, which is associated to Pedido.

![Diagram](image1.png)

### 2.1.14 LicenseRequests Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

**Problem Section**

IE Entidade has generic requests.

**Solution Section**

**Consequences Section**

Entidade has an association with Pedido, which has an association with Licenca.
### 2.1.15 LicenseTypes Pattern

#### Overview Section

<table>
<thead>
<tr>
<th>Id</th>
<th>P-15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>LicenseTypes Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Licença has an association with Tipologia Licença.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

#### Context Section

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

#### Problem Section

IE Licença has different types.

#### Solution Section

![Diagram of LicenseTypes Pattern](image)

#### Consequences Section

Licença has an association with Tipologia Licença.

### 2.1.16 RequestTypes Pattern

#### Overview Section

<table>
<thead>
<tr>
<th>Id</th>
<th>P-16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>RequestTypes Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Pedidos has an association with Tipos Pedido.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

#### Context Section

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

#### Problem Section

There are different types of requests.

#### Solution Section

![Diagram of RequestTypes Pattern](image)

#### Consequences Section

IE Pedido has an association with Tipo Pedido.
2.1.17 Actions Pattern

Overview Section

<table>
<thead>
<tr>
<th>Id</th>
<th>P-17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Actions Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Entidade has an association with Registo Accoes.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Context Section

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

Problem Section

Each Entidade has an action recorder.

Solution Section

![Diagram of the Actions Pattern]

Consequences Section

The IE Entidade has an association with Registo Ação.

2.1.18 RequestStates Pattern

Overview Section

<table>
<thead>
<tr>
<th>Id</th>
<th>P-18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>RequestsStates Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Pedido has an association with Pedido Estado.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Context Section

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

Problem Section

Each IE Pedido needs to have a state, as well as its states types.

Solution Section

![Diagram of the RequestStates Pattern]

Consequences Section

IE Pedido has an association with Pedido Estado, which has an association with Tipo Estado Pedido.
### 2.1.19 Tasks Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Id</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Alias</strong></td>
</tr>
<tr>
<td><strong>Summary</strong></td>
</tr>
<tr>
<td><strong>Version</strong></td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

**Problem Section**

Each task has different states.

**Solution Section**

![Diagram of Task and Task State Associations](image)

**Consequences Section**

The IE Tarefa has an association with Tarefa Estado, with attributes including the date, the state and observations.

### 2.1.20 TaskTypes Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Id</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Alias</strong></td>
</tr>
<tr>
<td><strong>Summary</strong></td>
</tr>
<tr>
<td><strong>Version</strong></td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Balcão do Empreendedor and AMA’s Informational Architecture.

**Problem Section**

Each state of each task has different types of states.

**Solution Section**

![Diagram of Task and Task Type Associations](image)

**Consequences Section**

IE Tarefa has an association with Tipo Estado Tarefa with its name as an attribute.
2.2 Portal do Cidadão – AMA Informational Architecture

2.2.1 Organization Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

Context Section

This pattern is generated in the context of Portal do Cidadão and AMA’s Informational Architecture.

Problem Section

Company is associated with an organization.

Solution Section

Consequences Section

IE Company has an association with Organization.

2.2.2 User-Email Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

Context Section

This pattern is generated in the context of Portal do Cidadão and AMA’s Informational Architecture.

Problem Section

A User has an e-mail address as a means of contact.

Solution Section

Consequences Section

Informational Entity User has an association with Contact, which is a generalization of EmailAddress.
### 2.2.3 User-Addresses Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Id</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Alias</strong></td>
</tr>
<tr>
<td><strong>Summary</strong></td>
</tr>
<tr>
<td><strong>Version</strong></td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Portal do Cidadão and AMA's Informational Architecture.

**Problem Section**

User needs to have an address associated with a region.

**Solution Section**

![Diagram of User-Addresses Pattern](image)

**Consequences Section**

A User has an association with address that includes a country and a region.

### 2.2.4 Addresses Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Id</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Alias</strong></td>
</tr>
<tr>
<td><strong>Summary</strong></td>
</tr>
<tr>
<td><strong>Version</strong></td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Portal do Cidadão and AMA's Informational Architecture.

**Problem Section**

Address needs to be complete.

**Solution Section**

![Diagram of Addresses Pattern](image)

**Consequences Section**

User has association with StreetType and District, which is related with County and Parish.
2.2.5 User-JobTitle Pattern

**Overview Section**

<table>
<thead>
<tr>
<th>Id</th>
<th>P-25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>User-JobTitle Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>User has an association with a Job.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Portal do Cidadão and AMA’s Informational Architecture.

**Problem Section**

A user needs to have a job associated with him.

**Solution Section**

**Consequences Section**

IE User, which is a specialization of Entity, has an association with JobTitle.

---

2.2.6 User-Message Pattern

**Overview Section**

<table>
<thead>
<tr>
<th>Id</th>
<th>P-26</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>User-Message Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>User has a message.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Portal do Cidadão and AMA’s Informational Architecture.

**Problem Section**

User needs to have a form of communication, such as a message, with another user specialized in a certain job title which provides services.

**Solution Section**

**Consequences Section**

User has an association with Message.
### 2.2.7 Organization-Message Pattern

**Overview Section**

<table>
<thead>
<tr>
<th>Id</th>
<th>P-27</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Organization-Message Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Organization sends messages.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Portal do Cidadão and AMA’s Informational Architecture.

**Problem Section**

Organization needs to send a message to a specialized professional.

**Solution Section**

![Diagram of Organization-Message Pattern]

**Consequences Section**

*Organization* has an association with *Jobtitle* and with *Message*. *JobTitle* also has an association with *Message*.

### 2.2.8 User-Phone Pattern

**Overview Section**

<table>
<thead>
<tr>
<th>Id</th>
<th>P-28</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>User-Phone Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>User has phone contact.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Portal do Cidadão and AMA’s Informational Architecture.

**Problem Section**

A User has different types of contacts.

**Solution Section**

![Diagram of User-Phone Pattern]

**Consequences Section**

*User* has an association with *Contact* which is a generalization of *Phone*. 
### 2.3 Portal do Cidadão – Balcão do Empreendedor

#### 2.3.1 Users Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Portal do Cidadão and Balcão do Empreendedor.

**Problem Section**

A User has many characteristics that describes him.

**Solution Section**

![Class Diagram](image)

**Consequences Section**

User has attributes that keep the entities' characteristics, such as name, status, contact, and so on.

---

#### 2.3.2 Users-Contacts Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Portal do Cidadão and Balcão do Empreendedor.

**Problem Section**

Users need to have contacts associated with them.

**Solution Section**

![Class Diagram](image)

**Consequences Section**

IE User is associated with Contact.
### 2.3.3 UserGroups-contact Pattern

**Overview Section**

<table>
<thead>
<tr>
<th>Id</th>
<th>P-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>UserGroups-contact Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Groups and Users have E-mail Address.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Portal do Cidadão and Balcão do Empreendedor.

**Problem Section**

Groups and Users need to have e-mail address.

**Solution Section**

![UML Diagram](image-url).

**Consequences Section**

*Group* and *User* have an association with the IE EmailAddress.

### 2.3.4 UserGroups-Message Pattern

**Overview Section**

<table>
<thead>
<tr>
<th>Id</th>
<th>P-32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>UserGroups-Message Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Groups and Users send messages.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Portal do Cidadão and Balcão do Empreendedor.

**Problem Section**

Groups and Users need to send messages between them.

**Solution Section**

![UML Diagram](image-url).

**Consequences Section**

*Group* and *User* have association with *Message*.
### 2.3.5 Users-District Pattern

**Overview Section**

<table>
<thead>
<tr>
<th>Id</th>
<th>P-33</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Users-District Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Users have a district and county.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Portal do Cidadão and Balcão do Empreendedor.

**Problem Section**

Users need to have an address.

**Solution Section**

![Diagram](image1)

**Consequences Section**

*User is associated with County and District.*

### 2.3.6 Statistics Pattern

**Overview Section**

<table>
<thead>
<tr>
<th>Id</th>
<th>P-34</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Statistics Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td></td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Context Section**

This pattern is generated in the context of Portal do Cidadão and Balcão do Empreendedor.

**Problem Section**

It's necessary to collect and analyze business data.

**Solution Section**

![Diagram](image2)

**Consequences Section**

There is an IE named Statistic with `companyId`, `statisticId`, `userNameId` and `data` as its attributes.
### 2.3.7 Organizations-Address Pattern

#### Overview Section

<table>
<thead>
<tr>
<th>Id</th>
<th>35</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Organizations-Address Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Organizations have addresses.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

#### Context Section

This pattern is generated in the context of Portal do Cidadão and Balcão do Empreendedor.

#### Problem Section

Organizations need to have addresses.

#### Solution Section

The diagram illustrates the relationship between Organizations, Districts, and Counties.

#### Consequences Section

IE Organization has associations with District and County.

### 2.3.8 Organizations-Services Pattern

#### Overview Section

<table>
<thead>
<tr>
<th>Id</th>
<th>36</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Organizations-Services Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Organizations has an association with Services.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

#### Context Section

This pattern is generated in the context of Portal do Cidadão and Balcão do Empreendedor.

#### Problem Section

Organizations request services.

#### Solution Section

The diagram illustrates the association between Organizations and Services, with attributes description, serviced, createdDate, companyId, and status.

#### Consequences Section

The IE Organization has an association with IE Service, with the following attributes: description, serviced, createdDate, companyId, and status.
### 2.3.9 User-Services Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

#### Context Section

This pattern is generated in the context of Portal do Cidadão and Balcão do Empreendedor.

#### Problem Section

A user can request services.

#### Solution Section

![Diagram](class Pattern 37.png)

User has an association with Service, with some attributes such as *id*, *description*, *created data*, and so on.

#### Consequences Section

User has an association with Service, with some attributes such as *id*, *description*, *created data*, and so on.

### 2.3.10 Test Anti-Pattern

<table>
<thead>
<tr>
<th>Overview Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Alias</td>
</tr>
<tr>
<td>Summary</td>
</tr>
<tr>
<td>Version</td>
</tr>
</tbody>
</table>

#### Context Section

This Anti-Pattern is generated in the context of Portal do Cidadão and Balcão do Empreendedor.

#### Problem Section

The functionality of the system needs to be tested.

#### Solution Section

![Diagram](class AntiPattern1.png)

AMA’s EIA will have an informational entity that shouldn’t be in the informational architecture, since it’s a bad design solution to have an informational entity to be used only for testing purposes.

#### Consequences Section

AMA’s EIA will have an informational entity that shouldn’t be in the informational architecture, since it’s a bad design solution to have an informational entity to be used only for testing purposes.
2.3.11 Address Anti-Pattern

### Overview Section

<table>
<thead>
<tr>
<th>Id</th>
<th>AP-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Address Anti-Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Anti-pattern regarding addresses and regions.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

### Context Section

This Anti-Pattern is generated in the context of Portal do Cidadão and Balcão do Empreendedor.

### Problem Section

An address has a region to represent the location.

### Solution Section

![Class diagram for Address Anti-Pattern]

### Consequences Section

If you use this [bad] solution, the EIA will have an informational entity that is out of date and needs to be updated with useful entities that are being used.

2.3.12 FormRequests Anti-Pattern

### Overview Section

<table>
<thead>
<tr>
<th>Id</th>
<th>AP-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>FormRequests Anti-Pattern</td>
</tr>
<tr>
<td>Alias</td>
<td></td>
</tr>
<tr>
<td>Summary</td>
<td>Formularios_Pedidos are used in the informational architecture.</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
</tbody>
</table>

### Context Section

This Anti-Pattern is generated in the context of Portal do Cidadão and Balcão do Empreendedor.

### Problem Section

Entidade needs to have a form to make requests.

### Solution Section

![Class diagram for FormRequests Anti-Pattern]

### Consequences Section

AMA’s EIA will an informational entity, namely, Formularios_Pedidos that is out of date and needs to be updated with useful entities.