Enterprise Information Architecture Patterns for Government

Rute Lemos
INESC-ID
Instituto Superior Técnico
rute.lemos@ist.utl.pt

***Abstract*—Currently, there isn't a specific method to find patterns for Enterprise Information Architecture (EIA) 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 the documentation of it. For demonstration purposes, an example based on a practical case is shown. Evaluation was made using interviews and surveys, and also by comparison with other Enterprise Information Architecture approaches.

Keywords - Enterprise Information Architecture, Patterns, Informational Entities, Pattern Documentation.

I. Introduction

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 is a structured set of multidimensional interrelated elements that support all information processes [2].

There are many patterns and anti-patterns yet to be discovered, because standard existing patterns, such as client-server [3], don't cover all existing architectures in EIA [4]. Even though each system is unique, there are many systems with areas that have similar architectures, and as such, there's a probability that patterns can be discovered.

Also in the Portuguese Public Administration (PA) context there is a lack of specific methodologies for improving and updating the Enterprise Information Architecture, regarding future projects with similar context. As so, this research 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, AMA (*Agência para a Modernização Administrativa*), the government agency responsible for the execution of the project of the information architecture for the Portuguese Public Administration.

The structure of this research follows the principles and guidelines that the Design Science Research Methodology (DSRM) indicates [5]. In section II, the problem is presented. In section III, the related work is described and section IV explains the solution proposal. Section V has a demonstration and section VI presents the evaluation. Finally, the Conclusion is in section VII.

II. Problem

The importance of information in the Portuguese PA, the lack of specific methodologies for improving and developing the Enterprise Information Architectures, the nonexistence of a method to create an EIA patterns catalog in the Portuguese PA, lead us to need for the Portuguese PA 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?
- 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?

Based on the problem questions described above, this paper contributes to the development of both Enterprise Information Architecture and pattern topics, 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.

III. Related Work

A. Enterprise Information Architecture

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

Regarding the composition of an EIA, it's possible to distinguish three fundamental concepts[6]:

- **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.

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].

B. Pattern

Patterns evolved from several initiatives. Kent Beck and Ward Cunning-ham, two of the pioneers of Smalltalk, came across the ideas of Christopher Alexander, who had developed a theory and collection of patterns in architecture [7]. Alexander developed a range of theories about patterns in architecture and published these in a series of books [8].

Douglass explained in his book [9] 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 [10].

Anti-pattern is also a concept worth mentioning. It describes a commonly occurring solution to a problem that generates decidedly negative consequences [11]. 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.

C. Work developed by the Portuguese Government

This research is developed in collaboration with *Agência para a Modernização Administrativa* (AMA), the government agency responsible for defining ICT guidelines for the Public Administration in Portugal.

AMA has 3 important projects that we will reference through the paper:

- **Citizen Portal** (*Portal do Cidadão*, in Portuguese): Its main goal is to simplify the relationship between citizens and public organisms, being the major channel to the services provided by the Public Administration. Its data is supported in a MySQL DBMS;
- **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 architecture is described in UML.
- **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 [1].

D. Other related work

Brás et al. [12] 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. 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 [12], the schema/model integration has 4 total phases. We will only use the second phase of this method, which is the following:

- 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. Giving two models S1 and S2, having a match means that for each concept in S1, we try to find a concept in S2 that it will be semantically similar.

A set of rules were defined to verify the existence of similarity between IE of different models, where these rules are grouped by order of importance. Formally, this similarity is quantified as:

[1] https://m6.ama.pt/docs/ArquiteturaInformacional.pdf

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

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.

IV. Solution Proposal

The main goal of this research is:

- 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, this paper 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 research 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;
- Reduce the costs for creating a Patterns Catalog, by reducing the number of human resources necessary to develop it.

Figure 1 shows the various steps of the solution proposal.

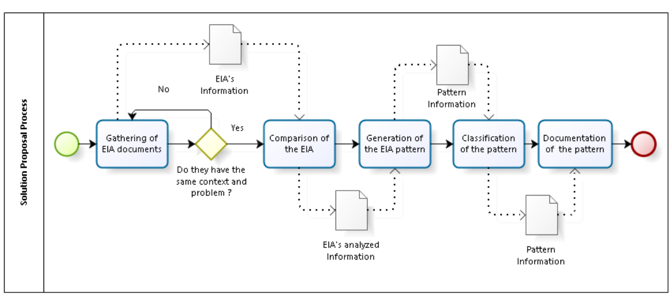


Fig. 1. Process model of the solution proposal.

A. 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. 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.

B. 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.

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.

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 a algorithm to calculate it, such as Jaro–Winkler distance.

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, them 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.

A set of similarity rules based on Brás' work were defined. 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 son. This way, we can guarantee that the most important rules have a considerable bigger weight than less important rules.

The process to compare the similarity between two IE is described as it follows:

1) Given two models with similar context, we intend to calculate the similarities between any pair of IE;
2) Choose a pair of IE to be compared;
3) Iterate, in order, for all the similarity rule groups and apply, the maximum, one rule of each group;
4) Sum weights of the rules applied;
5) When sum of weights is higher than the cut-off (0.70) or equal, the IEs are considered correspondent and the process ends;
6) If at the end of the process, the value obtained is lower than the cut-off (0.70), IEs are not considered a match.

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 meta-models, 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 meta-model used, to verify the similarity between different IE, 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 meta-model of the two architectures, but the process is the same.

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 EIAs 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 I**.

Rule	Description	Weight
1	If two IE have the same Uniform Resource Identifier, they are equal.	1.0
2	If two IE have the same instances, they are equal.	1.0
3	If the name of two IE is the same or similar, it is likely that the two IE are equal or similar.	0.5
4	If two IE have the same primary key, it's likely they are equal.	0.5
5	If the attributes of two IE are equal, it is likely that the two IE are also equal.	0.5
6	If the description of two IE is the same or similar, it is likely that the two IE are equal or similar.	0.3
7	If the hierarchical path to the IE is equal, the compared IE are similar.	0.25
8	If the super entities are the same, the compared IE are similar	0.25
9	If the sub entities are the same, the compared IE are similar.	0.25
10	If two IE have equal "sisters IE", the compared IE are similar.	0.25
11	If two instances have the same "mother IE", they are similar.	0.25
12	IE that have equal instance quantity, they are similar.	0.25
13	If two instances are connected to another instance through the same property, they are similar among themselves.	0.25
14	If two IE have similar functions, they may be equal.	0.2
15	If the type of the attributes being compared are equal, the attributes may be equal.	0.2

TABLE I
SIMILARITY RULES.

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, the similarity rules used will be most of the rules in **Table I**, except rules 2, 4, 11, 12 and 13.

3) Representation of EIA both in Relational Databases and UML 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.

Figure 2 summarizes the reverse engineering transformations from section 2.8 to be used in this phase in case all the architectures aren't in the same meta-model, more especifically in lower levels, like relational databases.

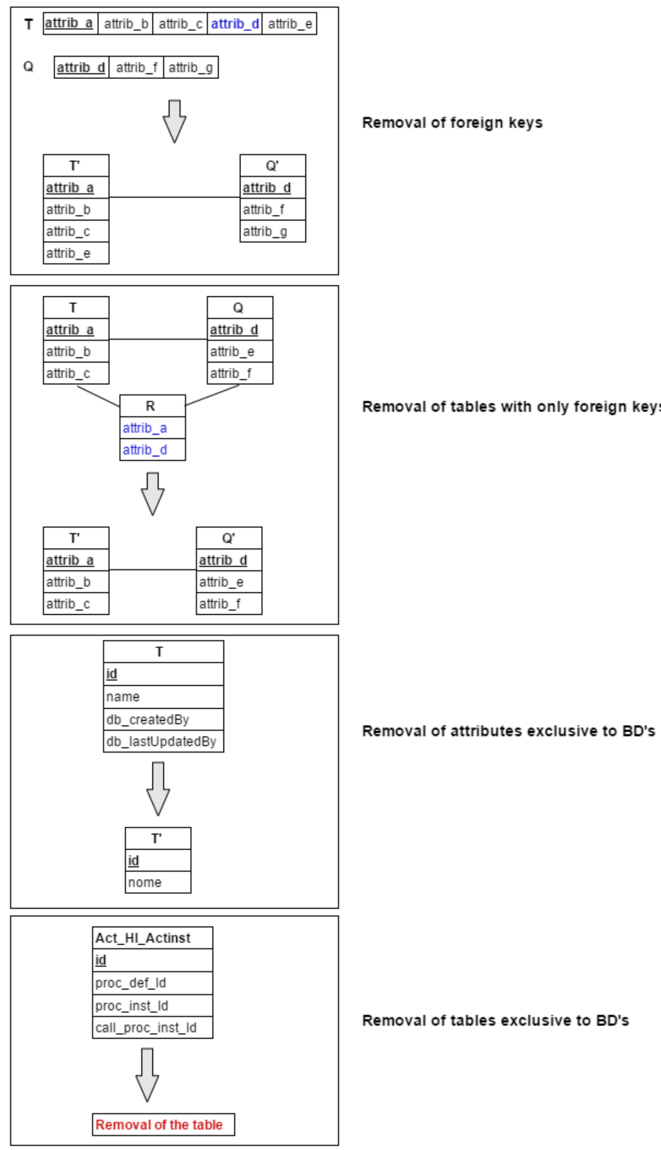


Fig. 2. Reverse Engineering Steps.

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 4.2.2.

Figure 3 resumes the application of the Similarity Rules according to the meta-model.

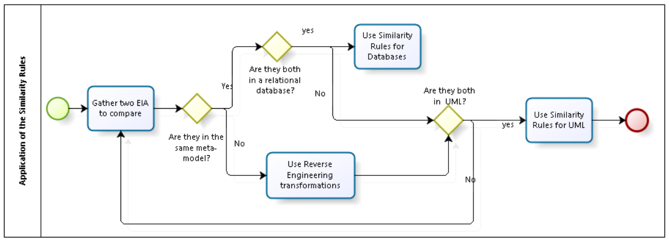


Fig. 3. Application of the Similarity Rules according to meta-model.

C. 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.

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.

Figure 4 shows this rule, 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. .

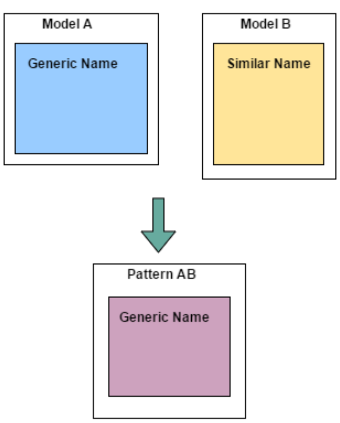


Fig. 4. Pattern Generation Rule 1.

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 5**.

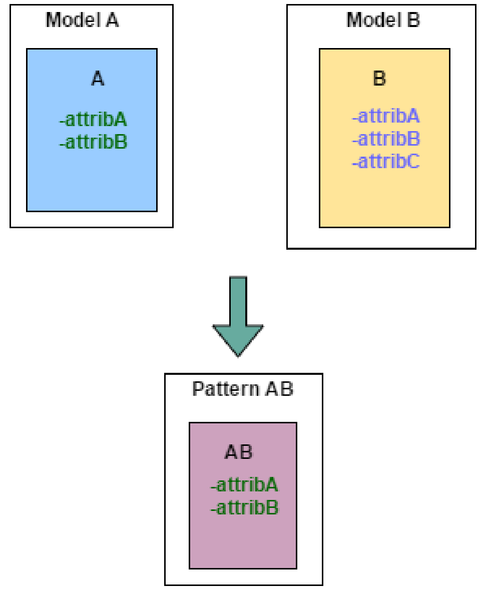


Fig. 5. Pattern Generation Rule 2.

3) Pattern Generation Rule 3: The generated pattern will have the same relationships that are common in both IEs, including the attributes that are also a match, as shown in **Figure 6**.

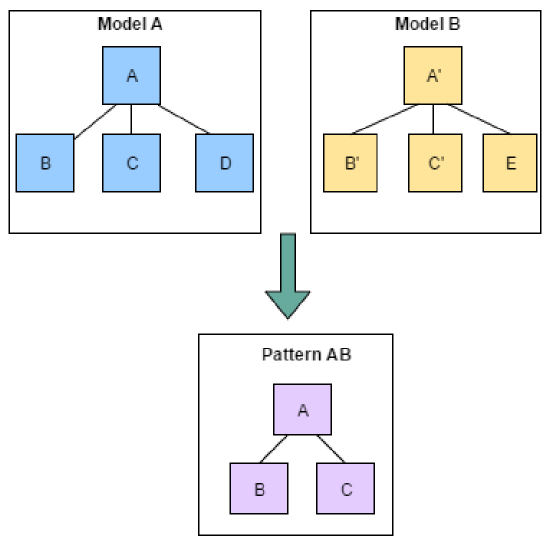


Fig. 6. Pattern Generation Rule 3.

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. **Figure 7** resumes this rule.

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.

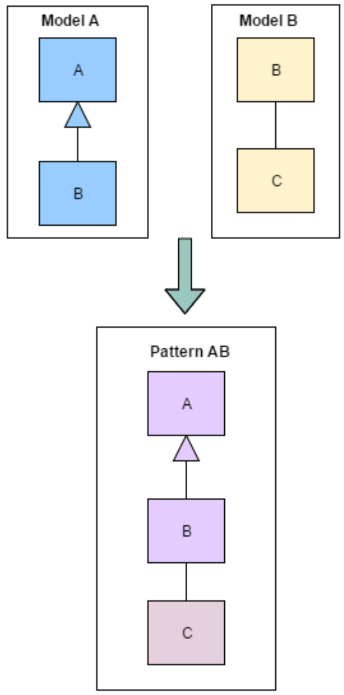


Fig. 7. Pattern Generation Rule 4.

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.

Figure 8 demonstrates this rule.

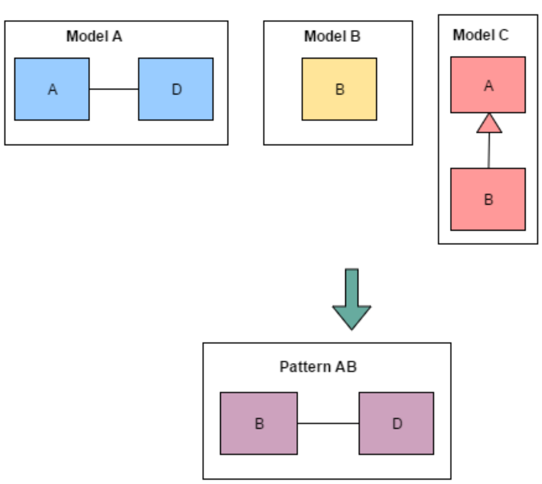


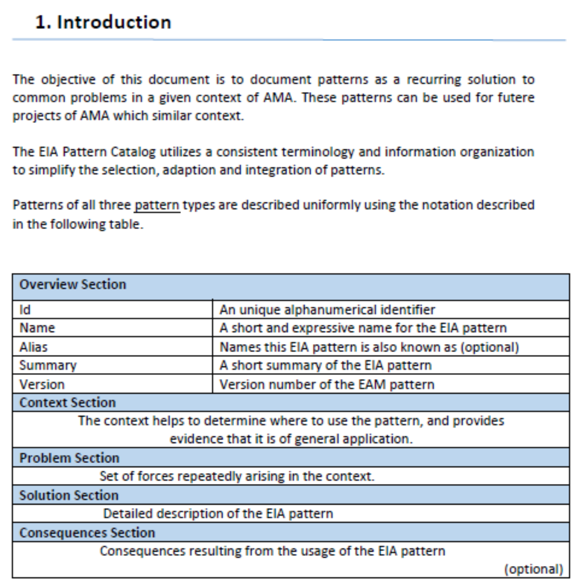
Fig. 8. Pattern Generation Rule 5.

D. 4^{th} Phase - Classification of the Pattern

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.

E. 5^{th} Phase - Documentation of the pattern in EIA Catalog

The Information Architecture Patterns and Anti-Patterns catalog structure is shown in **Figure 9**.



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 futere projects of AMA which similar context.

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

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

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

Fig. 9. Patterns Catalog.

V. DEMONSTRATION

In this demonstration, a simple example is used to show the several steps of the solution proposal.

For the first phase described in the solution's proposal section of this work, we needed to select and analyze Enterprise Information Architectures with similar contexts. In this case, let's suppose that **Figure 10** is part of a Model A, representing an EIA in relational database tables, and **Figure 11** is part of a Model B, representing another EIA where both of them have similar problem and context, verifying the conditions described in section 4 (first phase of the solution proposal). It is also verified that they are in the same language, in this case, English.

From the tables given in **Figure 10**, we can conclude that they are represented in a different model than **Figure 11**, 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

Table 1 - Entity

Field Number	Field	PK	FK
1	entityID	•	
2	label		
3	designation		
4	status		
5	addressID		•

Table 2 - Organization

Field Number	Field	PK	FK
1	organizationID	•	
2	name		
3	contactID		•
4	NIPC		
5	CAE		

Table 3 - Address

Field Number	Field	PK	FK
1	addressID	•	
2	district		
3	county		
4	parish		
5	zipCode		
6	streetSuffix		
7	doorNumber		

Table 4 - Contact

Field Number	Field	PK	FK
1	contactID	•	
2	phone		
3	eMail		
4	fax		
5	dbCreatedBy		
6	addedByLDAPImport		

Table 5 - Person

Field Number	Field	PK	FK
1	personID	•	
2	firstName		
3	lastName		
4	age		
5	jobTitle		
6	addedByLDAPImport		

Fig. 10. Database tables of Model A from the academic example.

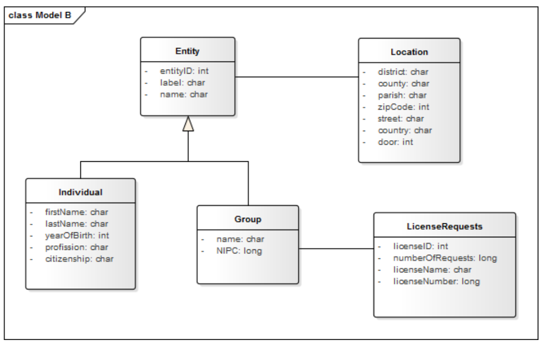


Fig. 11. Model B from academic example.

removed in a higher abstraction level, and so on. The reverse engineering techniques are described in **Figure 2**.

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*;
- Attributes that only have a meaning at the level of databases are removed (for example, addedByLDAPImport 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 12**.

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 11** and **Figure 12**, 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.

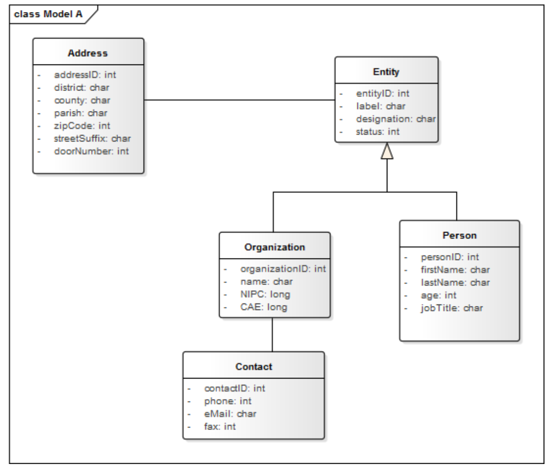


Fig. 12. Model A after reverse engineering techniques, from academic example.

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 **Table I**, 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 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 **Table I**, the first rule that can be applied is *Rule 3 - 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 5 - 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 synonms or have more than 70% string simularity 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.
- **For pair A.Person/B.Individual:** Iterating through the groups of the similarity rules, the first rule that can be applied is *Rule 3 - 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 5 - 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 synonms or have a value superior to 0.70 value of string simularity

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 5, would be *Rule 14 - 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, the first rule that can be applied is *Rule 3 - 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 5 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 correspondent.

Finally, we can now generate the pattern, which is the third phase of our solution proposal. We apply the pattern generation rules, 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 resulting pattern is represented in **Figure 13**.

Next, we proceed to the forth phase. We need to analyze if this pattern has a positive or negative consequences in order to classify the pattern. If the collaborators of AMA consider this pattern to be a solution and a general, reusable solution to a common problem in a given context, then this pattern doesn't have negative consequences. As such, we can consider that it is a 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 4 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 13**, which are:
  - Object – Entity: can be a person or an organization.
  - 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.

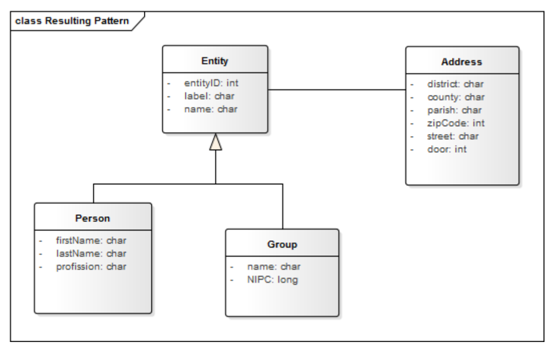


Fig. 13. The resulting pattern from academic example.

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

VI. Evaluation

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 14** 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.

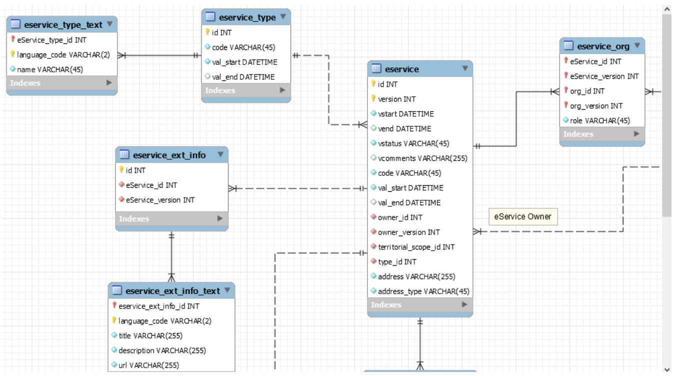


Fig. 14. Part of AMA's Service Catalog.

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 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 this paper's methodology, which means it needs less human resources. **Table II** resumes this analysis.

Metric Variables	**AMA's Service Catalog**	**This work Patterns Catalog**
Minimal number of people needed to create the catalog	12	1
Optimal Number of people needed to create the catalog	12	3

TABLE II
ANALYSIS OF THE TWO CATALOGS.

Another evaluation was made analyzing each different methodology to create the patterns catalogs considering each factor of Moody & Shanks Framework [13], in which we concluded that:

- **Completeness:** Comparing AMA's Service Catalog and this paper's EIA 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 paper's 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.
- **Simplicity:** Due to the structure of this paper's catalog, the EIAs models are divided into smaller patterns, and documented according to the informational entities types. So this paper is simpler than AMA's Service Catalog, since they have more informational entities and relationships.
- **Flexibility:** Both catalogs are similar regarding the flexibility factor.
- **Integration:** Both catalogs are similar regarding the integration factor.
- **Understandability:** This paper's 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 paper's catalog is easier to understand.
- **Implementability:** Both catalogs are similar regarding the implementability factor.

VII. Conclusion

We believe that the solution proposal described in this work adds value to it's 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 5-phased method proposed in section IV 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 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 V 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. Thus, 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 objective was accomplished.

The identification of correspondences between IE of different information architectures and the identification of IE attributes between different EIAs from AMA were possible by using the second phase of the solution proposal, named Comparison of the EIA, where 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.

We were also able to identify common problems in this specific industry concerning Enterprise Information Architecture and extract the patterns, by using the third phase of our solution proposal, 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, 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.

The goal of implementing solutions to this specific industry based on historical data analyzed and information necessary for optimized architecture design decisions has been possible to achieve 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.

Finally, it's possible to reduce the costs for creating a Patterns Catalog, by reducing the number of human resources necessary to develop it, since 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.

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.

By using the knowledge gathered under this 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.

REFERENCES

[1] Godinez, M., Hechler, E., Koenig, K., Lockwood, S., Oberhofer, M. and Schroeck, M. *The Art of Enterprise Information Architecture: A Systems-Bases Approach for Unlocking Business Insight*. IBM Press, 1993.

[2] Watson, R. W. An Enterprise Information Architecture: A Case Study for Decentralized Organizations. *IEEE Computer Society*, 2000.

[3] Fowler, M. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2002.

[4] Reinolds, Y. PATETA: PATterns for EnTerprise reference Architectures. Master's thesis, Instituto Superior Técnico, Universidade de Lisboa, 2015.

[5] Peffers, K., Tuunanen, T., Rothenberger, M. and Chatterjee, S. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–78, 2008.

[6] Vasconcelos, A. Arquitecturas de Sistemas de Informação no Contexto do Negócio. Master's thesis, Instituto Superior Técnico, Universidade de Lisboa, 2001.

[7] Fowler, M. *Analysis Patterns: Reusable Object Models*. Addison-Wesley Professional, 1 edition, 1996.

[8] Alexander, C., Ishikawa, S. and Silverstein, M. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, USA, 1977.

[9] Douglass, B. *Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems*. Addison-Wesley Professional, 2002.

[10] Ernst, A. A Pattern-based Approach to Enterprise Architecture Management. Master's thesis, Fakultät für Informatik der Technischen Universität München, 2010.

[11] Brown, W., Malveau, R., McCormick, H. and Mowbray, T. *AntiPatterns: refactoring software, architectures, and projects in crisis*. Wiley & Sons, New York, NY, USA, 1998.

[12] Brás, T. and Vasconcelos, A. Information Reference Architecture for the Portuguese Health Sector. *Journal of Enterprise Architecture*, 12(3), 2016.

[13] Moody, D. and Shanks, G. Improving the quality of data models: empirical validation of a quality management framework. *Information Systems*, 28(6):619–650, 2003.