Enterprise Architecture of the Public Portuguese Administration

Eduardo Filipe Sanches Gonçalves Custódio Correia

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Examination Committee

Chairperson: Prof. Mário Jorge Costa Gaspar da Silva
Supervisor: Prof. José Manuel Nunes Salvador Tribolet
Co-Supervisor: Prof. Pedro Manuel Moreira Vaz Antunes de Sousa
Members of the Committee: Prof. André Ferreira Ferrão Couto e Vasconcelos

October 2013

Acknowledgments

At the time of the delivery of this dissertation, I look back and see all the hard battles I had to face to get to where I am. In all these battles, there was always the support and the unconditional love of my parents and brother to whom I dedicate this work.

To my parents who were always there to help and show their wisdom. For their sense of ambition and always providing the best possible environment for my learning.

My brother, who since childhood has always protected me and helped whenever possible.

To Teresa, for helping me overcome all academic stages, even in the hard times of illness, supporting me unconditionally and always providing me an environment without stress accompanied by a smile full of affection.

To my colleagues and friends, especially to Luis, Joana, Marcos and António for the advices given. My Master Jorge Ricardo and my training partners, to Pedro Martins, Andronic Ando and Junior Oliveira for the good environment and long workouts full of laughs.

The whole universe of Link Consulting, who helped me realize this thesis and allowed me to put into practice what has been developed in this work.

To Professor Pedro Sousa, whom without his guidance this thesis had not been possible. For all the support he gave me during the course of this work, always showing availability, kindness and time to answer my questions. A special thank you!

To Professor José Tribolet, even if the contact has been little, always present and ready to share his wisdom, insight and knowledge.

To everyone I own the achievement of this work, Thank you!

Enterprise Architecture of the Public Portuguese Administration

Abstract

Organizations are constantly evolving. Changes in their environment are inevitable and change the way of working in various areas. A major objective of this work is to see how these changes affect or will affect the various projects or departments involved and how information systems are affected. This dissertation will use tools to help in the alignment between IT and business in order to allow a correct update of architectures.

The main objective of this thesis is to propose a solution to model the evolved metamodel of our information systems from the public service, in order to meet the needs of our Public Administration. Given that each domain may have different needs in certain requirements specifications, we have to respond to these various changes with minimal changes possible and following well-established steps.

The evolution of this metamodel has been achieved through the study of multiple scenarios, following the different steps of the proposed methodology, we have provided a set of entities to evolve into a metamodel that is flexible.

Since all organisms are currently using this metamodel, it was necessary to create a set of primitives that provided an automation transition between the old model and the new model.

The end result of this work is a metamodel that can support the necessary extensions of the different areas of Public Administration, such as Health, Defense, Education, etc.. We used the tools EAMS from *Link Consulting* as a tool for representation and governance of the developed architectures and Archi from the *Institute of Educational Cybernetics* for the design of the architectures.

Keywords: Architecture, Public Administration, Primitives, Governance, Metamodel, Cost Savings, Efficiency.

Enterprise Architecture of the Public Portuguese Administration

Resumo

As organizações estão em constante evolução. As mudanças no seu ambiente são inevitáveis e alteram o modo de trabalho em vários domínios. Um dos principais objectivos deste trabalho é conseguir ver de que forma essas mudanças afectam ou irão afectar os vários projectos ou departamentos envolvidos e de que forma os sistemas de informação são afectados. Nesta dissertação serão utilizados instrumentos que ajudem no alinhamento entre o IT e o Negócio de forma a permitirem uma correcta actualização das arquitecturas.

O objectivo principal desta tese é propor uma solução para modelar a evolução do metamodelo dos sistemas de informação da função pública, de forma a ir ao encontro das necessidades da nossa Administração Pública. Dado que, cada domínio pode apresentar diferentes necessidades em determinadas especificações de requisitos, temos que conseguir dar resposta a essas várias alterações com o mínimo de mudanças possíveis e seguindo etapas bem estabelecidas.

A evolução deste metamodelo foi conseguido através do estudo de múltiplos cenários, seguindo os diferentes passos da metodologia proposta, que permitiram obter um conjunto de entidades de forma a evoluir-se para um metamodelo flexível.

Dado que todos os organismos estão actualmente a utilizar este metamodelo, foi necessário criar-se um conjunto de primitivas para que houvesse uma automação na transição entre o modelo antigo para o modelo novo.

O resultado final deste trabalho é um metamodelo que consegue suportar as extensões necessárias á abordagem de diferentes domínios da Administração Pública, como por exemplo, da Saúde, da Defesa, da Educação, etc. Foram usadas as ferramentas EAMS da *Link Consulting* como instrumento para representação e governação das arquitecturas desenvolvidas e Archi da *Institute of Educational Cybernetics* para o desenho das arquitecturas.

Palavras-Chave: Arquitectura, Administração Pública, Primitivas, Governação, Metamodelo, Redução de Custos, Eficiência.

Index

List of Figures

List of Tables

List of Acronyms

AMA - Agência para a Modernização Administrativa

ADM - Architecture Development Method

BPMN - Business Process Modeling Notation

DSRM - Design Science Research Methodology

EAMS - Enterprise Architecture Management System

TRM - Technical Reference Model

UML - Unified Modeling Language

XML - Extensible Markup Language

XSD - XML Schema Definition

1 Introduction

1.1 Context

In the current times, enterprises prioritize the improvement in their information among their set of objectives [1]. The activities and processes, executed in the context of an organization, depend on the usage and management of information, bringing very high values associated with these practices.

Given these arguments, the information technologies are more and more important and have become such powerful tools to achieve goals that they constitute a large sum of every organization expenses [2].

Specifically, the agreement made by the Portuguese in the memorandum signed with the International Monetary Fund, the European Central Bank and European Commission, in the measure 3.46 propose in point ii): ii. «*Rationalize the use of IT resources within the central administration by implementing shared services and reducing the number of IT entities in individual Ministries or other public entities*. [Q4-2012].*».*

Nowadays, one of the biggest concerns of the organizations is the correct alignment between the IT and the business. The constant pressure to upgrade the information systems and optimize them is real and is increasingly important to deal.

Furthermore, within the public sector, the technologies systems can´t stop so every changes that are made, must be done in real-time with the minimum impact possible. Through the next sections we will address these problems, in the context of the Portuguese Public Administration, as well as provide the first glance towards a methodology to achieve the desired solution.

1.2 Problem Statement

As said before the information technologies have become so powerful that a well-built architecture is needed for the maximum efficiency in any organization. This thesis focuses on the problem of providing a solution to model the evolved metamodel from the Public Administration, which will be used as a tool to improve the efficiency of the different organisms of the Public Administration.

A metamodel is the foundation for that well-built architecture. It describes the background of an organization pointing out what is relevant and what level of detail we should be working.

Currently, the different organisms of the Public Administration are using a low-performance and outdated architecture. They require different models for solving different problems. However it is important that these models are compatible with each other, allowing a better articulation and integration.

It should also ensure that the final metamodel will be aimed to the achievements of each body of the Portuguese Services as well as done in real-time with the least possible intervention of humans to cause the minimal impact on the services.

After clarifying the context of the problem, we take place by defining the set of objectives associated with this work. In terms of objectives, this work will aim at answering to the two main questions Q1 and Q2 and their sub-questions:

> ***Q1:*** *How can we model the evolved metamodel of the Public Administration?*

In order to answer this question, we had first to study which enterprise architecture modeling language we should use. Within the process of answering the problem, we determined that the architectural framework ArchiMate 2.0 was the more accurate for this work, given the good versatility and uniformity of his metamodel.

To provide a uniform representation for diagrams that describe enterprise architectures, the ArchiMate enterprise architecture modeling language offers an integrated architectural approach that describes and visualizes the different architecture domains and their underlying relations and dependencies.

ArchiMate is a lightweight and scalable language in several respects:

- Its architecture framework is simple but comprehensive enough to provide a good structuring mechanism for architecture domains, layers, and aspects.
- The language incorporates the concepts of the “service orientation” paradigm that promotes a new organizing principle in terms of (business, application, and infrastructure) services for organizations, with far-reaching consequences for their enterprise architecture.[2]

Given that we have a starting metamodel from our Public Administration, we had to proceed to a requirement elicitation of problems that the outdated metamodel had. Once we finished the requirement elicitation we undergo maintenance to evolve it.

All the modeling was done using the tool Archi[1] from the *Institute of Educational Cybernetics*, since it is intended to provide a low cost entry (i.e. free) solution to users or even organizations that can´t afford expensive modeling tools.

Nonetheless, one problem arose with the maintenance of the metamodel. This problem was:

> ***Q1.1:*** *How can we maintain track of the versions without using multiples models?*

Nowadays, this is one of the biggest issues that most companies have. They fail to have a tool that can organize the multiples models and therefore, leads to an effort greater than what should be required.

[1] http://archi.cetis.ac.uk/, acessed 03-07-13.

"…the truth is that companies fail to have such maps, claiming that update costs are simply too high given the rate of changes of the organization artifacts…" [3]

To keep this problem at bay, we used the tool *EAMS* from *Link Consulting* that allow the generation of blueprints, thereby clustering several models into one model using a timeline to visualize older modifications.

With both this tools, a solution was possible but it brought another problem:

> ***Q1.2:*** *How can we transit models from the tool Archi to the EAMS?*

To overcome this problem, a parser had to be made that could translate all the information from the Archi models into information that could be read by the database from the *EAMS*. The database used was the *EADB* and the language used was the *EALang*, both developed by *Link Consulting* and its unit directed by *Prof. Pedro Sousa*.

Which it bring us to the second question regarding the automation of this work:

> ***Q2:*** *What primitives are needed to make an evolutionary maintenance on the current metamodel of the Public Administration?*

In order to answer this second question, we had to resort to the primitives from the conceptual database modeling. Primitives are the building blocks which are used for design methodologies. They should be applicable to a variety of situations and environments, backed up by strategies that are applied in order to enhance them [4].

A set of primitives have been created in order to respond to this question. Which bring us to the sub-question:

> ***Q2.1:*** *What strategy is the most efficient to apply during the evolutionary maintenance?*

Between the four strategies: *Top-Down, Bottom-Up, Inside-Out, Mixed;* a obvious question arises: *Do the four strategies always lead to the same final schema?* Of course the answer is negative, since each strategy follows a specific design philosophy. Indeed, even adopting the same strategy, we may happen to model the same requirements in very different ways [4].

Due to the nature of the strategies, we had to choose the one with more flexibility in order to enhance the set of primitives used.

Since we had access to the initial version of the Public Administration metamodel (fig.1) and after the requirement elicitation, we had all the means necessary to perform the evolutionary maintenance. All that was left was to create and apply the set of primitives to this metamodel.

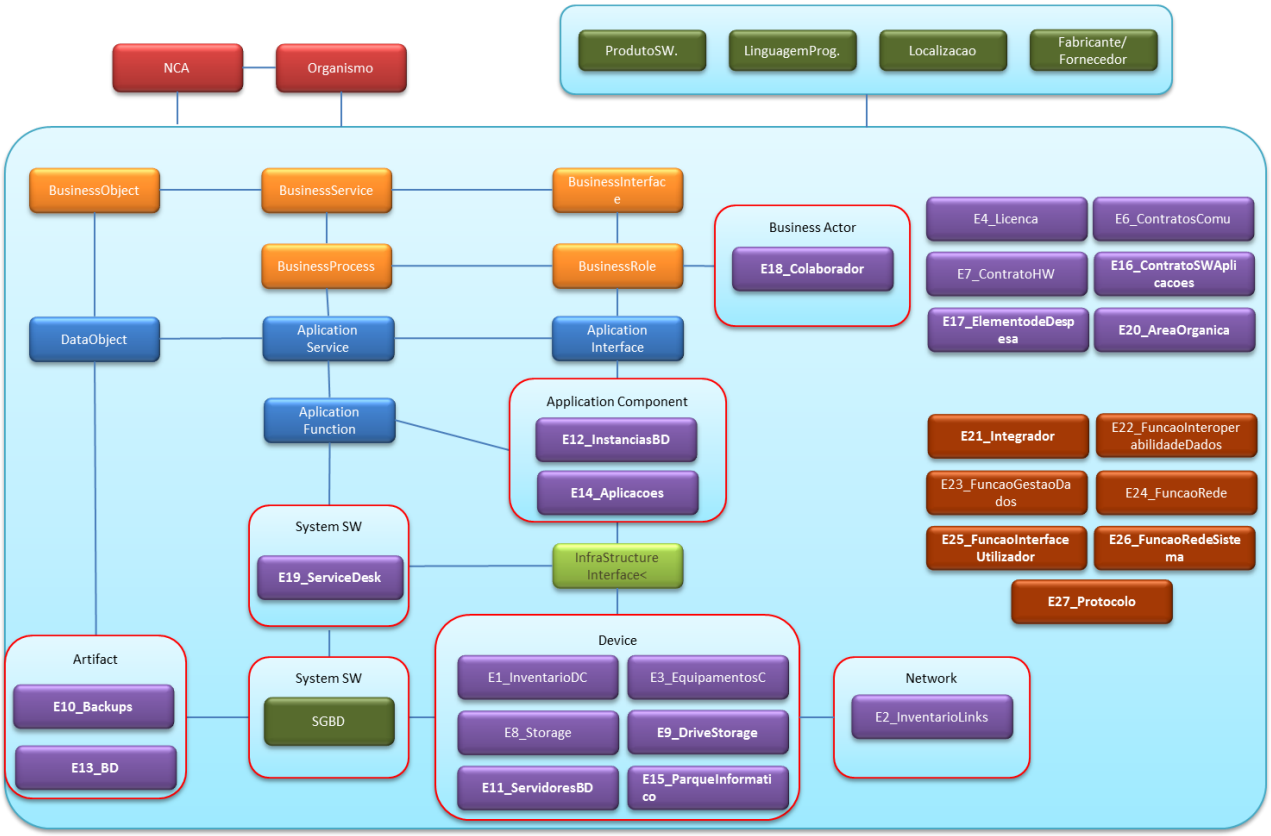

Figure 1: Initial version of the metamodel proposed by AMA.[2]

By answering all these questions we successfully address the problem underlying, and are able to build a solution to model the evolved metamodel of the Public Administration.

1.3 Thesis Objectives

As stated in the previous section, this work will be divided among two major components that complement themselves: an analysis-based component, and a development-based component.

This means that in the first based component, we will study and put the current metamodel in an updated context, analyze it and come up with a solution. This will be part of the maintenance of the current metamodel. In the second based component, we will proceed to the development of the necessary tools to evolve this metamodel so that in the future it can be updated with the least manual effort.

[2] Definição de Meta-modelo e Importação de Ficheiros, realized by Link Consulting – Tecnologias de Informação, S. A.

In short we want to perform an evolutionary maintenance on the outdated metamodel from our Public Administration, build a solution capable of modeling this evolved version, and create a set of primitives that will allow a more swift and automated evolution from one metamodel to another.

In the next section we will define how these objectives will be tackled in the various chapters of this work.

1.4 Structure of the document

The thesis is divided in 7 chapters each comprising various sections.

Chapter 1:

In this chapter starting with this section, we began by introducing the context of our work and its objectives, the problem underlying and the academic and enterprise contributions. The aim is to motivate the reader into the problematic and challenges that surrounded this work, transmitting what we wanted to achieve with the work described along the follow sections.

Chapter 2:

The second chapter is focused on presenting an overview of related work concerning concepts (or architecture descriptions), tools that can generate graphical representations and frameworks that might serve as input towards developing future work.

Chapter 3:

The third chapter is a theoretical overview over what were the methodologies and principles used during the research. The scientific and design science research methodologies are presented in this chapter.

Chapter 4:

The fourth chapter approaches the proposition for a methodology towards the goals we are committed to. The definition of a methodology for addressing this problem is presented in a series of steps along this chapter, and a description about the type of work made in each of them, as well as the various steps taken with the knowledge about the theme.

Chapter 5:

The fifth chapter is where the proposed metamodel is explained. The results from the instantiation of our methodology, our research, and our analysis reinforced the evolutionary maintenance of the metamodel which is presented here. It begins on the current version (As-Is), through the proposed changes, finishing on the proposed evolution (To-Be) and it´s primitives for the automation.

Afterwards we present the parser that was created in order to make it possible to connect the two tools, *Archi* and *EAMS,* used in this work. It will also be presented the framework which the parser was integrated so that it becomes even more versatile in the future.

Chapter 6:

In the sixth chapter is where the proposed methodology begins to be put to practice. It will be presented several practical cases where the solution is being used, reinforcing the impact that this work is having.

Chapter 7:

This final chapter comprises the conclusion of this work.

2 Related Work

2.1 Concepts

This section is aimed at providing a formal definition of the concepts that served as a starting point of this work. We will start by the theoretical concepts; the Public Administration and enterprise architecture, since they are the most connected to this work. Afterwards, we will approach the practical concepts; metamodels and the conceptual designs since this work resolves around the usage of these specific concepts, with the objective of developing a solution to model the evolved metamodel.

2.1.1 Public Administration

Whenever people co-operate to achieve some ends, the activities which they have to perform to achieve the goal in view, it is administration. Public Administration is a particular area of the broader field of administration. It is administration of the governmental affairs. It includes activities like the collection of tax by the Income-tax Officer, the arrest of a criminal by the police, construction of public roads, highways, bridges, canals, etc.

The Public Administration can be described as the management of public programs. Public Administration as we know it today began as the study of Government Administration, and that study began as part of late-nineteenth-century efforts to reform governmental operations.

Even though work in public and nonprofit organizations is guided by commitments to democratic ideals, it is also involved with management, and, for that reason, Public Administrations is often confused with business management. Indeed, such confusion has occasionally been prominent in the field of Public Administration. Certainly, there are some similarities between business and public administration. Managers across all sectors - public, private, and nonprofit - are involved in questions of organizational design, the allocation of scarce resources, and the management of people. [5]

It is distinguished from other forms of administration by the fact that its ultimate purpose is general interest and public good. It is the activity of the State in the exercise of its political powers. In a narrow sense, it is the activity of the executive departments in the conduct of the Government. Thus Public Administration is primarily concerned with the implementation of public policy laid down by representative of political bodies. Its main task is the implementation and enforcement of public policy and the law of the State.

Therefore, a system of public administration is the composite of all the laws, regulations, practices, relationships, codes and customs that prevail at any time in any jurisdiction for the fulfillment or execution of public policy. [6]

2.1.2 Enterprise Architecture

More and more, the notion of architecture is applied with a broader scope than just in the technical and IT domains. The emerging discipline of Enterprise Engineering views enterprises as a while as purposefully designed systems that can be adapted and redesigned in a systematic and controlled way. An enterprise in this context can be defined as any collection or organizations that have a common set of goals and/or a single bottom line.

The emerging instrument of enterprise architecture promises to provide management with insight and overview to harness complexity. Where classical approaches will handle problems one by one, enterprise architecture aims to deal with these issues in a coherent and integral fashion, while at the same time offering a medium to achieve a shared understanding and conceptualization among all stakeholders involved and govern enterprise development based on this conceptualization. As such, enterprise architecture plays a key role in the governance of organizations and their evolution. [7]

Architecture at the level of an entire organization is commonly referred to as enterprise architecture. Enterprise architecture is a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise´s organizational structure, business processes, information systems, and infrastructure. [8]

Enterprise architecture captures the essentials of the business, IT and its evolution. The idea is that essentials are much more stable than the specific solutions that are found for the problems currently at hand. Architecture is therefore helpful in guarding the essentials of the business, while still allowing for maximal flexibility and adaptively. Without good architecture, it is difficult to achieve business success.

The most important characteristic of and enterprise architecture is that it provides a holistic view of the enterprise. Within individual domains local optimization will take place and from a reductionist point of view, the architectures within this domain may be optimal. However, this need not lead to a desired situation for the company as a whole. For example, a highly optimized technical infrastructure that offers great performance at low cost might turn out to be too rigid and inflexible if it needs to support highly agile and rapidly changing business processes. A good enterprise architecture provides the insight needed to balance these requirements and facilitates the translation from corporate strategy to daily operations.

In short, enterprise architecture is a system that is based on the evolution of the organization, offering flexibility and tools to achieve success in an organization based on views that convey the AS-IS and TO-BE of an organizational structure.

2.1.3 Metamodels

Metamodeling is a simple concept, yet difficult to understand. A metamodel approximates the behavior of a more complex model. A common and superficially attractive way to develop a metamodel is to generate large-mode data and use off-the-shelf statistical methods without attempting to understand the model´s internal

workings. [9] The metamodels can be considered as a construction of a set of concepts within a given domain. It is an abstraction of a particular model, which highlights the properties of the model itself. [10]

A metamodel may be compared subjectively as a computer program that complies with the programming language in which it was written.

First, let's take an analogy:

It is quite common that documents contain a glossary. The glossary explains the wording used through the document. Also, it commonly defines concepts wording. But defining concepts in a natural language is always error-prone because of ambiguities, conflicts with readers (private) understanding and a lot of other reasons. A model is also a document that needs some kind of glossary. But, even more than any other document, a model needs a formal glossary that accurately assess underlying concepts meaning and relationships. This formal glossary is a metamodel.

The primary responsibility of the metamodel is to define a language for specifying models. Metamodels are typically more elaborate than the meta-metamodels that describe them, especially when they define dynamic semantics.

Examples of meta-objects in the metamodeling layer are: *Class, Attribute, Operation, and Component*.

A model is an instance of a metamodel. The primary responsibility of the model layer is to define a language that describes an information domain.

Abstract from underlying technology: each technology brings an implied model with it. Building on that model pins the result with this technology. Doing a metamodel will help to better identify a technology based model dependencies and so to assess more generic domain models.

Also, the metamodels concepts can be captured into a development workshop (i.e: UML). Reusing those domain concepts increase developers productivity. Finally, a metamodel helps to partition the problem domain.

2.1.4 Conceptual Designs

Conceptual design starts from the specification of requirements and results in the conceptual schema. The purpose of conceptual design is to describe the information content rather than the storage structures that will be required to manage this information. In fact, conceptual design should be performed even if the final implementation uses conventional files and programming languages. [11]

Primitives and strategies are the building blocks with which we develop design methodologies. A good methodology for conceptual design should ideally be a compromise between two contrasting aspects. First, the methodology should be rigorous; it should suggest a strategy for all the important decisions that are to be made in the design process. Such strategy should be based on a formal approach, and each

decision process should ideally correspond to an algorithm. On the other hand, the methodology should also be flexible; it should be applicable to a variety of situations and environments. Hence the methodology should have sufficient flexibility so that each designer may adapt it to specific organizational constraints and follow his or her own design style.

We should emphasize that conceptual design cannot be very much helped by automatic tools; the designer has full responsibility for the process of understanding and transforming requirements into conceptual schemas. After the first conceptualization, many database systems offer tools for fast prototyping, using fourth-generation languages.

A stronger influence of the final user on design decision has many positive consequences: the quality of the conceptual schema improves, the project most likely converges toward an expected result, and development costs decrease. More importantly, users who are more involved in the decision process are eventually more willing to accept and use the information system.

The final argument in favor of conceptual design, emphasizes the use of conceptual schemas after the end of the design process. The conceptual schema should not be considered as an intermediate design document, to be disregarded after logical and physical design; rather, it should remain as part of the specifications, organized with a variety of documents that also describe in detail the requirement acquisition and design process. Thus, the final and possibly most important advantage of conceptual design shows up, when the conceptual model and its documentation ease the understanding of data schemas and of applications that use them, and thus facilitate their transformation. [11]

In the following sub-sections we will approach, the various methodologies to use in conceptual design, starting by the primitives and following through with the strategies.

2.1.4.1 Primitives

The design of a conceptual schema is the result of a complex analysis of user requirements. As a consequence, the schema is usually produced by an iterative process. During such, a process, we start from some draft version of the schema and perform a set of schema transformations that eventually produce the final version. The situation is similar to what happens in software design, where the final program is usually produced by performing a set of transformations that enrich, in a series of steps, a first-draft version of the program.

Primitives are classified in two groups, top-down and bottom-up. Top-down primitives correspond to pure refinements; that is, refinements that apply to a single concept (the starting schema) and produce a more detailed description of that concept (the resulting schema). By contrast, bottom-up primitives introduce new concepts and properties that do not appear in previous versions of the schema. [11]

2.1.4.1.1 *Top-Down* Primitives

Top-down primitives are characterized by the following properties:

- They have simple structure: the starting schema is a single concept, and the resulting schema consists of a small set of concepts.
- All names are refined into new names, describing the original concept at a lower abstraction level.
- Logical connections should be inherited by a single concept of the resulting schema.

When applying refinement primitives, we have implicitly to respect certain constraints. Some samples from top-down primitives are demonstrated in the figure below:

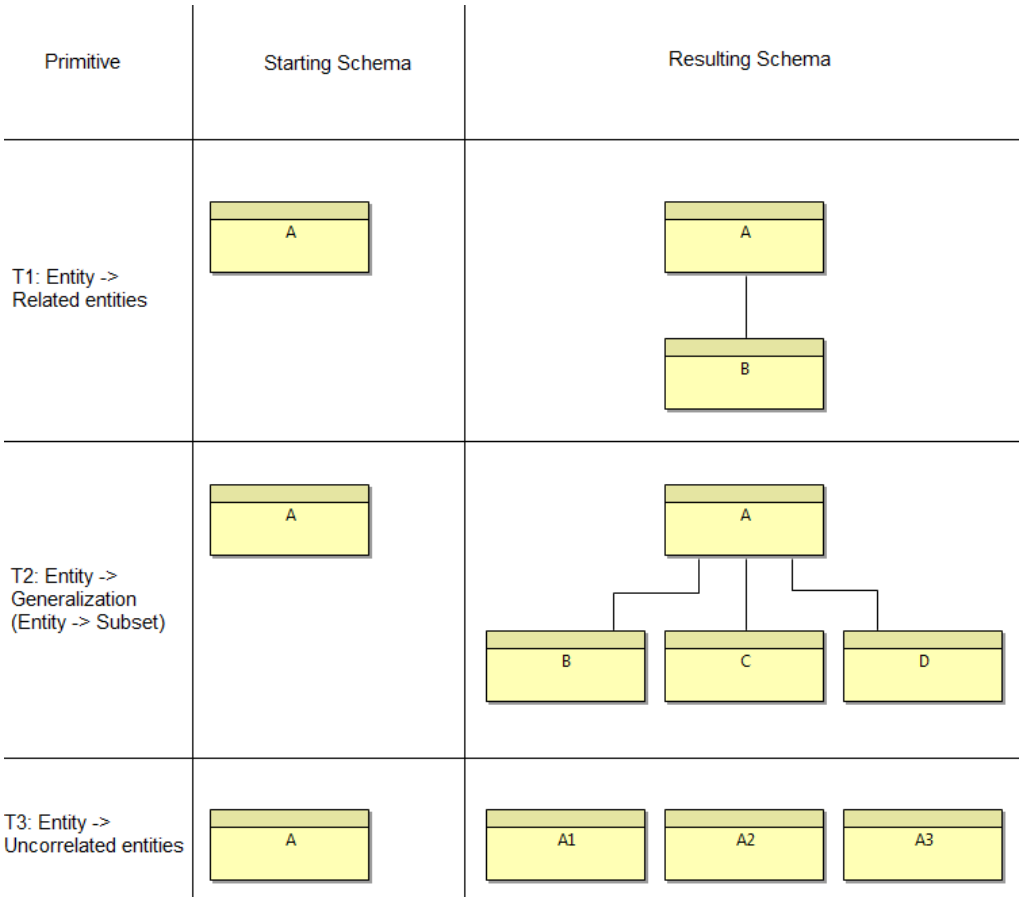

Figure 2: Top-Down Primitives Samples

Top-down primitives are presented and classified in figure 2 and exemplified in figure 3. Each primitive performs a specific type of refinement on the schema.

- Primitive T1, refines an entity into a relationship between two or more entities. The example in figure 2 in an application of this primitive
- Primitive T2 refines an entity into a generalization hierarchy or a subset.

- Primitive T3, splits an entity into a set of independent entities. The effect of this primitive is to introduce new entities, not to establish relationships or generalizations among them.

In figure 3, it´s presented an example of usage of primitives. In this example, there is a dependency between two entities in the starting schema. Consequently, in the final schema, there will never be a relationship between the derived entities by the two entities from the initial schema.

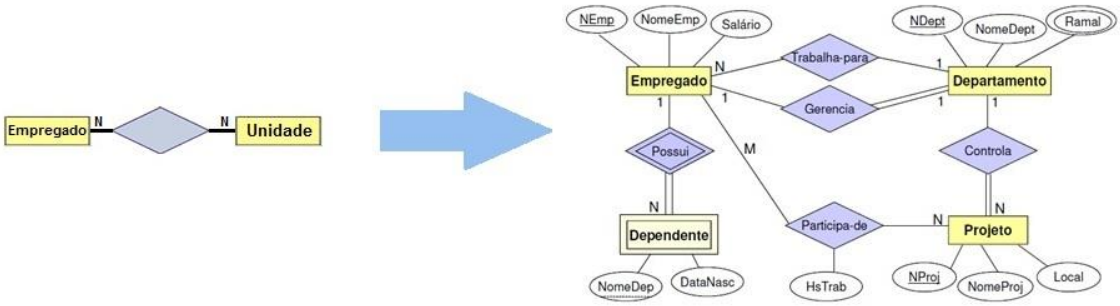

Figura 3: Example of usage of primitives

2.1.4.1.2 *Bottom-Up* Primitives

Bottom-Up primitives introduce new concepts and properties that did not appear in previous versions of the schema, or they modify some existing concepts. Bottom-up primitives are used in the design of a schema whenever we discover features of the application domain that were not captured at any level of abstraction by the previous version of the schema. Bottom-up primitives are also applied when different schemas are merged into a more comprehensive, global schema.

The most typical bottom-up primitives used in the course of a design are classified in figure 4:

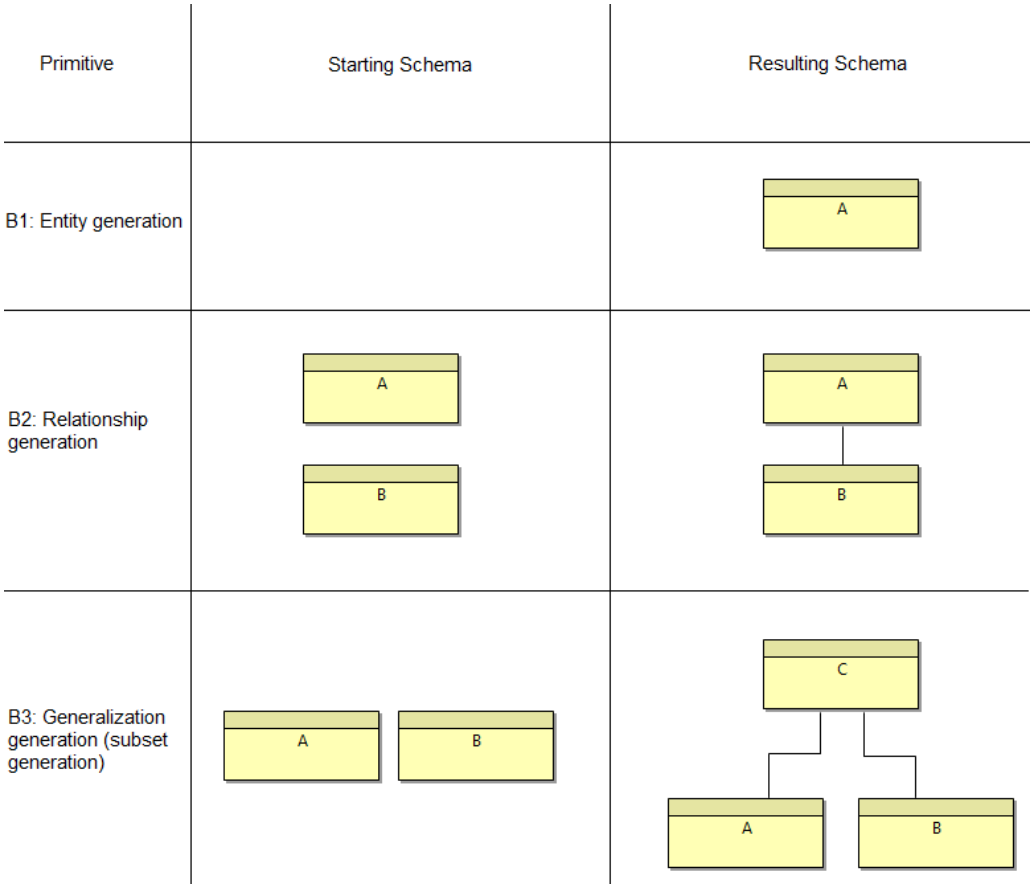

Figura 4: Bottom-Up Primitives Samples

Such primitives may cause changes in the schema developed, which can often lead to a partial or a nearly complete redesign.

2.1.4.2 Strategies for Schema Design

We distinguish four strategies for schema design: top-down, bottom-up, inside-out and mixed. Each of them is characterized by the use of particular types of primitives.

2.1.4.2.1 *Top-Down* Strategy

In the top-down strategy, a schema is obtained applying pure top-down refinement primitives; each primitive introduces mew detail in the schema. The process ends when all the requirements have been represented.

This strategy starts by applying the top-down primitives in a given domain. After this first transformation, it follows to another domain, unchanging the application layers.

It is important to notice that in a “pure” top-down strategy all concepts to be represented in the final schema must be present at each plane of refinement.

2.1.4.2.2 *Bottom-Up Strategy*

With the bottom-up strategy, we obtain a schema by applying pure bottom-up refinement primitives, starting from elementary concepts and building more complex concepts out of them; requirements are decomposed, independently conceptualized, and finally merged into a global schema.

With a different bottom-up approach, one can introduce fragments of a schema corresponding to clusters of concepts and then connect them.

The advantage of the bottom-up strategy is its simplicity: by attacking one fragment of the whole problem at one time, we can quickly produce draft versions of intermediate design products. The main disadvantage of the bottom-up approach is the need for restructuring the schema.

We can visualize the difference between the top-down and bottom-up approaches as follows: the top-down approach allows us to see very clearly a forest but not the trees, whereas the bottom-up approach allows us to see very clearly the trees but not the forest. [11]

2.1.4.2.3 *Inside-Out* Strategy

The inside-out strategy is a special case of the bottom-up strategy. Here, we first fix the most important or evident concepts and then proceed by finding first the concepts that are conceptually close to the starting concept and then “navigating” toward the more distant ones.

In the inside-out strategy the order of refinements is disciplined, as it is in the top-down approach. However, the levels of abstraction of the concepts introduced in successive versions of the schema are similar; this, the advantage of proceeding by abstraction levels is lost. [11]

2.1.4.2.4 *Mixed* Strategy

This strategy was the one adapted to our work due to its similarity with the method adopted with this work and its flexibility into using both top-down and bottom-up primitives.

The mixed strategy takes advantage of both top-down and bottom-up strategies, by allowing a controlled partitioning of requirements. When the application domain is very complex, the designer partitions the requirements into subsets, which are later separately considered. At the same time, the designer produces a “skeleton” schema, which acts as a frame for the most important concepts of the application domain and embeds the links between partitions. [11]

The overhead introduced by this step is rewarded, since the presence of the skeleton schema allows an easier bottom-up integration of the different schemas produced.

2.2 Related Tools

"Many enterprises start off using diagramming tools and spreadsheets to document their architectures. Although this can be useful initially, ensuring consistency of these documents becomes extremely difficult once artifacts appear in multiple places. For example, an application might appear on a diagram depicting a server, a diagram depicting a business process and a diagram depicting the application's interfaces. Changes to the application might require opportunities for inconsistency and inaccuracy." [12]

2.2.1 Modeling Tools

2.2.1.1 Archi (Institute of Educational Cybernetics)

After a careful analysis, the chosen modeling tool was Archi[3] from the *Institute of Educational Cybernetics*[4]. There were three main factors leading to this decision.

Archi is a free, open source, cross-platform tool and editor to create ArchiMate models. Archi is targeted toward all levels of Enterprise Architects and Enterprise Modelers. One of the main factors of this work was to provide the most cost-efficient solution to organizations with low budgets that are looking for a cross-platform ArchiMate modeling tool.

Archi fulfills the needs of most Enterprise Architects and associated stakeholders, but it can also be regarded as an introductory ArchiMate tool for those wishing to engage with the language before committing to a commercial solution. Since Archi is an easy tool to use and to model ArchiMate solutions, it will be easy for organizations to adapt it.

Since its introduction, Archi has been widely adopted for real-world use in the commercial and educational sectors and is used in-house by major global companies and consultants. It is rapidly becoming the de facto open source ArchiMate modeling tool. Another reason is due to its fully alignment with TOGAF[5].

2.2.1.2 Enterprise Architect (Sparx Systems)

Enterprise Architect[6] is a modeling tool produced by *Sparx Systems*[7] for the Windows platform. It is a multi-user, graphical tool designed to help teams build robust and maintainable systems.

[3] http://archi.cetis.ac.uk/, accessed on 18/07/2013
[4] http://www.bolton.ac.uk/IEC/Home.aspx, accessed on 18/07/2013
[5] http://www.opengroup.org/togaf/, accessed on 18/07/2013
[6] http://www.sparxsystems.com.au/products/ea/index.html, accessed on 18/07/2013
[7] http://www.sparxsystems.com.au/, accessed on 18/07/2013

Enterprise Architect is considered one of the state-of-the-art for modeling all types of frameworks. Since it can even model ArchiMate, it would be a viable option.

After a careful study on this tool, we quickly found that, even with all the diversity the tool offered it would lack a fundamental key for this work - Rules. Contrary to Archi that comes with a set of rules about the metamodel and focus solely on ArchiMate, Enterprise Architect wouldn´t follow any type of rules. In an organization environment, where user's don´t know how to work with ArchiMate, this would pose a dreadfull danger to our work, leading to a course of erroneous choices and bad modeling.

Given this terms, we had no other choice than abandon the idea of using this tool. Furthermore, since it´s availability is limited and at long terms it could present a higher cost than Archi, it would go against one of the goals set in this work.

2.2.2 Enterprise Architecture Tools

2.2.2.1 Enterprise Architecture Management System (Link Consulting)

Enterprise Architecture Management System (EAMS)[8] is an enterprise architecture tool produced by *Link Consulting*[9]. EAMS was the tool used to visualize the models developed during this work.

Using the same modeling tool (Archi) to visualize the models was proven insufficient due to the common problem in most of the architecture tool – there isn´t an effective way to see the changes (As-Is & To-Be) that are being made in the models. There is always the possibility to open simultaneous views in the same tool, but it´s not practical and not efficient enough.

EAMS give the possibility to visualize the As-Is and the To-Be through a life cycle bar (fig.5), tackling the problem of having several files to see the changes made through time on the models.

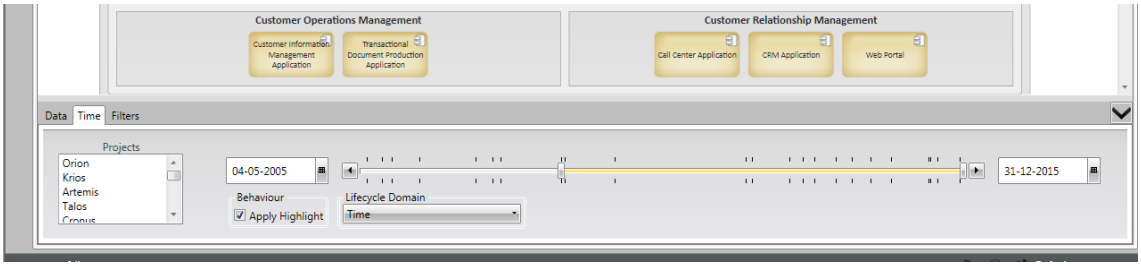

Figure 5: EAMS – Life cycle

Given this terms, EAMS was the solution proven most efficient. Due to the environment which this work was developed and the full support from *Link Consulting*, there were no costs associated with the use of their tool.

[8] http://www.link.pt/eams/, accessed on 18/07/2013
[9] http://www.link.pt/, accessed on 18/07/2013

During the course of this work, there were several implementations made, one of them was the possibility of integrating parsers with this tool (Fig. 6). Since we were using Archi as the modeling tool, we had to create a parser that would integrate with EAMS and could transform files from Archi to his data sources.

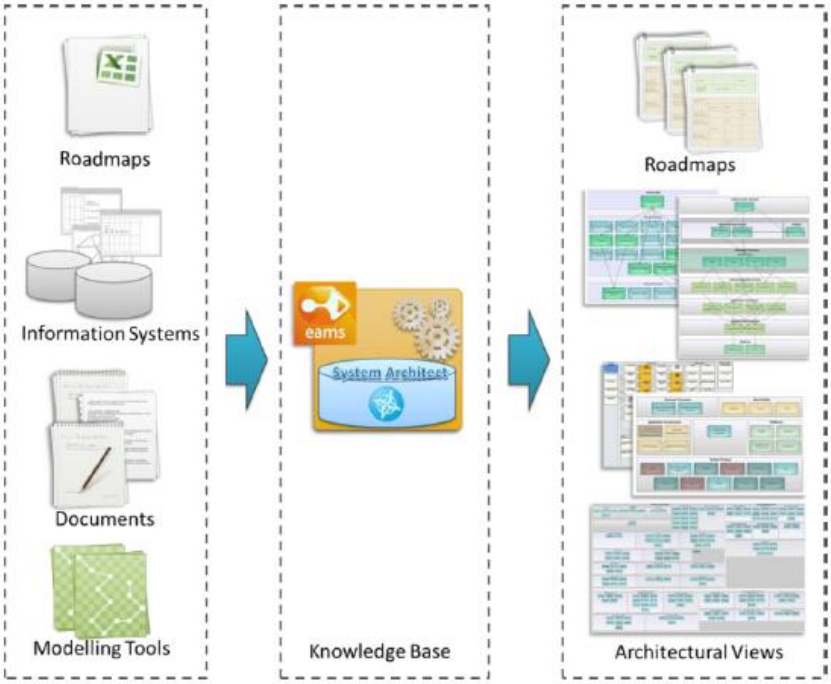

Figure 6: EAMS Process

2.2.2.2 Rational System Architect (IBM)

Rational System Architect[10] is an enterprise architecture tool produced by IBM[11]. It is an enterprise architecture solution for visualizing, analyzing, and communicating enterprise architecture and business process analysis. This solution provides decision support, process optimization, and integration into solution delivery.

This tool is already integrated as an information system in EAMS.

2.2.2.3 Gartner – Magic Quadrant for Enterprise Architecture Tools

The enterprise architecture tools market is constantly growing, especially when teams of enterprise architectures and chief information officers have unique challenges.

Although it is likely that enterprise architecture teams start off using drawing tools, spreadsheets, and other office productivity and content management tools to document and share their enterprise architecture, this approach becomes extremely difficult once artifacts appear in multiple places. For example, representations of an application

[10] http://www-03.ibm.com/software/products/us/en/ratisystarch/, accessed on 21/07/2013
[11] http://www.ibm.com/pt/pt/, accessed on 21/07/2013

might appear on a diagram depicting a server, a diagram depicting a business process and a diagram depicting the application's interfaces — all on different content management systems, spreadsheets or diagrams. Changes to the application might require updates in all locations of this data, introducing additional complexity and opportunities for inconsistency and inaccuracy.

Organizations should consider the need for their enterprise architecture tool solution to link with other solutions in the enterprise. Other solutions may contain data related to areas such as portfolio and project management, life cycle management and financial information. The need for end-to-end modeling capabilities, and to equip business leaders to make key decisions, may demand such integration. A classic choice will need to be made between implementing a comprehensive "suite" tool or using multiple best-of-breed tools and integrating them (Fig. 7).

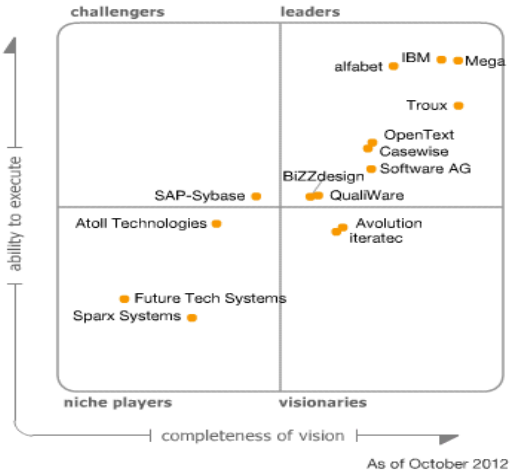

Figure 7: *Magic Quadrant for Enterprise Architecture Tools - Gartner* [13]

Adoption of an enterprise architecture tool, together with the discipline to use it properly and keep it up-to-date, can help address this issue of keeping all changes in sync by providing a single environment that enables changes to be reflected in the entire architecture across all viewpoints (such as business, information, technology and solutions). It is up to each organization to choose what should be the most appropriate tool for their purposes.

Since we do not want to "reinvent the wheel", and beside the EAMS have been developed on the IBM tool that in itself is one of the best in the market, we integrated the ArchiMate Framework through the use of Archi with EAMS to be able to module and visualize the models developed during this work.

2.3 Enterprise Modeling Languages and Frameworks

In this section we will talk about the frameworks used in this work. We can considerate the ArchiMate framework as being the principal framework used, since all the metamodels developed were based on their concept. Still, there are many well-established architectural frameworks and methodologies available for the systems

engineer. The list includes well-known frameworks, such as the Zachman Framework [14], the Spewak Methodology [15], the DoD Architectural Framework (DoDAF) [16], the Federal Enterprise Architecture Framework (FEAF) [17], and the Treasury Enterprise Architecture Framework (TEAF) [18]. All of these frameworks and methodologies were constructed to support the development of information systems architectures.

2.3.1 TOGAF & ADM

TOGAF is an architecture framework - The Open Group Architecture Framwork. TOGAF is a tool for assisting in the acceptance, production, use, and maintenance of enterprise architectures. It is based on an iterative process model supported by best practices and a re-usable set of existing architectural assets [19].

The Architecture Development Method (ADM) describes how to derive an organization-specific enterprise architecture that addresses business requirements. The ADM is the major component of TOGAF and provides guidance for architects on a number of levels. It provides a number of architecture phases (Business Architecture, Information Systems Architectures, Technology Architecture) in a cycle, as an overall process template for architecture development activity. It provides a narrative of each architecture phase, describing the phase in terms of objectives, approach, inputs, steps and outputs. It provides a cross-phase summary that cover requirements management.

The ADM consists of a number of phases that cycle through a range of architecture domains that enable the architect to ensure that a complex set of requirements is adequately addressed. The basic structure of the ADM is shown in figure 8.

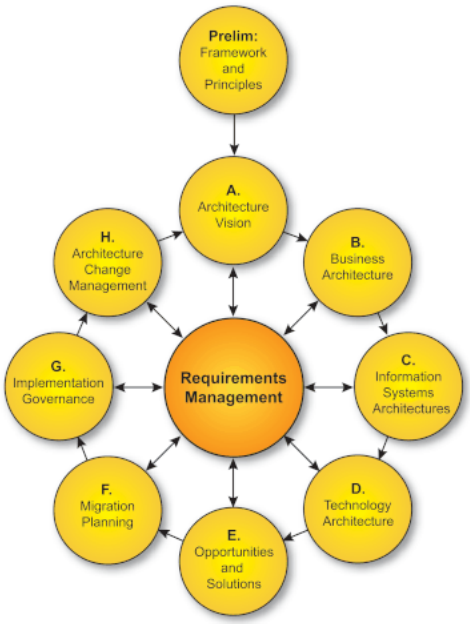

Figura 8: Artifacts TOGAF & ADM[12]

[12] Introduction to ADM - http://pubs.opengroup.org/architecture/togaf8-doc/arch/ ; Accessed on 12-11-2012

The ADM is applied iteratively throughout the entire process, between phases, and within them. Throughout the ADM cycle, there should be frequent validation of results against the original requirements, both those for the whole ADM cycle, and those for the particular phase of the process. Such validation should reconsider scope, detail, schedules, and milestones. Each phase should consider assets produced from previous iterations of the process and external assets from the marketplace, such as other frameworks or models [20].

2.3.2 ArchiMate Framework 2.0

The unambiguous specification and description of enterprise architecture's components and especially of their relationships requires an architecture modeling language that addresses the issue of consistent alignment and facilitates a coherent modeling of enterprise architectures.

The role of the ArchiMate standard is to provide a graphical language for the representation of enterprise architectures over time (i.e., including transformation and migration planning), as well as their motivation and rationale. The evolution of the standard is closely linked to the developments of the TOGAF standard and the emerging results from *The Open Group* forums and work groups active in this area. As a consequence, the ArchiMate standard does not provide its own set of defined terms, but rather follows those provided by the TOGAF standard [21].

A key challenge in the development of a general metamodel for enterprise architecture is to strike a balance between the specificity of languages for individual architecture domains, and a very general set of architecture concepts, which reflects a view of systems as a mere set of inter-related entities.

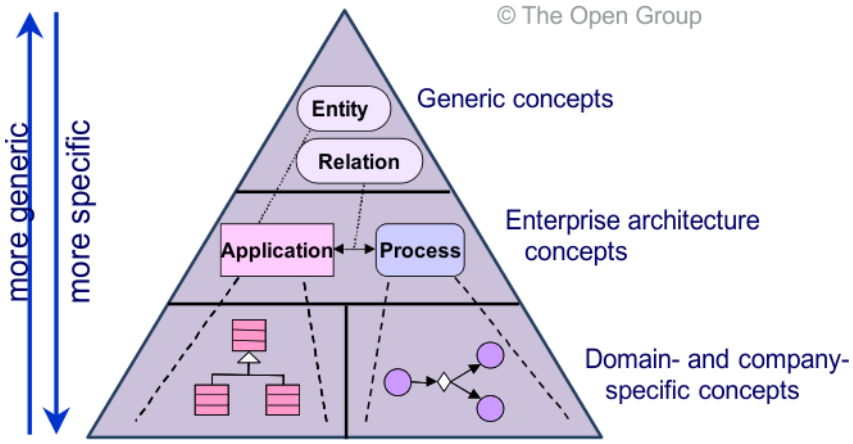

Figure 9: Metamodels at Different Levels of Specification[13]

[13] http://pubs.opengroup.org/architecture/archimate2-doc/chap02.html#_Toc309639713, acessed on 24/07/13

In this work, the metamodel from the ArchiMate 2.0 Specification was used as the foundation to perform the evolutionary maintenance on the metamodel of the Public Administration. It can be divided in three principal layers and two extension mechanisms. The core concepts of ArchiMate focus on describing the architecture of systems that support the enterprise. Not covered are the elements which, in different ways, motivate the design and operation of the enterprise. These motivational aspects correspond to the “Why” column of the Zachman framework [22]. Each of these layers has specific rules, with specific concepts:

Business Layer: The structure aspect at the business layer refers to the static structure of an organization, in terms of the entities that make up the organization and their relationships. It contains services and products that customers use to trigger business processes which are practiced by the stakeholders.

Application Layer: Supports the Business Layer with applications and services realized by the technology layer.

Technology Layer: Contains all the services from the infrastructure of an organization. These services are compromised by hardware, software and network that support the two layers above.

Motivation Extension Layer: The Motivation extension of ArchiMate adds the motivational concepts such as goal, principle, and requirement. It addresses the way the enterprise architecture is aligned to its context, as described by motivational elements. In addition, the Motivation extension recognizes the concepts of stakeholders, drivers, and assessments. Stakeholders represent (groups of) persons or organizations that influence, guide, or constrain the enterprise. Drivers represent internal or external factors which influence the plans and aims of an enterprise. An understanding of strengths, weaknesses, opportunities, and threats in relation to these drivers will help the formation of plans and aims to appropriately address these issues. The main reason to introduce motivational concepts in ArchiMate is to support requirements management and to support the Preliminary Phase and Phase A (Architecture Vision) of the TOGAF ADM [23], which establish the high-level business goals, architecture principles, and initial business requirements.

Implementation and Migration Extension Layer: The Implementation and Migration extension of ArchiMate adds concepts to support the late TOGAF ADM phases, related to the implementation and migration of architectures: Phase E (Opportunities and Solutions), Phase F (Migration Planning), and Phase G (Implementation Governance). This extension includes concepts for modeling implementation programs and projects to support program, portfolio, and project management, and a plateau concept to support migration planning.

The ArchiMate Framework 2.0 metamodel are shown in figure 10:

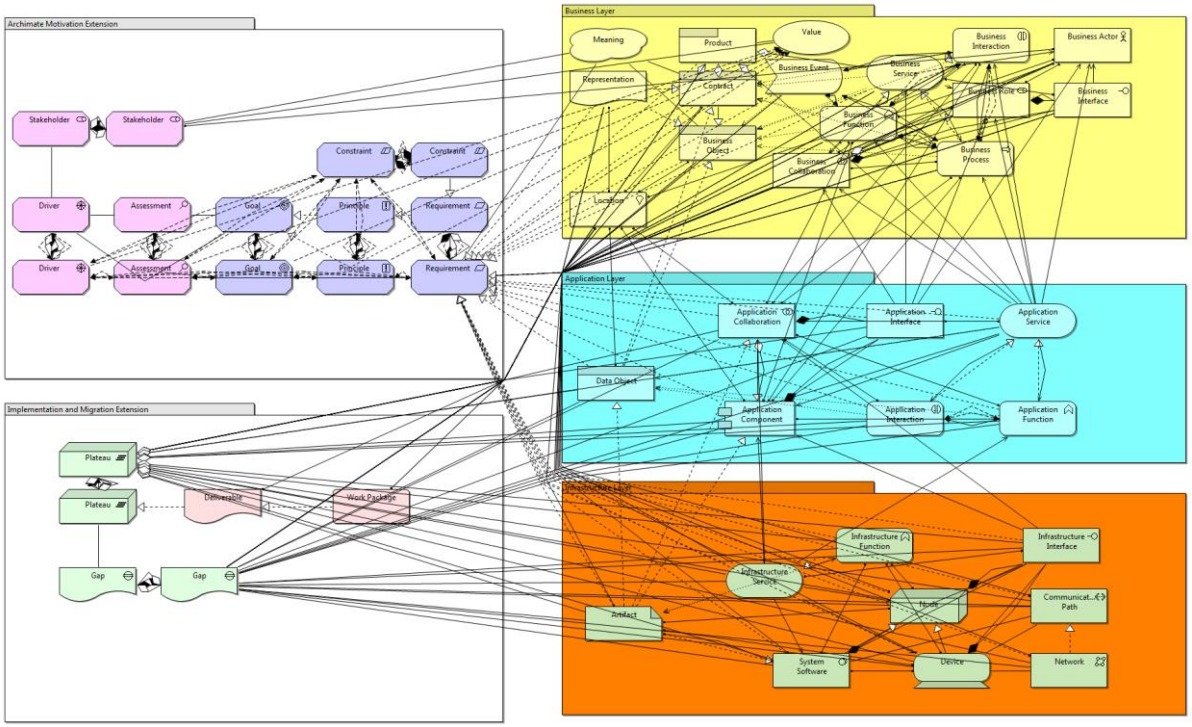

Figure 10: ArchiMate Metamodel and relationships

The structure of the core ArchiMate language closely corresponds with the three main architectures as addressed in the TOGAF ADM. This is illustrated in Figure 11. This correspondence would suggest a fairly easy mapping between TOGAF views and the ArchiMate viewpoints.

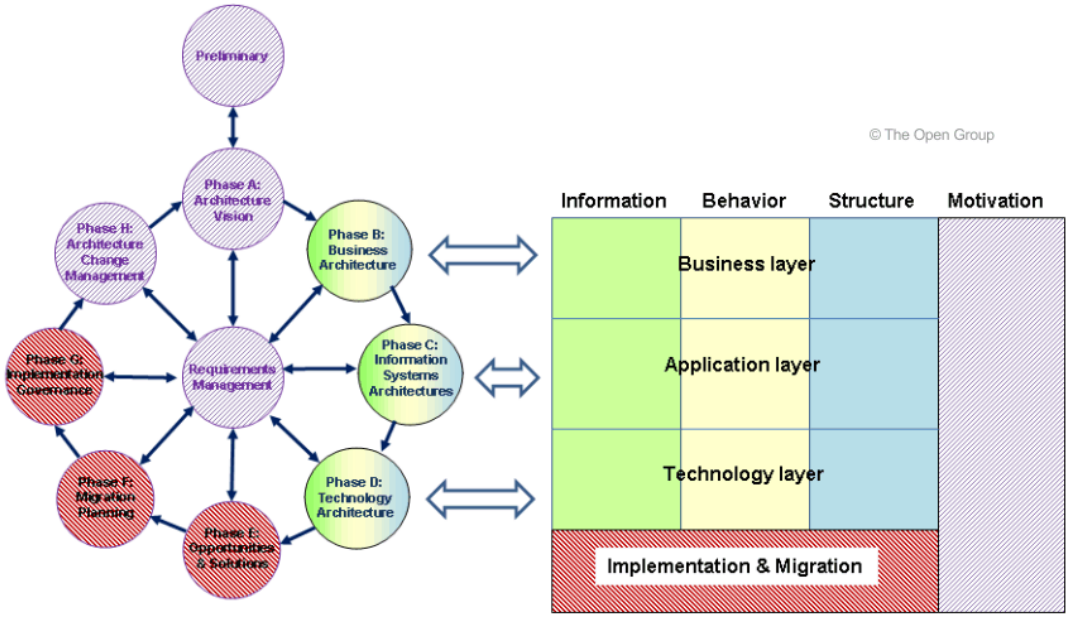

Figure 11: Correspondence between ArchiMate (including extensions) and TOGAF[14]

[14] http://pubs.opengroup.org/architecture/archimate2-doc/chap02.html#_Toc309639713, acedido 24-07-13

Although some of the viewpoints that are defined in TOGAF cannot easily be mapped onto ArchiMate viewpoints, the ArchiMate language and its analysis techniques do support the concepts addressed in these viewpoints. While there is no one-to-one mapping between them, there is still a fair amount of correspondence between the ArchiMate viewpoints and the viewpoints that are defined in TOGAF. Although corresponding viewpoints from ArchiMate and TOGAF do not necessarily have identical coverage, we can see that many viewpoints from both methods address largely the same issues.

TOGAF and ArchiMate can easily be used in conjunction and they appear to cover much of the same ground, although with some differences in scope and approach.

2.3.3 Unified Modeling Language

The *Unified Modeling Languagem* (UML) is an open standard for managing objects through graphical interfaces.

This language is used to build and display artifacts of a given system by following rules of good practice. The most common in this framework is the view it offers over the systems and elements that are connected. Besides being intuitive, is simple and clear enough for anyone in an organization can interpret the results shown.

This language was used to transform ways more complex into simplistic ones.

Of course there are many other languages suitable for modeling organizations; from the IDEF [24], which is actually a family of languages for modeling and analysis of organizations, BPMN, EPC [25] among others.

3 Proposed Methodology

In this chapter we will aim at describing what kind of approach, method and methodologies were used for developing the work and evaluate the obtained results. Scientific investigation is defined as a "*work developed methodologically, when facing a problem, and for which one seeks an adequate solution of scientific nature*", accordingly to Salomon [26]. The same author defines that scientific work is characterized by a "*written scientific investigation and processing of the methodologically approached questions*".

Thus, when developing a solution, the approach and methods used are what assures the credibility of the results obtained. Therefore, when developing a certain type of work, the validity of the obtained results must be supported through the utilization of a scientific method that, not only supports the work, but also allows for its repeatability [27].

In this section we will present the DSRM, the research methodology we used in our research and the strategies that can be used to evaluate the artifacts that result from the research methodology.

3.1 Approach

The existent approaches are divided into two major categories [28]: quantitative approaches and qualitative approaches. Since the objective of this work is not the pursuit of the theory revolving work evaluation methodologies, we will only make an overview of the chosen methods.

In this work we followed a quantitative approach. This kind of approach has its roots on the technological field, allowing the investigators to study the problem and giving a proper solution. The main characteristics are the fact that the role of the investigator is fundamental in order to collect data, and it has a high proximity with the studied phenomena [29]. Considering the type of work being executed, this approach fitted as we were inserted into the real context revolving the main problem.

In terms of methods, we used the design science research methodology (DSRM). In the inductive method, we start from particular observations and try to develop a generic solution. The usage of this method relates to the fact that we were inserted into a real context with a real problem, existent over different organizations, and then identified which one of them were common to the public administration.

3.2 Design Science Research Methodology (DSRM)

As stated earlier, this research followed the Design Science Research Methodology (DSRM). This research methodology has been chosen because it is appropriate for the research that seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts. DSRM is also active with respect

to technology, engaging in the creation of technological artifacts that impact people and organizations.

To overcome such organizational problems, proposes the creation and evaluation of artifacts which can either be:

Construct – vocabularies; **Model** – abstractions and representations; **Method** – algorithms and practices; **Instantiation** – implementations and prototypes.

As previously stated this research will focus on a method, since methods provide guidance on how to solve problems [42].

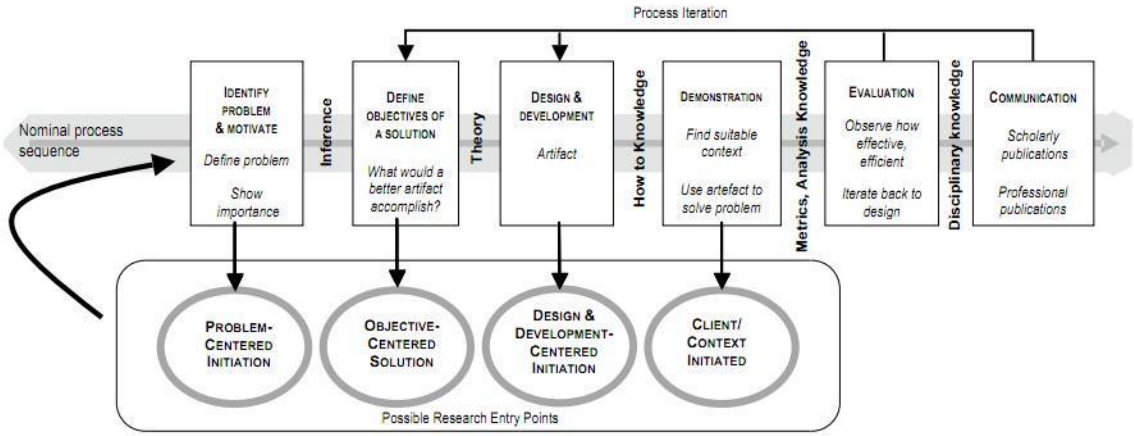

Figure 12: The DSRM process [41]

Apart from artifacts, DSRM is based on a process. This process is highly iterative and includes precise methods needed to be done in order to produce and evaluate the artifacts. There are six steps in the DSRM process, which can be seen in Fig. 12 and it is important to guarantee that each of these steps should be done sequentially to achieve the expected results:

Problem identification and motivation: define the specific research problem and justify the value of a solution (section 1.2, Problem Statement). It may be useful to atomize the problem conceptually so that the solution can capture its complexity.

Definition of the objectives for a solution: infer the objectives of a solution (section 1.3, Thesis Objectives) from the problem definition and knowledge of what is possible and feasible, they can be either quantitative or qualitative. The objectives should be inferred rationally from the problem specification;

Design and Development: The creation of the artifact that supports the defined objectives. This activity includes determining the artifact's desired functionality (Chapter 2, Related work) and its architecture (Chapter 4: Proposal) and then creating the actual artifact (Chapter 5: Demonstration; section 5.2, Archi2EALang);

Demonstration: The actual proof that the artifact developed solves the problem purposed (Chapter 5: Demonstration). To do so the artifact is used to solve one or more instances of the problem. This can be achieved by experimentation, simulation, case study, proof or other appropriate activity (Chapter 6, Evaluation);

Evaluation: Measurement of how can the artifact produced be an effective solution to the problem. The initial objectives of the solution are compared to the actual results obtained from the demonstration using knowledge of relevant metrics and analysis techniques. After evaluation, the process can be iterated back to activity 3 (to improve the effectiveness of the artifact) or continue on to communication (Chapter 6, Evaluation);

Communication: The communication step is fundamental because only with support from the experts in the field it is possible to assure the problem and the artifacts are important, useful, novel, rigorous and effective. This step was accomplished with the submission of this thesis.

4 Proposal

In this chapter we present the different set of steps included in our proposed methodology for evolving the metamodel, which we followed during the execution of this work, whereas in the previous chapter we presented the scientific methodology used while developing this work. It is therefore imperative for the development of the solution to be aligned with the methodology in order to enable its application. Facing the problems identified on section 1.2, considering the related work on chapter 2 - particularly the Archimate Framework 2.0, and the evaluation methodology to be applied, we devised a coherent sequence of steps/phases that allowed the obtaining of a solution.

The development of the solution started with a transformation of the current metamodel aligning with the Archimate framework, from which resulted the solution of this thesis. During the next sections we provide further insight on each of the steps of the proposed methodology, as well as on the work realized in them.

4.1 Context

In context we aimed at defining which concepts of the public administration we want to target and what steps are needed to create the solution for this work. This includes not only the steps taken to evolve the metamodel of the public administration, which represent an important role of this work, but also the tools created and used directly related to the process of the evolutionary maintenance. The concepts served as basis for this work are enumerated as follows:

Business Layer:

- Serviços de Negócio
- Processos de Negócio
- Entidades Informacionais
- Localização
- Organização
- Contratos

Application Layer:

- Soluções

Infrastructure Layer:

- Comunicações
- Software
- Serviços Tecnológicos
- Artefactos
- Equipamento

For each of these concepts we proceeded to its representation using the ArchiMate Framework 2.0 from a set of gathered data, e.g., project documentation and meetings with people enrolled at AMA. Here, we were able to gather different sets of information that allowed us to create a vision of the As Is.

Furthermore, to be possible to relate each concept to the process of being born or death, in the tools developed, we had to create a set of rules so that the users could easily define the birth date and death date. Thus, some properties types had to be associated to the modeling tool Archi. The properties are:

- eDay - Used for dates such as born dates and death dates
- URL - Used for URL links
- Boolean - Used for yes/no answers
- Hour - Used to define hours
- Name - Used to define names
- Currency - Used to define currencies
- Float - Used for numbers with comas
- Number - Used for numbers
- RealNumber - Used for real numbers
- Double - Used for numbers as doubles

Each of these property types is read by the tool created in this work and transformed in the language EALang, which is read by the data source of the visualization tool used. This process can be seen in figure 13:

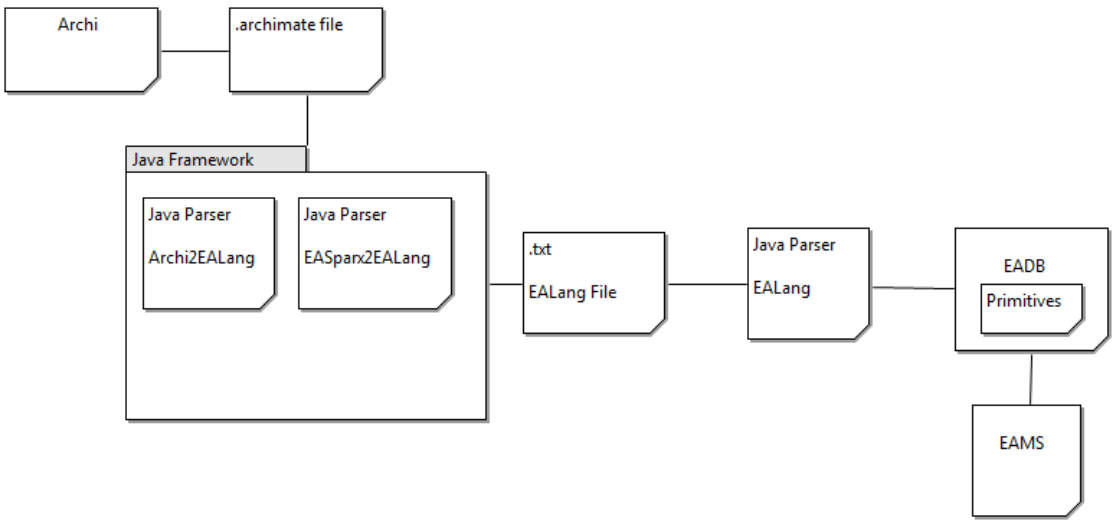

Figure 13: Process of the work of this thesis

4.2 Analysis

In analysis, we reunited and cross-reference the information gathered from the previous step and, using Archi as the modeling tool, proceeded to the representation of the concepts gathered and their properties, in a structure manner. Also, after further gathering information and with the start of the development of the parser, this triggered the end of the definition of the As Is, and started the processes of definition of the To Be. Each of the specific concepts of the public metamodel was represented using a specific concept of the Archimate Framework, and for each concept, the corresponding properties were assigned (fig. 14).

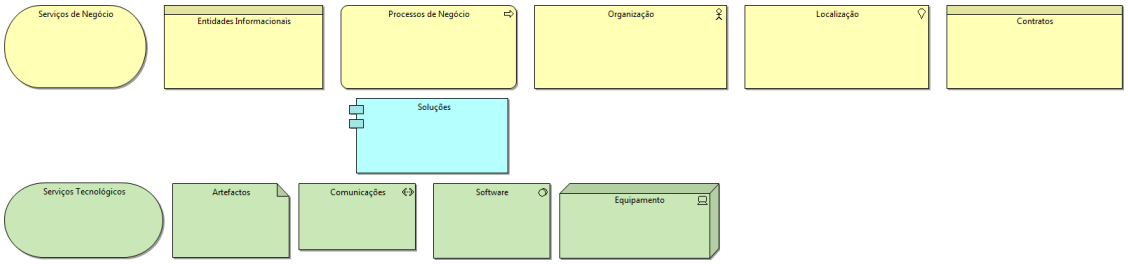

Figure 14: Evolved Concepts of the Public Administration

The same procedure was followed to deal with the more detailed concepts, this time by creating elements for each corresponding concept of the metamodel. Once the model was finished, we would proceed to his validation with the Archimate 2.0 metamodel, so that in the future, all the instances created would follow the rules established by this framework. At this point, we had enough information about each concept, a representation and a structured evolution of the metamodel. None the less, we had already developed work around the subject of traceability, and identified the relations inside the concepts analyzed. This was a necessary step towards understanding the relations between each of the concepts.

Once the parser was finished, and all the rules for implementation in EADB were met, we could initiate the process of the work of this thesis.

4.3 Summary

This chapter intends to start showing the context in which this work is integrated. We start by revealing which concepts of the public administration we want to target, and what steps were taken that made this implementation possible. Afterwards we analyzed the information gathered from the previous step and using the framework ArchiMate, we started the process of the To Be. Finally, after presenting the concepts in ArchiMate artifacts, we illustrated the process needed to transit the models done in the modeling tool, Archi, to the visualization tool, EAMS.

5 Demonstration

In this chapter we demonstrate how our proposals can be used in practice. We will start by explaining the steps taken to evolve the current metamodel in section 5.1. Finally, we will demonstrate the produced solution that made all the process of this work possible in section 5.2.

5.1 Proposed Metamodel

This section is dedicated to the evolutionary maintenance of the metamodel. We start by describing the current metamodel and its limitations with a practical case associated. Afterwards we will present the proposed changes to the metamodel and his impacts. Finally, we will demonstrate the roadmap of the primitives that can make the automation process of the evolutionary maintenance.

5.1.1 As-Is

As described earlier, this work was developed in collaboration with AMA, the agency responsible for the execution of the project of the metamodel for the Portuguese Public Administration. Despite the fact that this metamodel is currently being used, it is not organized, not perceptible and can induce a lot of errors. There is no division between the business, application and infrastructure layer. This metamodel (fig. 15) provided us the initial basis to work on:

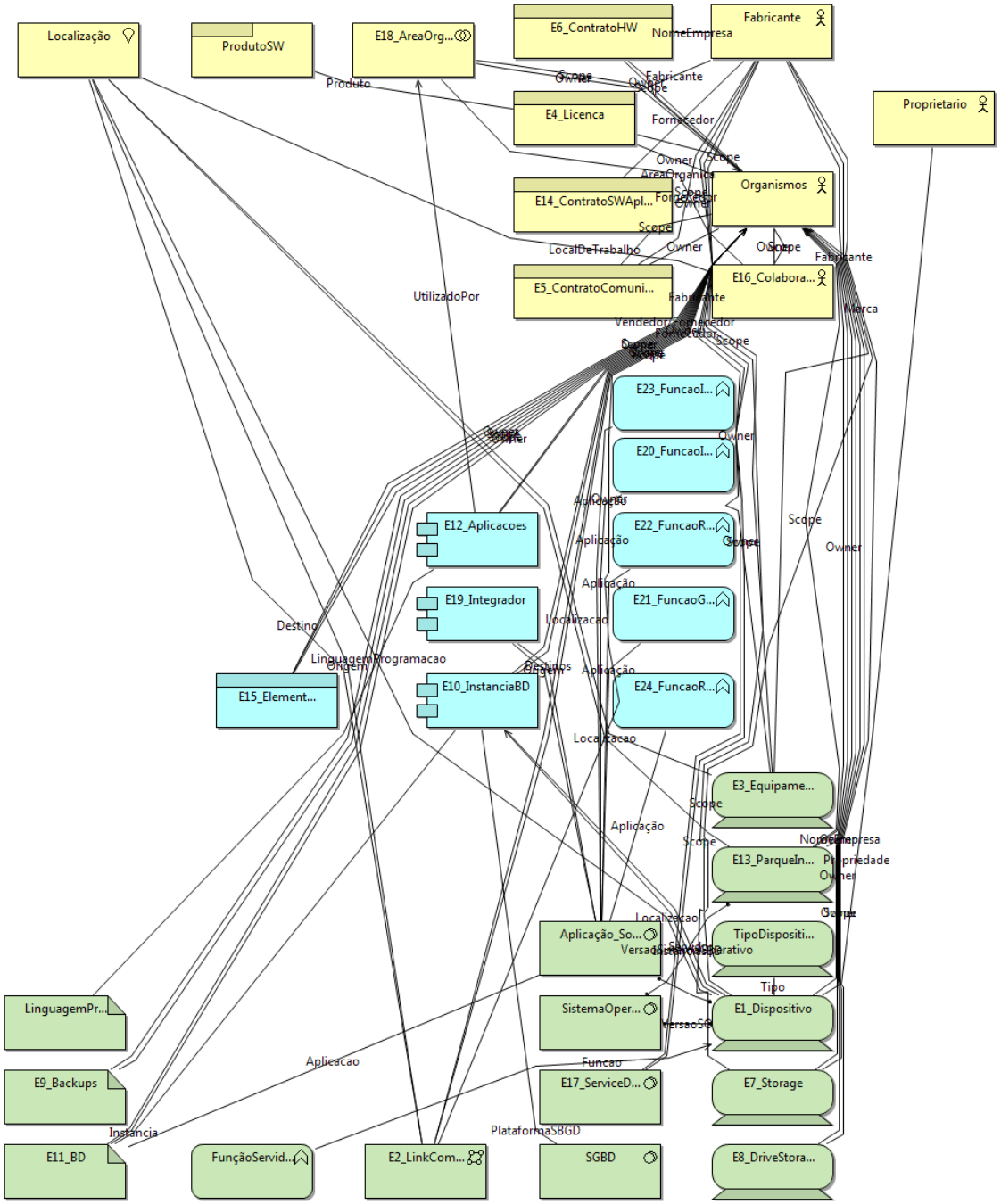

Figure 15: As-Is of the Portuguese Public Administration Metamodel

This architecture aims at providing a basis for each organization to develop its own architecture, giving the necessary support, and facilitating the use of common code and language through the different organizations [30]. The presented architecture gave us a formal basis to work on, providing us with a perspective into the problem, and allowed us to use it as starting point towards evolutionary maintenance, with the objective of accomplishing our work's goals and contributions already defined.

5.1.2 Proposed Changes (To-Be)

In this section we present the proposal of the evolutionary maintenance of the public administration metamodel to integrate with the ArchiMate metamodel concepts. All the proposed changes are implied in the sheets provided by the organization AMA in conjunction with the propositions made during the meetings.

In order to assess the list of modifications proposed to the actual version of the public administration metamodel, we must first consider the methodology our work was based on, the type of work developed, the intermediate results achieved, and the context of the work.

Hence, as referred in chapter 4, the evolution of the metamodel was based on the ArchiMate 2.0 framework. This option allowed us to add the concepts that were necessaries to the importation of the existing information. In the current metamodel (fig. 14) there are several concepts that may mislead information as referred in section 5.1.

ArchiMate divides the metamodels in 3 main groups - Business Layer, Application Layer, and Infrastructure Layer. Each of these layers is compromised by several concepts (chapter 2, section 2.3.2).

Since our goal was to evolve the metamodel into a more clear, rule-based and well-constructed metamodel, we started by analyzing the main concepts that were fundamentals to the public administration. We divided, as the ArchiMate metamodel, into the three main groups and created the respective concepts. Afterwards, we specialized each of the main concepts, going one level lower, allowing us to be more detailed. Finally, in the infrastructure layer, some concepts needed even more detail, requiring going one more level lower. In the end, the evolved metamodel is simple with a well-built structure and rule-based.

The main concepts are:

<u>Business Layer</u>

Serviços de Negócio: This concept should provide a unit of functionality that is meaningful from the point of view of the environment. It has a purpose, which states this utility. The environment includes the (behaviour of) users from outside as well as inside the organization. Business services can be external, customer-facing services (e.g., a travel insurance service) or internal support services (e.g., a resource management service).

Processos de Negócio: This concept describes the internal behaviour performed by a business role that is required to produce a set of products and services. For a consumer the products and services are relevant and the required behaviour is merely a black box.

Entidades Informacionais: This concept represents the important “informational” or “conceptual” elements in which the business thinks about a domain. Generally, it is used to model an object type (cf. a UML class), of which several instances may exist within the organization. A wide variety of types of

business objects can be defined. Informational entities are passive in the sense that they do not trigger or perform processes.

Localização: This concept is used to model the distribution of structural elements such as business actors, application components, and devices. This is modelled by means of an assignment relationship from location to structural element.

Organização: This concept performs the behavior assigned to (one or more) other concepts. It is an organizational entity as opposed to a technical entity; i.e., it belongs to the business layer. Organizations may, however, include entities outside the actual enterprise; e.g., customers and partners. Examples of these concepts are humans, departments, and business units.

Contratos: This concept may be used to model a contract in the legal sense, but also a more informal agreement associated with a product. It may also be or include a Service Level Agreement (SLA), describing an agreement about the functionality and quality of the services that are part of a product. A contract is a specialization of a business object.

Application Layer

Soluções: This concept is a self-contained unit of functionality. As such, it is independently deployable, re-usable, and replaceable. It performs one or more application functions. It encapsulates its contents: its functionality is only accessible through a set of application interfaces.

Infrastructure Layer

Comunicações: This concept is used to model the logical communication relations between nodes. It is realized by one or more networks, which represent the physical communication links. The communication properties (e.g., bandwidth, latency) of a communication path are usually aggregated from these underlying networks.

Software: This concept is a specialization of a node that is used to model the software environment in which artefacts run. This can be, for example, an operating system, a JEE application server, a database system, a workflow engine, or COTS software such as ERP or CRM packages. Also, system software can be used to represent, for example, communication middleware.

Serviços Tecnológicos: This concept exposes the functionality of a node to its environment. This functionality is accessed through one or more infrastructure interfaces. It may require, use, and produce artefacts. It should be meaningful from the point of view of the environment; it should provide a unit of functionality that is, in itself, useful to its users, such as application components and nodes. Typical infrastructure services may, for example, include messaging, storage, naming, and directory services. It may access artefacts; e.g., a file containing a

message. In this concept we adopted the TOGAF TRM [15] and each of the roles will be explained further ahead.

Artefactos: This concept represents a concrete element in the physical world. It is typically used to model (software) products such as source files, executable, scripts, database tables, messages, documents, specifications, and model files.

Equipamento: This concept is a specialization of a node that represents a physical resource with processing capability. It is typically used to model hardware systems such as mainframes, PCs, or routers. Usually, they are part of a node together with system software. Devices may be composite; i.e., consist of sub-devices.

Each of these concepts were considered primary and the only relationships existent will be between each of the concepts referred above. Furthermore, each one of the main concepts had proprieties, allowing the lower levels to inherit these same properties. Since with only one level we couldn´t be very specific, a second level was necessary:

Serviços de Negócio

Serviços IT: This concept will be to characterize the IT Governance that the department will adopt.

Serviços de Negócio: All the business services will be addressed in this concept.

Localização

Salas Técnicas: In all the departments there are rooms that have different purposes. This concept will contain all the different types of rooms and what equipment's each have.

Sede: Used to describe the location of the current headquarters of the organization.

Organização

Organismos: This concept will contain all the departments that a Ministry has to control.

Empresas: Used to describe all the organizations that are involved with the process and aren´t a part of any department from a Ministry.

Roles: This concept is used to define all the responsibilities of roles involved in the process for performing a specific behaviour.

Contratos

Contratos Serviços: Identification of the existing contracts for the services provided.

[15] http://pubs.opengroup.org/architecture/togaf8-doc/arch/chap19.html, acessed on 17/09/13

Contratos HW: Listing of maintenance contracts and support of the existing software. Each contract will have a brief description.

Contratos SW: List of active software contracts. Short description of the contract.

Contratos Comunicação: Identification of the existing contracts of the Global Telecommunications Operators.

Soluções

Aplicações: Will be used to identify all the applications that will perform one or more functions. Its functionality is dependent of a set of interfaces.

Componentes: Used to describe the application components used in the solutions provided by the business layer.

Serviços de Aplicações: The services application is defined as a service that exposes automated behaviour. It should be meaningful from the point of view of the environment; it should provide a unit of functionality that is, in itself, useful to its users. It has a purpose, which states this utility to the environment.

Dados: This concept will be used as a passive element suitable for automated processing. It should be a self-contained piece of information with a clear meaning to the business, not just to the application layer.

Comunicações

Comunicações Voz Fixa: Identification and characterization of communications services fixed voice hired by location and technology.

Comunicações Fixas Dados: Identification and characterization of the technology from the data services and/or fixed Internet in each location of the departments.

Comunicações Móveis Voz: Identification and characterization of the voice mobile contract, including SMS and data in cell phones.

Banda Larga Móvel: Identification and characterization of the broadband service contract (service or data cards and/or not mobile internet in mobile data)

Soluções M2M: Identification of existing communications solutions between machines (M2M).

Software

Software Servidores: Listing of the major applications in each of the servers. Brief description of each of these applications

Bases de Dados: Listing of existing databases in different servers. Brief description of the databases deployed in each server.

Software Sistemas: Designed to operate and control the components from the application layer. System software can include the operating system or any utility software that can help analyse, configure, optimize and maintain the computer.

Plataformas: Used to describe the software installed in the databases such as Oracle.

Serviços Tecnológicos

Data Interchange Services: It provides a specialized support for the interchange of information between applications and the external environment. These services are designed to handle data interchange between applications on the same platform and applications on different (heterogeneous) platforms. An analogous set of services exists for object-oriented data interchange, which can be found under Data Interchange services and Externalization services. [31]

Data Management Services: Central to most systems is the management of data that can be defined independently of the processes that create or use it, maintained indefinitely, and shared among many processes. [31]

User Interface Services: User interface services define how users may interact with an application. Depending on the capabilities required by users and the applications. [31]

Graphics and Imaging Services: Graphics services provide functions required for creating, storing, retrieving, and manipulating images. [31]

International Operation Services: International operation provides a set of services and interfaces that allow a user to define, select, and change between different culturally-related application environments supported by the particular implementation. In general, these services should be provided in such a way that internationalization issues are transparent to the application logic. [31]

Security Services: Security services are necessary to protect sensitive information in the information system. The appropriate level of protection is determined based upon the value of the information to the business area end users and the perception of threats to it.

To be effective, security needs to be made strong, must never be taken for granted, and must be designed into an architecture and not bolted on afterwards. Whether a system is standalone or distributed, security must be applied to the whole system. It must not be forgotten that the requirement for security extends not only across the range of entities in a system but also through time.

In establishing a security architecture, the best approach is to consider what is being defended, what value it has, and what the threats to it are. [31]

Location and Directory Services: Location and directory services provide specialized support for locating required resources and for mediation between service consumers and service providers.

The World Wide Web, based on the Internet, has created a need for locating information resources, which currently is mainly satisfied through the use of search engines. Advancements in the global Internet, and in heterogeneous distributed systems, demand active mediation through broker services that include automatic and dynamic registration, directory access, directory communication, filtration, and accounting services for access to resources. [31]

Network Services: Network services are provided to support distributed applications requiring data access and applications interoperability in heterogeneous or homogeneous networked environments. [31]

System and Network Management Services: Information systems are composed of a wide variety of diverse resources that must be managed effectively to achieve the goals of an open system environment. While the individual resources (such as printers, software, users, processors) may differ widely, the abstraction of these resources as managed objects allows for treatment in a uniform manner. The basic concepts of management - including operation, administration, and maintenance - may then be applied to the full suite of information system components along with their attendant services.

System and network management functionality may be divided in several different ways; one way is to make a division according to the management elements that generically apply to all functional resources. [31]

Transaction Processing Services: Transaction Processing (TP) services provide support for the online processing of information in discrete units called transactions, with assurance of the state of the information at the end of the transaction. This typically involves predetermined sequences of data entry, validation, display, and update or inquiry against a file or database. It also includes services to prioritize and track transactions. TP services may include support for distribution of transactions to a combination of local and remote processors.

A transaction is a complete unit of work. It may comprise many computational tasks, which may include user interface, data retrieval, and communications. A typical transaction modifies shared resources. Transactions must also be able to be rolled back (that is, undone) if necessary, at any stage. When a transaction is completed without failure, it is committed. Completion of a transaction means either commitment or rollback.

Typically a TP service will contain a transaction manager, which links data entry and display software with processing, database, and other resources to form the complete service. [31]

Software Engineering Services: The functional aspect of an application is embodied in the programming languages used to code it. Additionally,

professional system developers require tools appropriate to the development and maintenance of applications. [31]

Operating System Services: Operating system services are responsible for the management of platform resources, including the processor, memory, files, and input and output. They generally shield applications from the implementation details of the machine. [31]

Equipamento

IT Processamento: This concept describes all the equipment that has computational power and can process information.

IT Storage: IT Storage describes all the storage that have a physical frame.

IT Comunicações: Listing of all the equipment used to communicate without computational power.

Ministério da Defesa: This concept had to be created due to the particularity of some devices used in the Ministry of Defence.

Ministério da Educação: This concept had to be created due to the particularity of some devices used in the Ministry of Education.

Ministério da Saúde: This concept had to be created due to the particularity of some devices used in the Ministry of Health.

Going more deep than two levels would start to lower the performance, more cost-efficient and more complex. Nevertheless, an exception had to be made in the infrastructure layer, more precisely in the equipment concept. This exception occurred due to the high detailed equipment and divergent goals:

Equipamento – IT Processamento:

Computador Desktop: Concept used to describe all desktop computers in the department at a given location.

Servidores: Concept used to describe all servers in the department at a given location.

Portátil: Concept used to describe all laptops in the department at a given location.

Equipamento – IT Storage:

Drives Storages: List of logical disks (LUNs) in the various storage systems.

Storages: Listing of equipment and existing SAN Storage. For storage we mean systems with storage capacity of shared information.

Backups: Listing of equipment dedicated for data backup.

Equipamento – IT Comunicações:

Equipamento Comunicações: Description of the equipment's to perform communications (Voice, Data, Fixed, Mobile, Wan Lan).

Video Conferência: Identification and characterization of existing videoconferencing systems.

Finally, with all the proposed changes above referred, we achieved a more simple, less cost-efficient and efficient metamodel. The evolved metamodel is represented in figure 16 below:

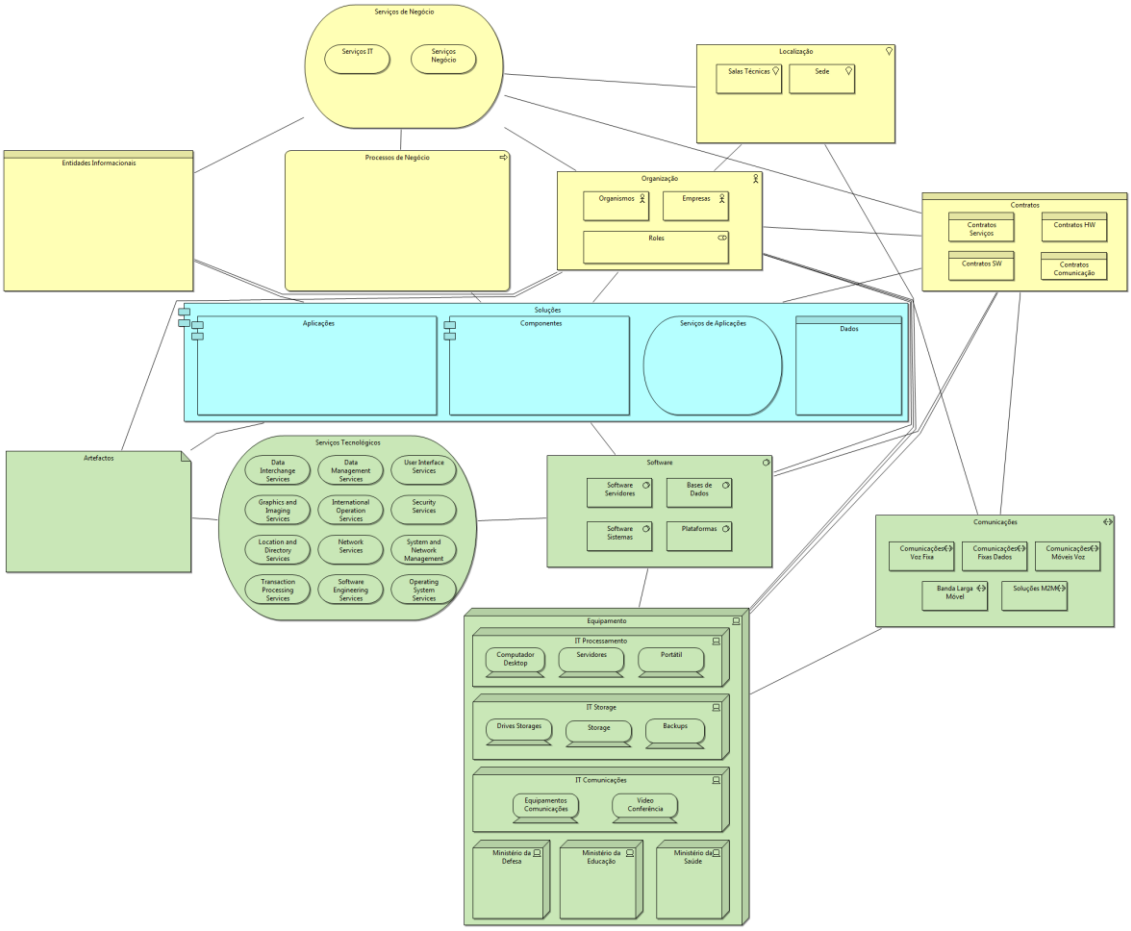

Figure 16: Developed metamodel of the Public Administration

In comparison with the first version, the concepts are clearer and shouldn´t pose any mislead as it will be described in chapter 6.

5.1.3 Primitives

The primitives were used in order to make a part of this work the more automated as possible. After the information is parsed and added to the database, the primitives will transform the information from the current metamodel to the evolved metamodel. In this section we will demonstrate the types of primitives used and the operations they will perform.

5.1.3.1 Types of Primitives

As explained in section 2.1.4.1, there are two types of primitives, the Top-Down and the Bottom-Up. In this section we will present the types of primitives that were used and how they will be represented to develop the roadmap for the automation of this work process.

Top-Down:

- Creation of a subset by aggregation of an entity: **(T2: Entity -> (Entity_1; Entity_2;…))**
- Copy of attributes and instance of an entity to another entity: **(Tw: Entity -> New entity (Attribute_1; Attribute_2;…))**
- Delete of an entity: **(Tx: Entity)**
- Rename of an entity: **(Ty: Name of Entity -> New name of entity)**
- Entity to attribute**: (Tz: Entity -> Name of entity which becomes an attribute of)**

Bottom-Up:

- Creation of an Entity: **(B1: Entity)**
- Creation of a new Relationship: **(B2: Relationshiptype (Entity_1 ; Entity_2))**
- Creation of an entity by aggregation of others: **(B3: NewEntity (Entity_1; Entity_2; …))**
- Creation of new attributes: **(B4: Entity (Attribute_1; Attribute_2;…))**

Each one of these primitives has operations associated that will be described in the next section. In the appendix 2 it is demonstrated the logic behind each of the primitives used.

5.1.3.2 Operations

Each one of the primitives has an operation associated. As explained in section 2.1.4.1, a Top-Down primitive have simple structure: the starting schema is a single concept, and the resulting schema consists of a small set of concepts. The Bottom-Up primitives can be identified by introducing new concepts and properties that did not appear in previous versions of the schema.

Top-Down:

- **T2:** Primitive T2, refines an entity into a aggregation hierarchy or a subset.
- **Tw:** Primitive Tw, creates a new attribute in a previously defined entity or relationship that already exists in a previously defined entity or relationship. It also copy the instance of the defined entity or relationship.
- **Tx:** Primitive Tx, deletes an entity and any relationship that is previously defined for this entity.
- **Ty:** Primitive Ty, renames an entity or a relationship with a new name.
- **Tz:** Primitive Tz, deletes an entity or a relationship that is previously defined. It also creates a new attribute in a previously defined entity or relationship which will have the same name as the deleted entity.

Bottom-Up:

- **B1:** Primitive B1, generates a new entity. This primitive is used when the designer discovers a new concept with specific properties that did not appear in the previous schema.
- **B2:** Primitive B2, generates a new relationship between previously defined entities.
- **B3:** Primitive B3, creates a new entity that is elected as an aggregation (either a subset or an aggregation hierarchy) among previously defined entities.
- **B4:** Primitive B4, generates a new attribute and connects it to a previously defined entity or relationship.

These primitives were used in the metamodel provided by the organization AMA and the practical case will be studied in the next section.

5.1.3.3 Primitives Process (As-Is -> To-Be)

Starting with the As-Is, the primitives initiate with the first metamodel shown in section 5.1.1, figure 15. In this part of the thesis we used a metamodel previously defined – As-Is; and refine it with the primitives in order to evolve it into a better structured one and more efficient – To-Be.

When applying the primitive's roadmap shown in Appendix II, we expect to obtain a series of transformation in the model. First, we start by tackling individually the business process layer, application layer and finally the infrastructure layer.

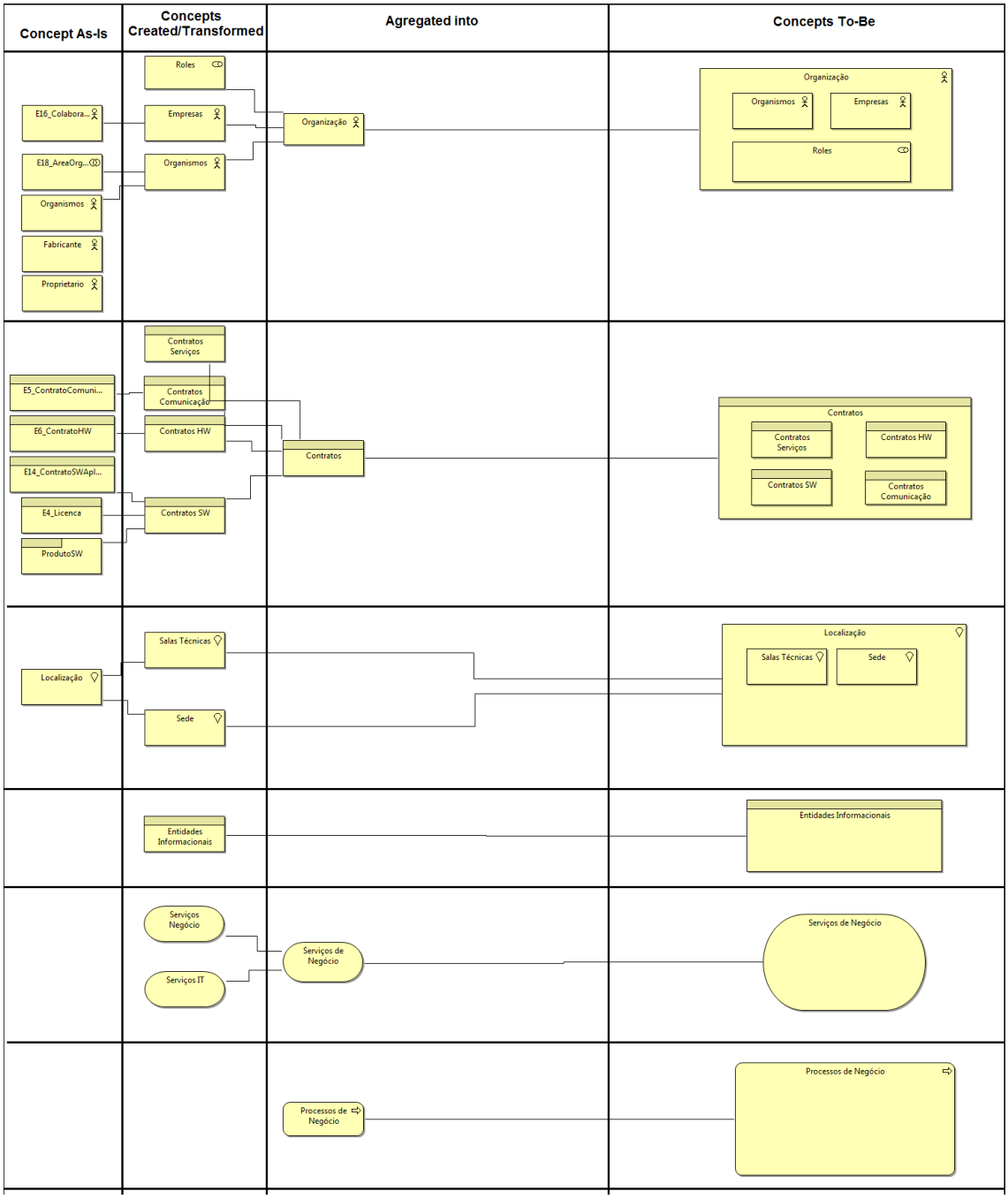

Figure 17: Business Layer Primitives Process

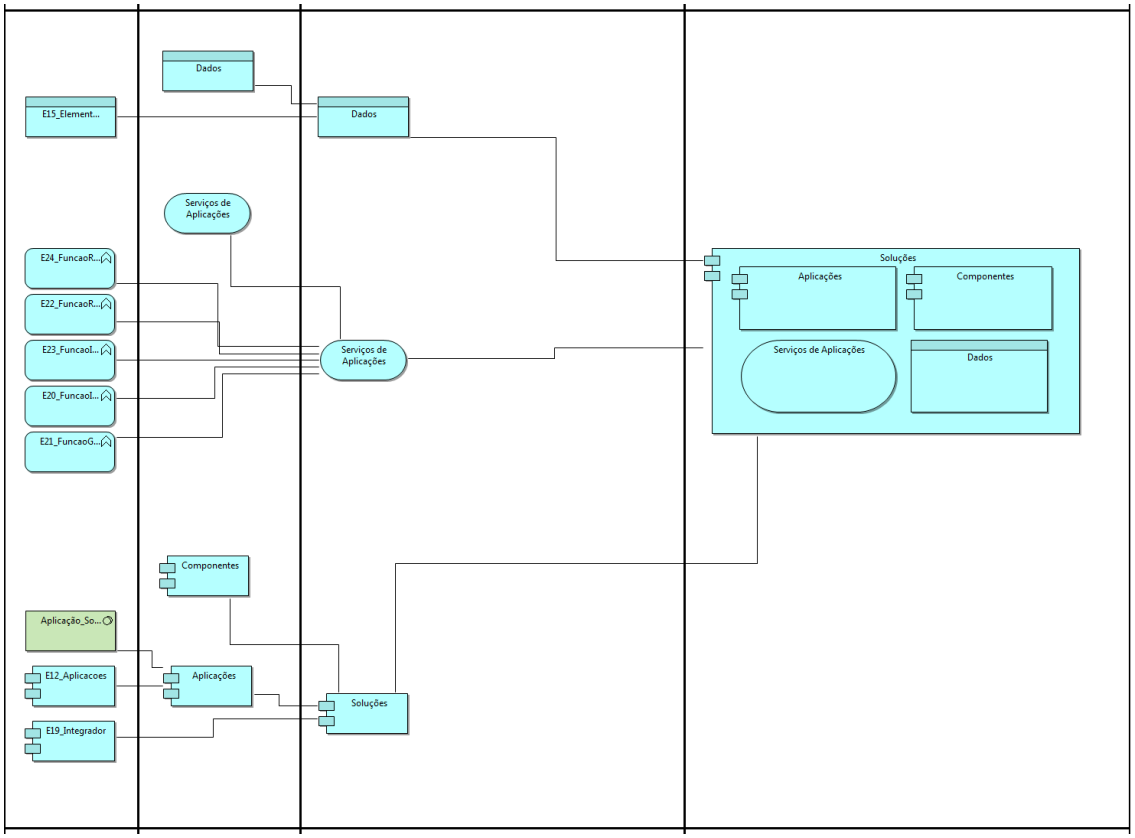

Figure 18: Application Layer Primitives Process

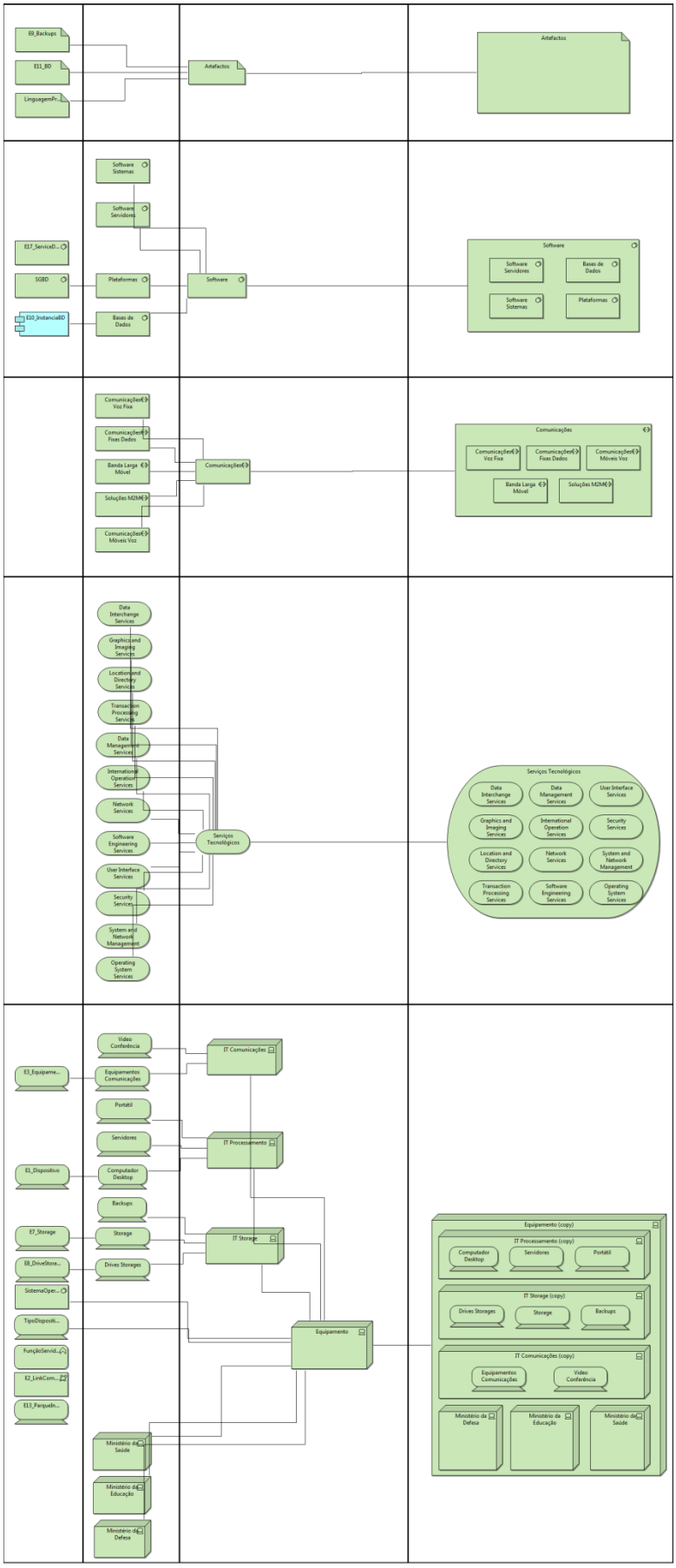

Figure 19: Infrastructure Layer Primitives Process

When all the concepts are created and transformed, we will proceed to the creation of the relationships between each concept as shown in Appendix III.

5.2 Archi2EALang

The solution Archi2EALang is a real-world implementation of this thesis subject. The solution was developed by the author as a collaboration of Link Consulting. Some of its capabilities and functionalities go beyond the scope at hand and are the result of collaboration with other colleagues and students. The descriptions here addressed concern only designs and implementations done solely by the author.

5.2.1 Introduction

English is a powerful language. With it, we can write plays and sonnets, grocery lists, business plans and contracts. If computers could understand English, we would have little need for Java or any other programming language.

Computers understand very little, and arguably nothing at all. They can add numbers and move strings of text but cannot in themselves understand the idea of, say, doing something 10 times. So on one hand we have English, which is enormously expressive, and on the other hand we have computers, which understand almost nothing. This is why programming languages emerge. A programming language such as Java is a compromise between the expressive eloquence of English and the primitive receptive abilities of a computer.

As the language used in this thesis, Java is an excellent choice of language when the problem at hand requires giving computers precise commands.

Parsers help computers, which work with objects, to cooperate with people, who read and write text. In practice, particularly in Java-based parsers, this implies that parsers translate text into objects. For example, a parser can translate a textual command into another textual format. [36]

Parsing is the process of structuring a linear representation is accordance with a given grammar. This definition has been kept abstract on purpose to allow as wide an interpretation as possible. [37]

This solution was developed with the intent to translate the grammar given by the modulation tool – Archi, to the grammar provided by the language EALang that can be read by the visualization tool EAMS from Link Consulting.

5.2.2 Parsing

The purpose of this section is to describe a more technical part of the solution developed. We will start by identifying the several types that the parser can identify and how it identifies them, how a class is processed and its relationships as well as the

instances. Afterwards we will describe how the properties are processed and the three modes that the database will handle.

Finally, we will talk about the interface and what is the goal of each button:

- ***Types***

By definition, all properties that are not declared will be considered as the type of **String**.

The types are declared with "[]" (without quotes), eg: **[Number]** Num. Contributor

In this example we declare that the property Num.Contribuinte is of type **Number**.

Using known types:

Types known by EADB and related statements in Archi are:

EADB	Archi	Description
Date	[Date]	Used to dates in the format YYYY / MM / DD
URL	[URL]	Used to URL links
Boolean	[Boolean]	Used to answer Yes / No - True / False
Hour	[Hour]	Used to set times in HH: MM: SS
Name	[Name]	Used to refer names. It is treated as a String.
Currency	[Currency]	Used to refer to values.
Float	[Float]	Used to numbers with commas.
Number	[Number]	Used to set numbers. It is treated as an Integer.
RealNumber	[RealNumber]	Used to define real numbers.
Double	[Double]	Used to set number doubles.
eDay	[eDay]	Used to define the born and death date in the format YYYY / MM / DD

Tabel 1: EADB Types

Using unkown types:

All types that are declared and are not on the list above will be treated as new types, for example: **[XPTO]** ComplexDescription

In this case, it creates a new type **XPTO** with the property ComplexDescription.

- ***Classes***

Creating classes:

All objects created in Archi will be treated as new classes, for example:

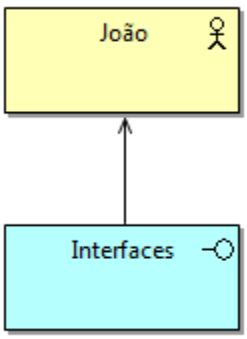

Figure 20: Relation between a Business Actor (João) and an Application Interface (Interfaces)

In this case it will create **two** classes. A class **Business Actor,** and a class Application **Interface**.

- ***Instances***

Creating Instances of classes:

For each object created in Archi, an instance of this object will be created. As in Figure 16, **two** instances are created, **one** for each class. This will create an instance **João** of the class **Business Actor** and an instance **Interfaces** of the class **Application Interfaces**.

The instantiation with values will be discussed in section Properties.

- ***Properties***

The properties are declared in Archi tool through the window "Properties".

Properties instance:

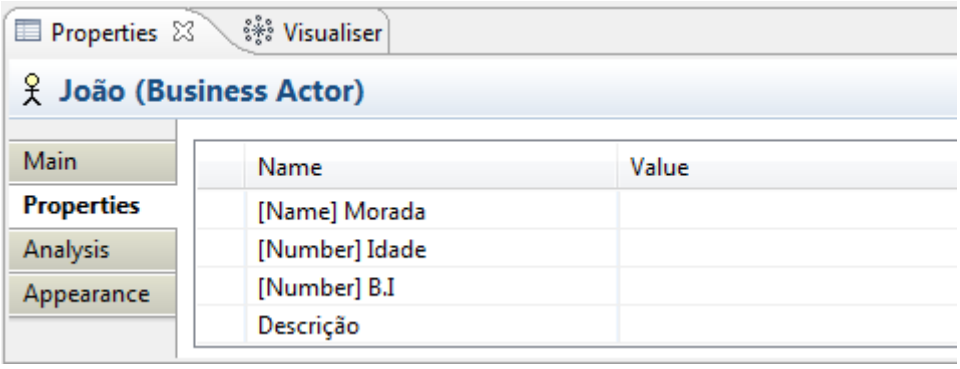

Figure 21: Properties Window from Archi

In this example, the object **João** has declared **four** different properties.

Properties with values:

When creating an instance with values, we have to fill the field named “Value” as in the following example:

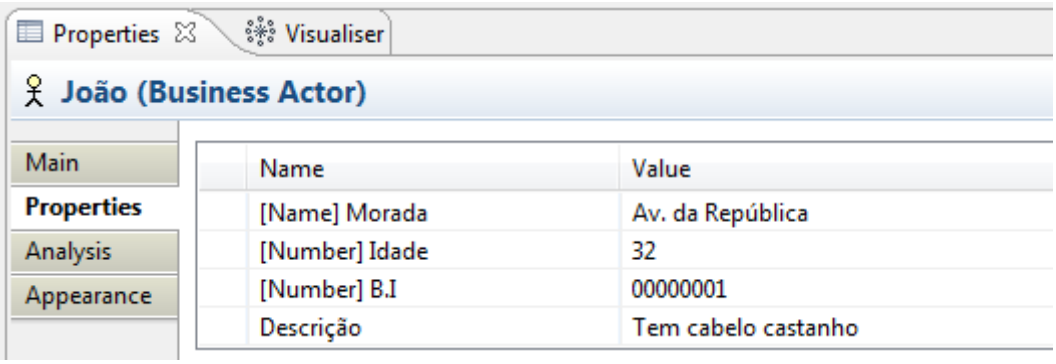

Figure 22: Properties Window with values

In this case it will be created the instance **João** from the class **Business Actor** with the **four** properties and its declared values.

- ***Mode***

There is an insert mode in EADB which is defined by the following:

Absolute: Uses U_Force mode. Always replaces the old value with the new. This mode is used as default.

Partial: Use mode U_Add. In case the property is a list, this mode will converge the original list with the new list. The properties that are not lists, work in the same way as the **Absolute** mode.

Fill Only Blanks: Uses the mode U_FOB. This mode will **only** populate the properties that do not contain values. If the property is NULL, the property will not be updated.

In the solution we can set modes in two ways - either globally or for each subject. To set globally we just need to select in the combo box our pretended option:

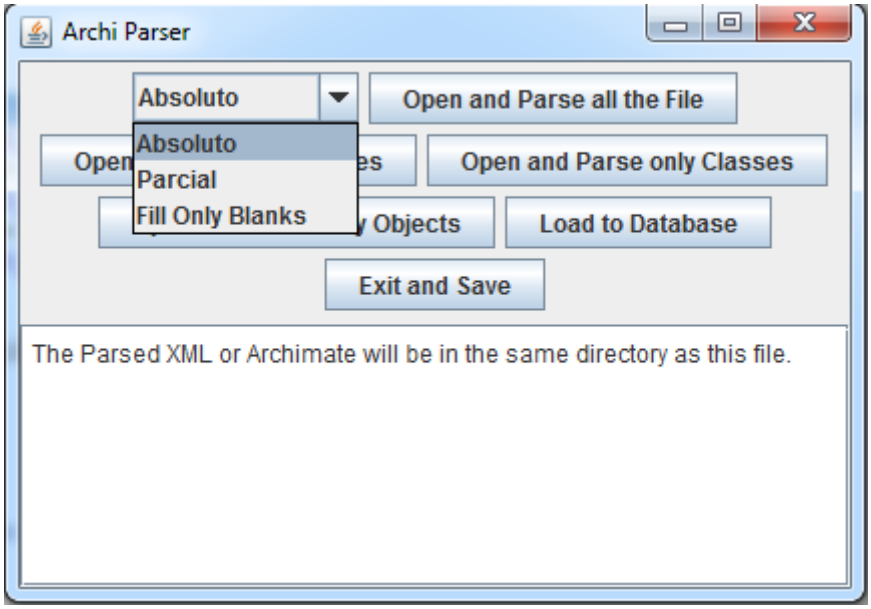

Figure 23: Global Interface Options

To define each object, it is necessary to define one of its properties as **Umode** as in the following example:

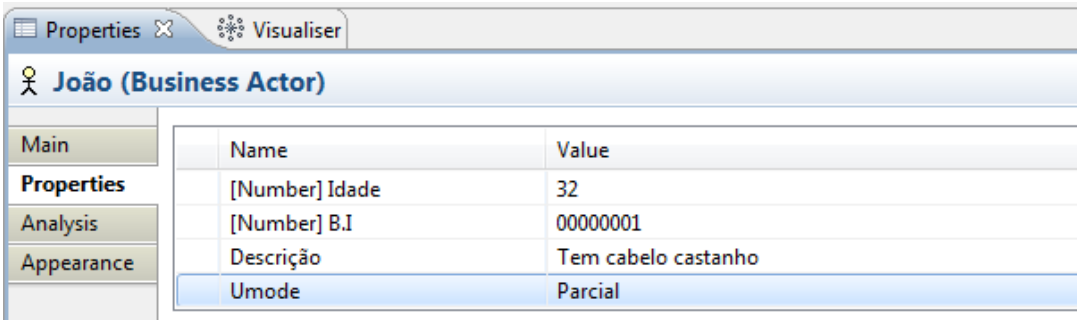

Figure 24 Umode in Archi Properties

- ***Interface***

There are four options available in the interface. Each time one of the chosen options is successfully executed, it is **mandatory** to **Exit and Save** so that the file is correctly processed:

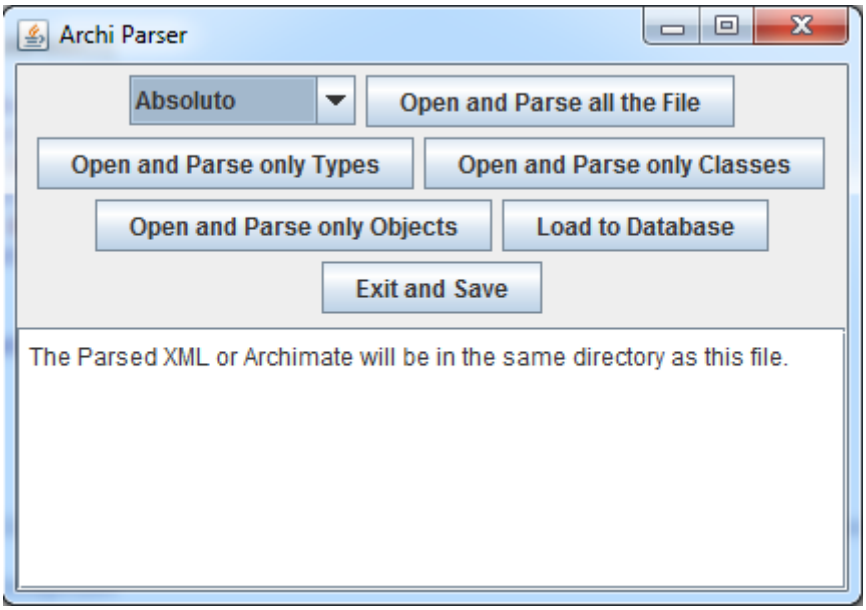

Figure 25: Interface Archi2EALang

Open and Parse all the File: This option will convert **the entire** file chosen to a file of type text.

Open and Parse only Types: This option will **only convert the types** you choose to a file of type text.

Open and Parse only Classes: This option will convert **only the classes** you choose to a file of type text.

Open and Parse only Objects: This option will convert **only the objects** you choose to a file of type text.

Load to Database: This option will convert **the entire** file chosen to the database EADB.

5.2.3 Framework Integration

Developers spend far too much of their precious time worrying about their backend database, its tables and their relationships, the names and parameters of stored procedures and views, as well as the schema of the data that they return. [36]

If in the future, new tools come to the market, and a parser is needed, why not integrate it with the existing parsers? That was the concept in which the parser was developed.

After its completion, the goal was to integrate it with a framework being developed. This framework had a parser already integrated from the tool EA Sparx.

Afterwards, the only step left was to integrate with EAMS (fig. 26).

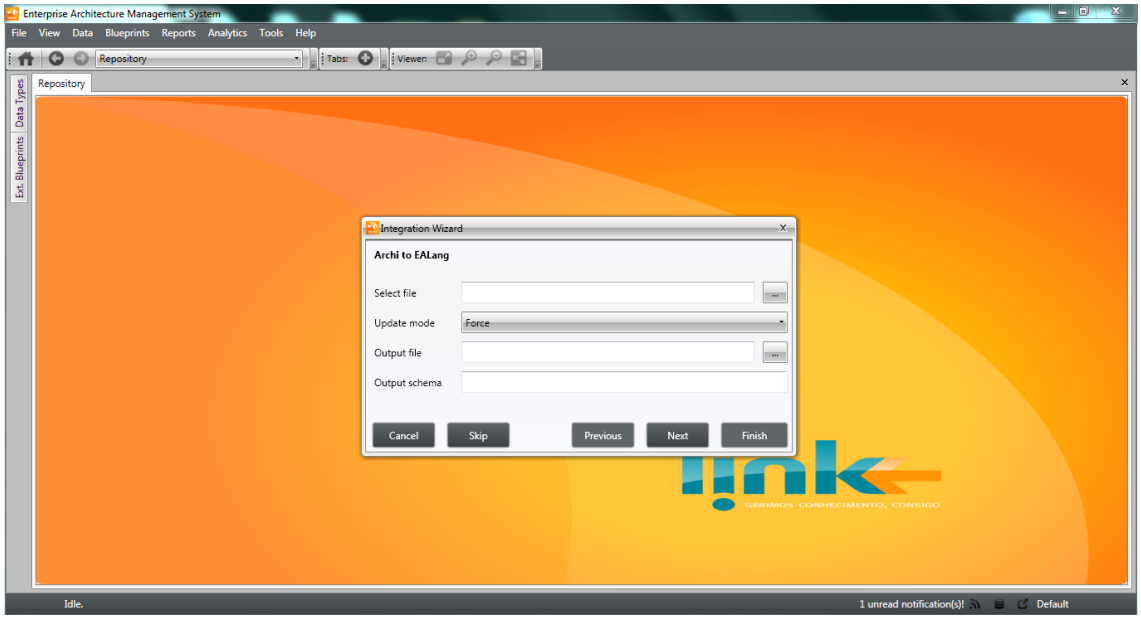

Figure 26: Framework integration with EAMS

As a future work, we hope that more parsers will be integrated in this framework, turning EAMS into a richer tool.

6 Evaluation

In order to evaluate our proposal [33] and to get some feedback on our design decisions, we used semi-structured interviews [34]. These interviews and questionaries' are presented in the appendix 1. In the next two sub-sections we explain how we conducted those interviews and present the main results.

6.1 Organizational Feedback

6.1.1 AMA - Agência para a Modernização Administrativa

As described earlier, this work was developed in collaboration with AMA[16], the agency responsible for the execution of the project of the enterprise architecture for the Portuguese public administration.

AMA - Agency for Administrative Modernization, P.I is the public institute in an indirect state administration whose mission is to operationalize the initiatives to modernize and boost the participation and involvement of different actors, institutions and responsible.

The AMA, I.P. mission is to develop, coordinate and evaluate measures, programs and projects in the areas of modernization and simplification of administrative and regulatory, management and distribution of electronic public services, within the framework of the policies defined by the Government.

Its tasks can be divided in:

Contribute to the definition of strategic guidelines and general policies related to electronic administration, administrative simplification and the distribution of public services, including interoperability in Public Administration;

Manage and develop networks of shops for citizens and businesses in systems integrated and specialized, while articulating with the care systems and voice network;

Promote the modernization of the supply and distribution of public services aimed at meeting the needs of citizens and businesses;

Promoting policies nature at a central, regional and local level in the area of information society, through the management of Internet spaces and others alike while administering and consulting the other entities with responsibilities in the information society;

Support the development and implementation of platforms and solutions for e-learning;

Ensure the external representation and to establish cooperation in the framework to other foreign entities, including within the European Union and Portuguese-speaking countries;

Give prior opinion and track projects in public investment (PIDDAC) and give a prior opinion on the allocation of European funds in the context of modernizing and simplifying administrative management and electronic;

[16] More about AMA at http://www.ama.pt

Streamline and coordinate inter-ministerial network of agents and modernization of administrative simplification;

Promote studies, statistical analyses while forward looking and stimulating activities for research, technological development and dissemination of good practices in the areas of administrative simplification and regulatory management and electronic;

Propose the creation of direct project teams, transient and inter-ministerial or interdepartmental, to achieve, development and evaluation of actions while ensuring a modernization of administrative simplification and regulatory framework, including through review of administrative law, in its simplification aspect of corrective. [32]

6.1.1.1 Evaluation

This thesis was made in cooperation with AMA while considering this organization as the practical case.

After the requirements elicitation obtained through the meetings and documents, a thoroughly study was made to ensure the correct evolution of the metamodel, tackling the main problems represented by the current metamodel.

Currently, there are several master's degree students at AMA using the metamodel produced from this work in order to test it and evaluate it as we are going to analyse in the section 6.2.

Since the evolved metamodel is based in ArchiMate 2.0, it is unavoidable the evolution of this framework. Despite this fact, there is always the possibility to continue the work done and keep up with the evolution of ArchiMate, resulting always on the goal of this thesis – to improve the efficiency of the information systems.

The solution developed is currently deployed and working at AMA, with the support from Link Consulting for futures improvements.

In appendix 3 it is demonstrated the roadmap developed for the purpose of this thesis regarding this practical case.

6.1.2 CDD – Centro Dados da Defesa

The CDD can be designated as a part of the structure from the SG (Secretária-Geral). The goal of this organization can be divided into five categories:

To ensure the provision of information technology to all bodies of the department of defence within the tasks provided by the governance of the defence;

To ensure the delivery of the infrastructure technology that supports shared information of the systems management;

To ensure the delivery of application systems and databases of the defence within the tasks provided by the governance of the defence;

To ensure the delivery of computer network defence, ensuring their adequate security, capacity, availability, interoperability and interconnection between all departments and agencies in the area of defence and other national and international entities, within the tasks provided by the governance model of the defence;

To ensure the support to users of the centralized information system management. [35]

6.1.2.1 Evaluation

In this study case, it was given the framework developed during this work and the EAMS solution.

In this study case we had to present a solution to tackle the complexity and the lack to maintain a controlled environment of the changes happening inside the CDD with the minimum cost possible. Since there was even modelling being performed in paper, it was evident that maintaining a rigorous control over the changes being made in the department was too much complex.

The solution provided by EAMS presented the best scenario possible, since with the timeline feature we could easily see the changes and impacts of each department or project was having. Thus, the only remaining factor was the modelling tools.

Attending the request of low-budget, it was presented the free modelling tool Archi, to represent the schemas needed to tackle the paper based methodology. Once a schema is finished with the required born and death dates, the framework will parse the archimate schema and EAMS will handle the visualization of the futures versions. Since Archi presents not only his free solution but also good policies for diagrams drawings, joining this strong side with the powerful features of EAMS was the most low cost, high efficiency solution possible.

This solution is currently deployed and working, with the support from Link Consulting for futures improvements.

6.2 Practitioners Feedback

6.2.1 AMA – António Barros

Every organism of the Public Administration has to send their proposals of their projects to be approved. However there is no automatic way to evaluate these architectures. This work focuses on creating a tool using metrics already developed in other work projects and assess whether they are going to meet the guidelines of the TIC from the Public Administration.

6.2.1.1 Evaluation

In this case, it was given all the process developed in this thesis. Since this work is being developed in cooperation with AMA, it had the same organization as context. The

work developed by this thesis helped the process to the extent that facilitated the introduction of architectures in EAMS in order to export them to XML.

After being exported to XML the metrics were applied in order to meet the guidelines of the TIC from the Public Administration.

Quoting António Barros:

"*From the evolved metamodel to all the process developed until the XML exportation, all helped in improving the efficiency of my work. The evolved metamodel is substantially better, more organized, more perceptible, with the correct division between the various layers (Business, Application, Infrastructure).*"

Since the parser was incorporated with the EAMS structure there is no to little difficulty on its usage. The only downside noticeable was the lack of error messages more elucidative.

6.2.2 AMA – Yessika Reinolds

As the study case written above, this work was developed in the same organization as the goal of this thesis. The goal of this case is to detect patterns and anti-patterns from existing architectures, so that in the future, organizations in the same business area that may want to develop enterprises architectures can have a base on which architectural patterns and anti-patterns already exists.

6.2.2.1 Evaluation

This study case is still in an early stage. Some testing was done to ensure the good functionality and performance promised by this thesis.

At the beginning, one of the requirements was to export architectural scenarios to XML files through EAMS, but there was a need of detail that couldn´t be reproduced through the process developed of this thesis.

The solution was to directly create the XML files, having as the foundation the evolved metamodel produced in this work. Develop enterprises architectures with a metamodel which has a set of rules and directives laid by an framework of reference such as Archimate 2.0, will be a major asset for the future development of this study case.

Quoting Yessika Reinolds:

"*The new metamodel is so much more organized and of simple perception, that will help me build my project with confidence since it is built on the foundation off the ArchiMate Framework. This will boost the productivity due to the low level of complexity, comparing with the older metamodel.*"

All the support needed for this study case to be succeeded will be provided through the next year.

7 Conclusion

This chapter will look at some of the ways knowledge-making demands of the research.[39] It will summarizes our thesis' main contributions, limitations and lessons learned. The results we obtained during our Evaluation step were quite positive and encouraging. Taking into consideration these results, we consider that our solution met its initial objectives.

The Portuguese Public Administration is being target of a modernization initiative which, among other aspects, contemplates the development of a metamodel. This action aims at addressing the incompatibilities existent between different information systems, on different departments, and their inefficiency in terms of dealing with information. There is a need for developing a metamodel with a good set of rules based on a respected framework, so that it can be perceived which are the entities and departments enrolled in the manipulation of the several concepts associated with business, application and infrastructure. Therefore, this will improve the work efficiency as well as the experience perceived by all the departments of the Public Administration.

With this work, we began by clarifying the problem and establish the motivation towards the solution. Then, we researched on the themes of the enterprise architecture, information parsing, state of the art languages and frameworks that allowed us to develop a coherent work. We researched on the scientific investigation methodologies that we used to support our work on scientific basis, and therefore guarantee its validity and continuity. The Design Science Research Methodology (DSRM) methodology has proven to be useful since an early stage of this work, and the solution's architecture is also aligned with its steps. After having a sense of perspective into the context of the work and the approaches to be taken, we devised a methodology for working towards a solution, with support on activities and objectives defined for each step. By using the knowledge gathered under this phase, and by applying it into the further development of the work, we were able to analyse and process the different data gathered and ultimately provide a revised version of the existent architecture, with improvements based on the entire work we developed. Finally, and after achieving our objectives, we adopted a critical perspective and made considerations about the work developed and eventual opportunities of improvement in future work around the subject. The subject and context of this work, as well as the people with whom we worked, were a sincere motivation towards its realization and completion. None the less, the fact that we integrated a project of significant dimension, already on the move, with multi-disciplinary teams and organizations involved, and for which there is a real need, put us in a position where we had the means and level of participation of the different intervenient that allowed us to perform the planned activities and with relative flexibility. Sometimes the elevated dimension of the project also proved to be a problem, especially while shifting between organizations. As we know, with sensitive materials there is always a pipeline of approvals that need to be granted and all that consumes a valuable resource: time.

We always had the notion that this was an ambitious work. In this case we were capable of developing relevant work for both the academic community and the professional world, for who we hope that our contributions prove useful. Finally, and on that note, we would like to leave the challenge open to anyone who continues working

on this theme, to attempt to evolve and enrich this solution. We are positive that its contribution to the related work on the area would be enormous, as well as a very differentiating factor and solid base to develop similar future work.

7.1 Lessons Learn

Over the course of this dissertation there were several aspects that were raised which are important to mention. These aspects resulted from the application of the DSRM process to this research, mostly in the problem identification step, the objective definition step, the design step and the evaluation step. We will structure this section to match these steps.

In the Problem Identification step we had issues in defining how the problem could be posed, since the Public Administration is too much complex, we had to define a set of problems that could be precise and very objective. We discovered that most of the information regarding the metamodels was outdated and lacked in some aspects (no division between layers, lack of information, lack of support, etc...). To better define the proposal to solve the problem, we had to look over many different approaches and other frameworks. Furthermore, many programming languages had to been considered due to the diversity of our problem.

The development of the Archi2EALang solution arouses several lessons. On one hand, the usability of the solution, on the other hand the complexity behind it. If we want a usable parser, we need to make it as simple as the majority of people with the least experience possible in the information technologies area can use it without any problems. The use of a simple's interface and techniques learned through the academic life, helped to tackle this problem.

Finally, through the evaluation we managed to gather feedback from the users that were going to use the solution developed in this thesis. The cooperation of these users, helped to correct bugs and modify the solution into a more efficient one. It is crucial, when implementing a solution into a real situation to have user inserted in the case that can identify problems with the process being developed.

On the other side, we have learned during the course of this thesis that the roadmap of the primitives we have developed weren´t implemented, and an example of this working side can´t be tested at the moment. If implemented, it can be used in production to improve even more the efficiency of the Public Administration. Finally, since the parser is a solution to translate only from the Archi tool, it has to be worked to be able to identify the modelling tool and parse any file from any modelling tool.

7.2 Contributions

Concretely focusing on the work developed, the research performed in terms of related work, and the fact that we had the support from Link Consulting and its work, was a major key factor into achieving the results we had. By already having an implemented metamodel, we were able to confine the scope of our work to improve part of it, making it more viable that attempting to develop a metamodel from ground zero. It surely had

been impossible to get to know all the public departments implicated on the design of the architecture, and produce a coherent and viable solution within the time available. With the methodology we devised and implemented, we were able to perform a full cycle of the investigation methodology.

Each of the steps from the DSRM allowed us to perceive and strict the relations between the different concepts, allowing us to always have a back work developed that supported the new step we were trying to execute, e.g., describe the relations between the entities from AMA's implemented metamodel and a specific context using roles, while already having the specific contexts modelled.

Given the type of work performed, there are changes and improvements that will have to be made in the future in order to continue aligned with the objectives of this work. Considering the questions in chapter 1, the work developed addressed a real problem happening not only in the Public Administration but in all organizations. The answer to Q1.1 and Q1.2 is answered in chapter 4, where we propose a methodology to realize this type of work, and the validation of the proposal was achieved by developing the solution made during the course of this thesis, proving that it was possible to evolve the current metamodel with the right solutions and tools. The validation to these two questions was presented in chapter 6.

From our perspective, there are two main contributions provided by the work developed, namely the evolved metamodel and the solution Archi2EALang.

In terms of the proposed information architecture, it is the result of the entire set of activities performed during this work, and it is the direct answer to Q1.1 and Q1.2.

None the less, the Q2.1 and Q2.2 was answered by developing a roadmap of the primitives to be used. The main contribution of this answer not only benefits the academic field, as it also has implications on the accessory contributions in the enterprise field, as the other main contribution, particularly to the academic field. The methodology, despite being implicated in the answers/contributions to questions Q1.1 through Q2.2, specifically targets the Q1.2.

7.3 Future Work

In this section we present a description of future works to further complement the knowledge presented in this dissertation.

Although we suggested a structure for the evolved metamodel, we did not suggest specific improvement measures or projects. That would imply extensive expertise about the public administration and its IT departments [40], which we could not acquire during the time period of this thesis. Moreover, due to the complexity of the thesis, there is the need to implement the roadmaps of the primitives directly in a MySQL server.

In order to provide a clear vision for what the future can and should be, we must make some considerations first. Taking into account the scope of this work, i.e., public administration, as well as the time and effort it would require to develop a solution that

would cover it entirely, one of the first measures we had to take was to restrict the amount of data to be analysed and processed.

The restriction was applied not only to the set of contexts analysed but also, in a higher level to the perspective in which the work was focused.

For instance, while this work is focused on the development of a solution and the evolution of the public administration metamodel, there are other perspectives that should be analysed, e.g., companies. Based on the methodology we presented, with direct relation to the percentage of the entire universe of the public administration analysed, the more of this universe is analysed, the likely it is to obtain a metamodel closer to a final and stable version. As stated earlier, this work is focused on the development of a solution and even so, there wasn´t a practical case where we could test our roadmap. With the contexts analysed and the four projects chosen for the study case, they provide mainly insight in the act of the stability and efficiency of the solution developed.

For each of the concepts, work should be done, towards defining what are the most important activities in their lifecycles, to what departments of the public administration are they related, who/what else intervenes and its relevant for their lifecycle, and which is their transversal relation to the rest of the information entities present in the public administration. An innovative and important approach to this work would be to have a dedicated group conducting more business-layer oriented work, identifying processes, and working on the development of an ontological view over the different concepts.

As important as evolving the metamodel, as a future work that as to be done, is to continue to develop parsers for other modelling tools and integrate them in the framework already existing, giving more advantage to EAMS in the enterprise architecture tools market.

Finally, since the roadmap of the primitives to use in the public administration metamodel is developed, the integration and development of this roadmap is essential and can be harnessed into a new thesis.

8 References

1. WATSON, Richard W (2000). An Enterprise Information Architecture: A Case Study for Decentralized Organizations. New York: IEEE Computer Society.

2. OPEN GROUP (2012). Archimate 2.0 Specificaction. 1st Edition. Chapter 1. Philadelphia: Van Haren Publications.

3. SOUSA, P., LIMA, J., SAMPAIO, A., PEREIRA, C. (2009). An Approach for Creating and Managing Enterprise Blueprints: A case for IT Blueprints. The 21st International Conference on Advanced Information Systems, Lecture Notes in Business Information Processing. Vol. 34, pp. 70-84. New York: Springer.

4. BATINI, Carol, NAVATHE, Shamkant B., CERI, Stefano (1991). Conceptual Database Design, An Entity-Relationship Approach. 1st Edition. pp. 56-62. Indianapolis: Addison-Wesley Professional.

5. DENHART, Robert B., DENHART, Janet V (2009). Public Administration: An Action Orientation: An Action Orientation. 7th Edition. pp. 1-3. Arizona: Cengage Learning.

6. SHARMA, Urmila, SHARMA, S.K. (2006). Public Administration. 1st Edition. pp. 7-8. India: Atlantic Publishers & Distributors.

7. LAND, Martin Op't, PROPER, Erik, WAAGE, Maarten, CLOO, Jeroen, Steghuis, Claudia. (2008). Enterprise Architecture: Creating Value by Informed Governance. 2nd Edition 2009. pp. 5. New York: Springer.

8. LANKHORST, Marc (2012). Enterprise Architecture at Work: Modeling, Communication and Analysis. 3rd Edition 2013. pp. 3-5. New York: Springer.

9. DAVIS, Paul K., BIGELOW, James H. (2003) . Motivated Metamodels: Synthesis of Cause-Effect Reasoning and Statistical Metamodeling. Arlington: RAND Corporation.

10. SMOLIK, Petr C. (2006). Mambo Metamodeling Environment. Library of the Faculty of Information Technology, Brno University of Technology, Czech Republic.

11. BATINI, Carol, NAVATHE, Shamkant B., CERI, Stefano (1991). "Conceptual Database Design, An Entity-Relationship Approach". 1st Edition. pp. 9-71. Indianapolis: Addison-Wesley Professional.

12. James, G. A. (2008). Magic Quadrant for Enterprise Architecture Tools. Gartner RAS Core Research Note G00156427.

13. BITTLER, R. Scott, http://www.gartner.com/technology/reprints.do?id=1-1CNYH8L&ct=121031&st=sb, Accessed in 21-07-2013.

14. ZACHMAN, J. A. (1987). A Framework for Information Systems Architectures. IBM Systems Journal. Vol. 26, Chapter 3.

15. SPEWAK, S. H (1992). Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology. 2nd Edition. New York: John Wiley & Sons.

16. STENBIT, John (2004). DoD Architectural Framework Version 1.0.

17. Department of the Treasury, Chief Information Officer Council (2000). Treasury Enterprise Architecture Framework. 1st Edition.

18. Chief Information Officer Council (1999). Federal Enterprise Architecture Framework. Version 1.1.

19. OPEN GROUP (2012). TOGAF Study Guide. 1st Edition. Chapter 1. Philadelphia: Van Haren Publications.

20. OPEN GROUP (2012). TOGAF Version 9. 1st Edition. Chapter 2. Philadelphia: Van Haren Publications.

21. OPEN GROUP (2012). Archimate 2.0 Specificaction. 1st Edition. Chapter 2. Zaltbomme: Van Haren Publications.

22. SOWA, J.F., ZACHMAN, J.A. (1992). Extending and Formalizing the Framework for Information Systems Architecture. Vol. 31, Chapter 3, pp. 590-616. IBM Systems Journal.

23. OPEN GROUP (2009). TOGAF Version 9. 1st Edition. Chapter 3. Philadelphia: Van Haren Publications.

24. KARL, Menzel and Mayer (1993). IDEF - http://www.idef.com/ . Accessed in 18-06-2013.

25. Event-driven process chain. http://www.ariscommunity.com/event-driven-process-chain. Accessed in 06-01-2013

26. SALOMON, G. (1996). Distributed Cognitions: Psychological and Educational Considerations (Learning in Doing - Social, Cognitive, and Computational Perspectives). Cambridge University Press.

27. VASCONCELOS, A. (2007). Arquitectura dos Sistemas de Informação: Representação e Avaliação. PhD Thesis. Lisbon: Instituto Superior Técnico.

28. MYERS, M. D (1997). Qualitative Research in Information Systems. Vol.21, Chapter 2, pp. 241- 242. University of Minnesota: MIS Quarterly.

29. BRYMAN, A. (1989). Research Methods and Organization Studies. United Kingdom: Routledge.

30. AMA, I.P. (2009). Arquitectura Informacional - Entidades e Relações. Version 0.9. Internal Documentation.

31. OPEN GROUP (2006). Architecture Framework. Enterprise Edition. Chapter 19-20. United Kingdom.

32. Diário da República (2010). PRESIDÊNCIA DO CONSELHO DE MINISTROS. 1.ª série - N.º 30 . Portaria n.º 92/2010.

33. ÖSTERLE, H., BECKER, J., FRANK, U., HESS, T., KARAGIANNIS, D., KRCMAR, H., et al. (2010). Memorandum on Design-Oriented Information Systems Research. Number 1, pp. 7-10. European Journal on Information Systems.

34. DREVER, Eric (1995). Using Semi-Structured Interviews in Small-Scale Research. A Teacher's Guide. Edinburgh: Scottish Council for Research in Education.

35. Diário da República (2012), MINISTÉRIOS DAS FINANÇAS E DA DEFESA NACIONAL. 1.ª série - N.º 65. Portaria n.º 86/2012 e n.º 87/2012.

36. METSKER, Steven John (2001). Building Parsers with Java. 3rd Edition. Chapter 1, pp.1-3. Boston: Addison-Wesley - Person Education.

37. GRUNE, Dick, JACOBS (2008). Parsing Techniques: A Practical Guide. 2nd Edition. Chapter 1, pp.1-2. New York: Springer.

38. LERMAR, Julia (2009). Programming Entity Framework. 2nd Edition. Chapter 1, pp.1. Sebastopol: O'Reilly Media, Inc..

39. GILTROW, Janet, GOODING, Rick, BURGOYNE, DANIEL & SAWATSKY, Marlene (2009). Academic Writing - An Introduction. 2nd Edition. Chapter 14 pp. 277-278. Broadview Press .

40. VAN DER VALT, T., COETSEE, A. D., & VOL SOLMS, S. H. (2013). Influence of International Best Practices on the South African Public Service's Corporate Governance of ICT. pp.1-4. ISACA Journal.

41. PEFFERS, Ken, TUUNANEN, Tuure, ROTHENBERG, Marcus, & CHATTERJEE, Samir (2008). A Design Science Research Methodology for Information Systems Research. Vol. 24, pp. 45-77. Journal of Management Information Systems.

42. HEVNER, Alan, RAM, Sudha, MARCH, Salvatore., & PARK, Jinsoo (2004). Design Science in Information Systems Research. Vol. 28, Number 1, pp. 75-105. University of Minnesota: MIS Quarterly.

9 Appendixes

9.1 Appendix I: Survey

Questionario:	Respostas
Breve descrição do que se trata o vosso trabalho	
Em que é que o processo que foi fornecido de Archi + Parsers facilitou na vossa tese e se deram isso a alguma entidade? Se sim, qual?	
Se encontraram dificuldades no uso da solução e se algo podia ser melhorado	
O que acham do metamodelo actual?	
Em comparação com o metamodelo que foi elaborado, o que acham que melhorou?	
Com duas ou três palavras descrevam o novo metamodelo.	

Tabel 2: Survey

9.2 Appendix II: Primitives Roadmap

Top-Down:

T2:

Create new entities.

Create new relation Aggregation between the entity and all new entities.

Make a query which will arrange all objects that will have the same properties as the new entities.

Tw:

Create new entity.

Make a query which will find all objects that have the reference to the old entity and make them point to the new entity.

Tx:

Deletes de entity given by argument

Ty:

Update the name of the entity with the new name both given by arguments.

Tz:

Make a query which selects all the targets entities.

Add a new Property with the name of the class and as a value the name of the object.

Bottom-Up:

B1:

Create a new entity given by argument.

B2:

Create a new relationship given by argument between two entities given also by arguments.

B3:

Create new entity.

Create new relation Aggregation between this new entity and all entities given by argument.

Make a query which will reference all the previously created entities to the new created entity.

B4:

Create a new attribute in an entity given by argument.

9.3 Appendix III: Primitives applied to the practical case AMA

Business Roles: (Roles)

B1: Roles

Business Actor: (Organização)

Ty: E16_Colaborador -> Empresas

Tw: E18_AreaOrganica -> Organismos ([Name] Nome; [Number] NrChefiasTIC; [Number] NrColaboradoresInternosTIC; [Number] NrColaboradoresExternosTIC; Observacoes)

Tx: E18_AreaOrganica

B3: Organização (Empresas; Organismos)

B2: Assigned to (Roles; Organização)

Contract: (Contratos)

Ty: E14_ContratoSWAplicacoes -> Contratos SW

Ty: E6_ContratoHW -> Contratos HW

Ty: E5_ContratoComunicacao -> Contratos Comunicação

Tz: Produto SW -> Contratos SW

Tz: E4_Licenca -> Contratos SW

B1: Contratos Serviços

B3: Contratos (Contratos Serviços; Contratos HW; Contratos SW; Contratos Comunicação)

Location: (Localização)

T2: Localização -> (Salas Técnicas, Sede)

Business Object: (Entidades Informacionais)

B1: Entidades Informacionais

Business Service: (Serviços de Negócio)

B1: Serviços IT

B1: Serviços Negócio

B3: Serviços de Negócio (Serviços IT; Serviços Negócio)

Business Process: (Processos de Negócio)

B1: Processos de Negócio

Data Object: (Dados)

B1: Dados

Ty: E15_ElementodeDespesa -> Elemento Despesa

B2: Aggregates (Dados; Elemento Despesa)

Application Service: (Serviços de Aplicações)

B1: Interface Utilizador

B1: Interoperabilidade Dados

B1: Rede

B1: Gestao Dados

B1: Rede Sistema

Tw: E23_FuncaoInterfaceUtilizador -> Interface Utilizador (P_DocumentosBrowser; P_SemanticaWeb; P_DocumentosBrowser; P_Script; P_AcessibilidadeInformaçãoConteúdo; P_AcessibilidadeServiçosOnline; P_AcessibilidadeInformaçãoConteúdo; P_AcessibilidadeServiçosOnline)

Tw: E20_FuncaoInteroperabilidadeDados -> Interoperabilidade Dados (P_ImagemRaster; P_ImagemVectorial; P_DescriçãoDocumentosDados; P_ConversãoXML; P_EstruturaDados; TransformaçãoApresentação;

P_CodificaçãoCaracteres; P_DocumentosEditáveis; P_DescriçãoDocumentosDados; P_SindicaçãoConteudos; P_TransferenciaSegura; P_Streaming; P_GeoCobertura; P_GeoFunções; P_GeoMapa; P_GeoProcessamento)

Tw: E22_FuncaoRede -> Rede (P_MensagensInstantâneas; P_CorreioAcessoLocal; P_FormatoMensagens; P_CorreioAcesso; P_CorreioEnvio; P_EnvioDados; P_SegurançaCorreioAcesso; P_SegurançaCorreioEnvio)

Tw: E21_FuncaoGestaoDados -> Gestao Dados (P_SGBD; P_DocumentosPartilhados)

Tw: E24_FuncaoRedeSistema -> Rede Sistema (P_AcessoDiretório)

Tx: E23_FuncaoInterfaceUtilizador

Tx: E20_FuncaoInteroperabilidadeDados

Tx: E22_FuncaoRede

Tx: E21_FuncaoGestaoDados

Tx: E24_FuncaoRedeSistema

B3: Serviços de Aplicações (Interface Utilizador; Interoperabilidade Dados; Rede; Gestao Dados; Rede Sistema)

<u>Application Component: (Soluções)</u>

Ty: E19_Integrador -> Integrador

Ty: E12_Aplicações -> Aplicações

B1: Componentes

B2: Aggregates (Componentes; Integrador)

B3: Soluções (Aplicações; Componentes)

B2: Assigned to (Soluções; Serviços de Aplicações)

B2: Associated (Soluções; Dados)

B1: Aplicação

Tw: Aplicação_Software -> Aplicação ();

B2: Aggregates (Aplicações; Aplicação)

Tx: Aplicação_Software

Tz: Fabricante -> Soluções

Artifact: (Artefactos)

Ty: E9_Backups -> Backups

Ty: E11_BD - > BD

B3: Artefactos (Backups; BD)

Tz: LinguagemProgramação -> Artefactos

System Software: (Software)

Ty: SGBD -> Plataformas

Ty: E17_ServiceDesk -> Service Desk

B1: Instancia BD

B1: Software Servidores

B1: Bases de Dados

B1: Software Sistemas

B3: Software (Software Servidores; Bases de Dados; Software Sistemas; Plataformas)

Tz: Fabricante -> Software

B2: Aggregates (Software Sistemas; Service Desk)

Tw: E10_InstanciaBD -> Instancia BD ([Name] Nome Intância)

B2: Aggregates (Bases de Dados; Instancia BD)

Communication Path: (Comunicações)

B1: Comunicações Voz Fixa

B1: Comunicações Fixas Dados

B1: Comunicações Móveis Voz

B1: Banda Larga Móvel

B1: Soluções M2M

B3: Comunicações (Comunicações Voz Fixa; Comunicações Fixas Dados; Comunicações Móveis Voz; Banda Larga Móvel; Soluções M2M)

Ty: Fabricante -> Comunicações

Infrastructure Service: (Serviços Tecnológicos)

B1: Data Interchange Services

B1: Data Management Services

B1: User Interface Services

B1: Graphics and Imaging Services

B1: International Operation Services

B1: Security Services

B1: Location and Directory Services

B1: Network Services

B1: System and Network Management Services

B1: Transaction Processing Services

B1: Software Engineering Services

B1: Operating System Services

B3: Serviços Tecnológicos (Data Interchange Services; Data Management Services; User Interface Services; Graphics and Imaging Services; International Operation Services; Security Services; Location and Directory Services; Network Services; System and Network Management Services; Transaction Processing Services; Software Engineering Services; Operating System Services)

Device: (Equipamento)

Ty: E1_Dispositivo -> Computador Desktop

Ty: E3_EquipamentoComunicacao -> Equipamentos Comunicações

Ty: E7_Storage -> Storage

Ty: E8_DriveStorage -> Drives Storages

B1: Servidores

B1: Portátil

B1: Backups

B1: Video Conferência

B1: Ministério da Defesa

B1: Ministério da Educação

B1: Ministério da Saúde

B3: IT Processamento (Computador Desktop; Servidores; Portátil)

B3: IT Storage (Drives Storages; Storage; Backups)

B3: IT Comunicações (Equipamentos Comunicações; Video Conferência)

B3: Equipamento (IT Processamento; IT Storage; IT Comunicações; Ministério da Defesa; Ministério da Educação; Ministério da Saúde)

Tz: Tipo Dispositivo -> Equipamento

Tz: Fabricante -> Equipamento

Tz: Proprietario -> Equipamento

Tx: FunçãoServidor

Tx: E2_LinkComunicação

Tx: E13_ParqueInformatico

Relationships:

B2: Associated (Serviços de Negócio; Entidades Informacionais)

B2: Associated (Serviços de Negócio; Processos de Negócio)

B2: Associated (Serviços de Negócio; Localização)

B2: Associated (Serviços de Negócio; Organização)

B2: Associated (Serviços de Negócio; Contratos)

B2: Associated (Entidades Informacionais; Organização)

B2: Associated (Entidades Informacionais; Soluções)

B2: Associated (Processos de Negócio; Soluções)

B2: Associated (Localização; Equipamento)

B2: Associated (Localização; Organização)

B2: Associated (Contratos; Organização)

B2: Associated (Contratos; Soluções)

B2: Associated (Contratos; Software)

B2: Associated (Contratos; Equipamento)

B2: Associated (Contratos; Comunicações)

B2: Associated (Organização; Artefactos)

B2: Associated (Organização; Comunicações)

B2: Associated (Organização; Equipamento)

B2: Associated (Organização; Software)

B2: Associated (Organização; Soluções)

B2: Associated (Soluções; Artefactos)

B2: Associated (Soluções; Software)

B2: Associated (Artefactos; Serviços Técnológicos)

B2: Associated (Software; Serviços Técnológicos)

B2: Associated (Software; Equipamento)

B2: Associated (Comunicações; Equipamento)

Attributes:

B4: Serviços de Negócio (Ref; [Name] Nome; Descrição; Tipo; [eDay] Data Nascimento; [eDay] Data Morte)

B4: Serviços IT ([URL] Link para Respostas Livres);

B4: Localização (Descrição; Morada; Código Postal; País; Localidade; Concelho; [eDay] Data Nascimento; [eDay] Data Morte)

B4: Salas Técnicas ([Boolean] Sala Dedicada a equipamentos?; [Boolean] Sala partilhada com outros organismos?; [Boolean] Existe Controlo de Acessos?; [Boolean] Possui Janelas?; [Boolean] Tem video vigilancia na sala de equipamentos?; [Boolean] Tem detecção de Incêndios automatica?; [Boolean] Tem Extinção Automática de Incêndios?; Manual/Automatico?; Qual o liquido / gás?; [Boolean] Tem Sistemas de arrefecimento dedicado?; Redundantes?; [Boolean] Possui Chão Falso?; [Boolean] Arrefecimento pelo Chão?; Altura do Chão Falso? (cms); Qual a Temperatura na sala? (ºC); Qual o Indice de Humidade da Sala? (%HR); [Boolean] Possui UPS?; Capacidade max. (kW); Load/Carga (%); [Boolean] Possui Gerador?; Capacidade max. (kW); Load/Carga (%); [Boolean] Existe quadro de eletrico dedicado a DC?; Redundante?; Área sala técnica (m2); [Number] Número de Bastidores Instalados na Sala; [Number] Capacidade Máxima de Bastidores na Sala; [Number] Nº de Servidores não Rack Mountable; Condições Humidade; Degradação Fisica da Sala; Confusão de Cabos; [Boolean] Possui RH dedicados?; Quantos RH?; Renda / Aluguer do Espaço; RH dedicados à Gest./Oper. do DC; Manutenção UPS; Manutenção Cooling; Manutenção Grupo Gerador; Manutenção Distribuição de Energia; Manutenção Sala / Espaço Físico; Notas)

B4: Entidades Informacionais ([Name] Nome; [eDay] Data Nascimento; [eDay] Data Morte)

B4: Processos de Negócio ([Name] Nome; [eDay] Data Nascimento; [eDay] Data Morte)

B4: Organização ([Number] NºContribuinte; Descrição; [Name] Nome; [eDay] Data Nascimento; [eDay] Data Morte)