A Tool for Co-evolution of Enterprise Architecture Meta-model and Model

Tiago Mendes de Almeida Rechau

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Miguel Leitão Bignolas Mira da Silva
Prof. Pedro Manuel Moreira Vaz Antunes de Sousa

Examination Committee

Chairperson: Prof. João António Madeiras Pereira
Supervisor: Prof. Miguel Leitão Bignolas Mira da Silva
Members of the Committee: Prof. Sérgio Luís Proença Duarte Guerreiro

October 2017

Acknowledgments

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

I would also like to acknowledge my dissertation supervisors Prof. Miguel Mira da Silva, Prof. Pedro Sousa, and also to Ph.D. student Nuno Silva for their insight, support, and sharing of knowledge that has made this Thesis possible.

Last but not least, to all my friends and colleagues that helped me grow as a person and were always there for me through thick and thin. Thank you.

Abstract

Enterprise Architecture (EA) provides means to align the organization's business and IT. Although several researchers have extensively elaborated on EA, the transformation phase has received little attention, in particular migrating from a baseline architecture to a target architecture. An important aspect of transforming an EA is the co-evolution of the EA meta-model and respective model. Existing EAM (Enterprise Architecture Management) tools support the process of EA by enabling the modeling and management of EA models. Surveys on existing EAM tools showed that there is lack of support for the co-evolution process of both EA meta-model and model. This research intends to address this problem by proposing an integrated system for co-evolving the EA meta-model and model. The system was evaluated, using interviews, focus groups and user tests, all of them providing positive evaluation.

Keywords

Enterprise Architecture; EAM Tool; EA Meta-model; EA Model; Co-evolution

Resumo

A Arquitetura Empresarial (AE) fornece meios para alinhar o negócio e as tecnologias de informação de uma organização. Embora vários investigadores tenham investigado extensivamente sobre AE, a fase de transformação recebeu pouca atenção, em particular a migração de uma arquitetura base para uma arquitetura futura. Um aspecto importante da transformação de uma AE é a co-evolução do meta-modelo AE e do respectivo modelo. As ferramentas de gestão de AE existentes suportam o processo de AE, permitindo a modelação e gestão de modelos AE. Investigações sobre ferramentas gestão de AE existentes mostraram a existência de falta de suporte das ferramentas para o processo de co-evolução do modelo e do modelo EA. Esta pesquisa pretende abordar este problema, propondo um sistema integrado para a co-evolução de EA, tanto meta-modelo como modelo. O sistem foi avaliado por testes de utilizador, entrevistas e *focus groups*, todos eles gerando resultados positivos.

Palavras Chave

Arquitetura Empresarial; Ferramente de gestão de Arquitetura Empresarial; Meta-modelo de Arquitetura Empresarial; Modelo de Arquitetura Empresarial; Co-evolução;

Contents

List of Figures

List of Tables

Acronyms

ACM Architecture Content Framework

ADM Architecture Development Cycle

API Application Programming Interface

DSR Design Science Research

EAF Enterprise Architecture Frameworks

EAMS Enterprise Architecture Management System

EAM Enterprise Architecture Management

EA Enterprise Architecture

JS JavaScript

JSON JavaScript Object Notation

MOF Meta Object Facility

REST Representational State Transfer

TOGAF The Open Group Architecture Framework

UI User Interface

1

Introduction

Contents

Organizations, whether they operate in public or private sectors, have to deal with the constantly changing environment driven by various factors, such as technology advancement, demanding customers, aggressive competitors as well as regulatory changes. In order to remain competitive, these organizations need to adapt in order to swiftly change their business strategy and/or business processes. To do so, organizations need to have a holistic view of their structure concerning the impact of changes in their environment.

EA is used as a means to design, communicate, and implement the desired organizational changes according to the business strategy, processes, and systems of the organization's business and IT domains. EA captures the essentials of the business and IT infrastructure by consolidating architecture decisions, along with migration and implementation plans to address strategic aspects of the enterprise. Fundamentally, EA describes 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. Once organizations have a clear picture of their EA, they can manage their assets better and plan the necessary changes according to their strategy more efficiently [10] .

The evolution of EA is the result of applying EA projects within organizations with the goal of accomplishing specific business requirements. Recent approaches seek to automate and improve EA practice across organizations by employing EAM tools. Hence, evolving the organization's EA meta-model is a consequence of fulfilling such initiatives.

Due to the complexity of an EA project, the idea of starting with a simplified meta-model and enriching it while growing within the organization is often preferred. However, addressing these projects implies that EA meta-models are supported as living artifacts that evolve according to the organization's needs [11].

Occasionally, the meta-models and concepts used to model the baseline model (AS-IS) are no longer enough to model the current AS-IS or even the current target (TO-BE) model. For example, an organization wants to reformulate its business model by changing, deleting and creating new business concepts. Additionally, the current information systems architecture that supports the AS-IS and the TO-BE business cannot be adapted in any way. Thus, the organization's EA needs to be re-designed business-wise as well as the information systems architecture supporting the TO-BE business. Migrating all these changes can lead to many issues, such as, uncoordinated, *ad-hoc* decisions concerning EA components and their interrelationships resulting in the duplication of efforts and resources, poor coordination and control, management and business performance issues, and inefficiencies in operation [12].

To facilitate the task of meta-model and model co-evolution, this research proposes a tool for the co-evolution process of both meta-model and model. The tool uses the core features of the Enterprise Architecture Management System (EAMS) [1] and expands them to allow the evolution of both EA meta-

[1] http://www.linkconsulting.com/eams

model and model in a visual manner. The tool was created based on existing literature and two rounds of interviews with seven practitioners.

This document is structured as follows. First, the document starts with an introduction to the theoretical background and research methodology in Section 1.1 and Section 1.2, respectively. This is followed by the contextualized research problem and the motivation for this research in Chapter 2. Chapter 3 presents the state-of-the-art regarding the scope of this research. The tool is explained in Chapter 4, followed by its demonstration in practice (Chapter 5). The evaluation of the proposed solution, the chosen validation methods, and criteria are described in Chapter 6. Finally, in Chapter 7 the document concludes with a research overview, discussion, limitations found concerning the proposed solution, and how to address these and other topics in future work.

1.1 Theoretical Background

Basic Concepts

In order to understand the following this research 3 main concepts should be understood:

Model is understood as the representation of systems via concepts and relations between them, allowing to understand difficult ones. In EA, models allow the representation of enterprise systems and structure via pre-defined concepts and relations. Model is an instance of a meta-model [5, 13].

Meta-model is understood as the concepts and relations between them allowed to be used in models, i.e., meta-model is the language used to create models [13].

Co-evolution is the activity of evolute two artifacts in same time span, i.e., a synchronized evolution. This research uses this concept where the two artifacts used are the meta-model and model, and finally, the time span in use is the time span of a project where both have an evolution [5, 13].

Enterprise Architecture

With the increased complexity of enterprises, their systems, and business it is necessary an architecture to manage all the complexity [10].

Architecture as in [14] is defined as the fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution. Using this definition at the level of an organization we have EA that is defined as a whole of principles, methods, and models used in the design and realization of an enterprise's organizational structure, business

processes, information systems, and infrastructure [15]. The aim of EA is develop an IT strategy and aligning business strategies implementations [16] by capturing the essentials of IT, business, and their evolution [10], providing a holistic view of the enterprise [10]. The holistic view of an enterprise business, organization, and IT structure allows EA to be a communication support in sharing the information with stakeholders. Using EA as a management instrument helps enterprises determine their needs and priorities from the business perspective, as well as assess how the company may benefit from technological and business innovations [10].

Enterprise Architecture Frameworks

To communicate with all stakeholders EA requires the employment of techniques in order to describe architectures in a coherent manner. Moreover, EA captures the essentials of an enterprise to provide stability to the architecture, the environment changes, new technological opportunities, and new insights as to what is essential to the business to prosper. As a result, EA changes and therefore needs to be managed properly.

Enterprise Architecture Frameworks (EAF) describe a method for designing EA in terms of a set of building blocks in which they provide generic concepts and common blocks and how these blocks fit together. More specifically EAF provide a list of recommended standards to manage EA, simplifying the development of the architectures and ensuring a common language for communication of all stakeholders of an EA.

Originally published in 1995 The Open Group Architecture Framework (TOGAF) is known as a global standard for EA. TOGAF is composed by a detailed method and a set of supporting tools for developing an enterprise architecture. It may be used freely by any organization wishing to develop an enterprise architecture for use within that organization [1]. TOGAF is structured as follows:

- An Architecture Capability Framework that provides reference materials for establishing an architecture which addresses the organization, processes, skills, roles, and responsibilities.
- The ADM (Figure1.1), describes a method for architects to use when developing and managing EA's lifecycle. The ADM is known as the core of TOGAF and consists of a step-by-step cyclic approach for the development of the overall EA.
- The Architecture Content Framework (ACM), allow TOGAF to be used as a standalone framework, describing how an EA should look like. Should be used in parallel with ADM.
- The Enterprise Continuum provides the architect with methods for classifying architecture and solution artifacts, being important for communication and understanding of EA.

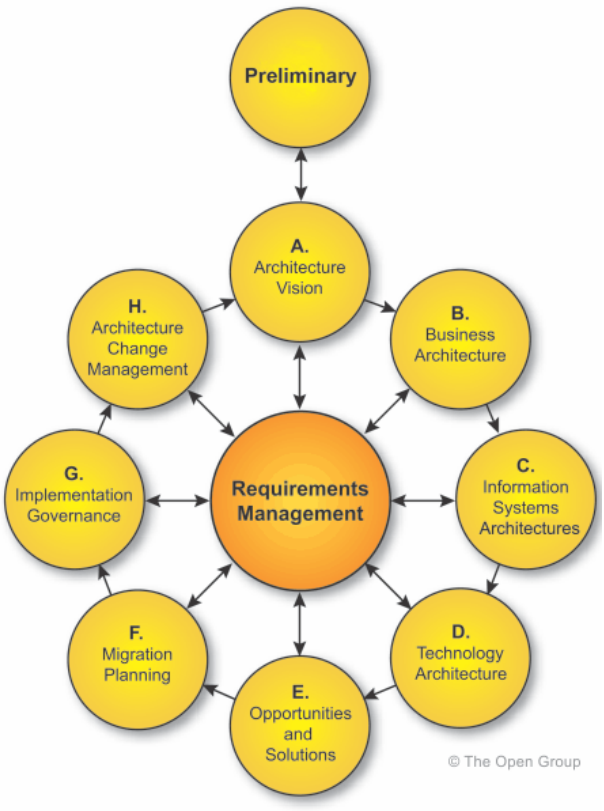

Figure 1.1: TOGAF ADM (from [1])

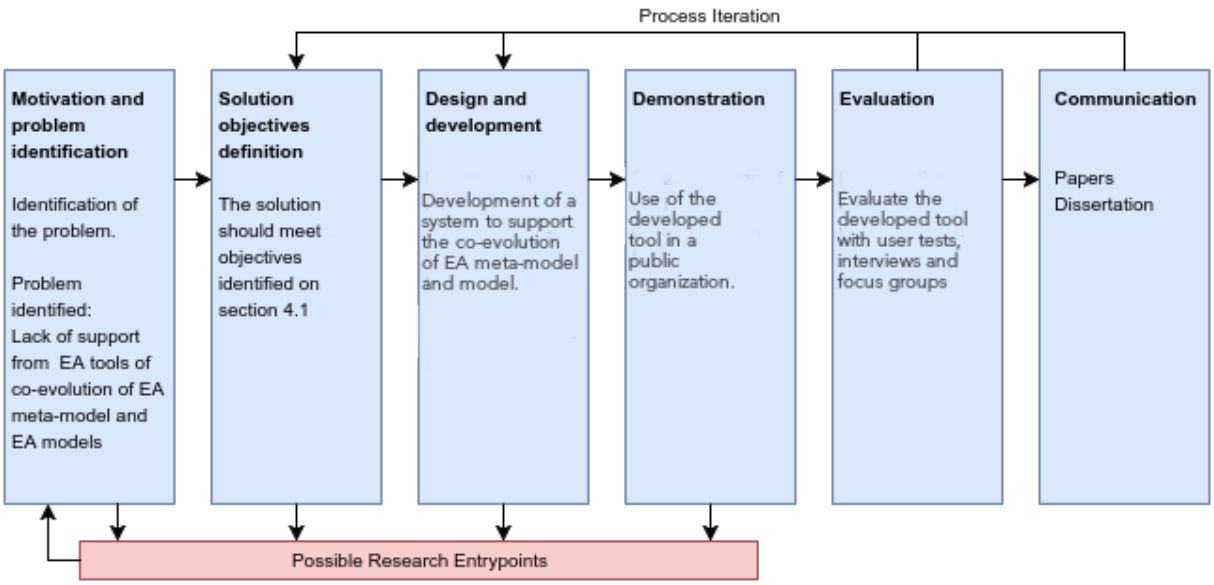

Figure 1.2: Mapping of Research Activities with DSRM. Adapted from [2]

1.2 Research Methodology

Throughout the conducted research, the DSR method proposed by [17] was considered suitable for addressing the research problem. Design science is inherently a problem-solving process based on the understanding and formulation of a design problem. Then, through design and implementation, a solution artifact is built based on the research objectives. Finally, the artifact is assessed against the defined objectives by applying it to instantiations of the problem [17].

The artifacts might include constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices) and instantiations (implemented and prototype systems). This iterative method includes the following phases: *Problem Identification*, *Objectives Definition*, *Design and Development*, *Demonstration*, *Evaluation* and *Communication* [2].

This methodology is appropriate to research that seek to extend the boundaries of human and organizational capabilities, by creating new and innovative artifacts. Also, DSRM is active with respect to technology, engaging in the creation of technological artifacts that impact people and organizations [17].

This methodology is adequate for this research since it aligns pro-activity on the solution design with organizational context. This alignment is essential to this research, as the solution requires theoretical foundation as well as organizational acceptance. The mapping of this research work is on DSR is presented on Figure 1.2.

2

Research Problem

The EAM process raises a set of challenges and issues that compromise its benefits [18]. Insufficient management commitment, lack of governance structures in EA projects, poor cross-organization coordination, and communication, lack of shared vision and vocabulary that causes decision-makers to operate with imperfect knowledge and understanding are some of the problems that make justifying for EA efforts a hard task.

Concerning the management of EA transformation, Buckl et al. states that many of today's enterprises face problems in managing the transformation from a baseline architecture to a target architecture via intermediary planned architectures [19]. Also, Aier et al. identified several causes regarding this issue, such as missing practical methodologies for architecture road-mapping, inadequate representation of the concept of time in architectural models and insufficient tool support for architecture planning [20].

One of the problems stated is the **lack of support from EAM tools for co-evolution process of the EA meta-model and conforming model**, that is problem of this research and justified in the next paragraphs.

More than just providing enterprise data visualizations, EAM tools are used when making strategic decisions concerning the enterprise [21]. Although recent approaches tried to automate and improve EA practice within organizations using EAM tools, there is still an insufficient tool support [12, 16] for tracking and maintaining a diverse collection of entities, such as strategic goals, objectives, stakeholders, business process descriptions, applications, data and so on [12].

Additionally, from the multitude of available tools (Troux, Aris, Casewise Corporate Modeler, Enterprise Architect, System Architect, ABACUS, etc.), there is lack of support for the co-evolution process of EA meta-models while maintaining a well-integrated EA repository containing the past, present, and future states of the architecture [4, 22, 23].

EAMS is an EAM tool that addresses part of the EA evolution problem by providing a life-cycle feature to all organizational elements that are part of the enterprise cartography [11]. However, even EAMS current state does not provide the means for handling the co-evolution process of EA meta-models and models from an existing EA repository.

An extensive analysis of EAM tools performed in the form of a survey by Matthes et al. [22, 23] corroborates the above-mentioned statements. The surveys were conducted in cooperation with 30 industry partners and analyzed the EAM tools produced by nine major players in the market. The survey pursued a threefold evaluation approach detailed in Chapter 3.

Furthermore, a tool for managing EA meta-model evolution was developed [24] but still required considerable manual work. Also, an automated analysis of the co-evolution of EA meta-models and EA models that support enterprise architects in decision-making throughout the EA process is still missing.

All things considered, although general guidelines and insights of performing the transformation process have been provided by EA frameworks, such as TOGAF [1], the implementation support by EAM

tools is still limited. All in all, EAM tools do lack to provide the necessary co-evolution features and impact indicators for managing co-evolution of EA meta-models and EA models throughout the EA transformation process. This issue compromises the success and impairs the benefits of the overall EA process.

3

Related Work

Contents

This section introduces a literature review regarding the topic of EA evolution. The author begins by presenting the topic of EA transformation, followed by an EAM tools analysis complemented with an analysis of two approaches to EA model change visualizations. To conclude, contributions to co-evolution of meta-models and models are presented.

3.1 Enterprise Transformation

This section provides, based on literature, general understanding, and definitions of key concepts regarding EA transformation. Enterprise transformation is known in the enterprise engineering community as a set of initiatives that change the organization's domain (its structure and dynamics) from its current AS-IS state to a predetermined TO-BE state [11, 25].

The current state of the EA referred as baseline architecture, is the initial state or the AS-IS of the EA [1]. A baseline architecture can be defined as the set of products that portray the existing enterprise, the current business practices, and the technical infrastructure. Based on the strategy and business goals, some development and incremental processes need to take place to reach the TO-BE state of the EA, also referred as the target architecture. Hence, target architecture can be defined as the set of models that portray the future or end-state enterprise, generally captured in the organization's strategic thinking and plans.

Evolving an EA is an iterative and incremental process [1, 26]. The transition architecture is a state of the EA between the baseline and target architectures. An enterprise might experience more than one transition architecture during its EA evolution. A road-map is an abstract plan for changes in business and/or technology, typically operating across multiple disciplines and over various years [1]. Architecture road-maps are used to describe the transformation path, over a certain period of time, from the baseline architecture to the desired target architecture. Thus, a timeline view is necessary for describing or visualizing the architecture road-map, hence showing the required activities needed to be performed to realize the target architecture.

A viewpoint and a conceptual model for road-mapping the development of EA over time were proposed by Buckl [19] from a pattern-based approach for EA management [27], meaning that both artifacts can be integrated into a multitude of organization-specific EAM approaches.

EAFs also provide some guidance on the development of EA over time. Zachman Framework [28] mentioned that all his view must be complete in order to have a successful EA. In terms of evolution, Zachman proposes two dimensions: Time and Motivation. The motivation refers the necessary views to show the motivation of changes, the ends, and means for change. Time refers to when do things happen relative to one another (the life cycles, the timing diagrams) [29].

TOGAF [1] provides a best-practice framework for adding value and enables organizations to build

workable and economic solutions addressing the business issues and needs. Regarding EA transformation, TOGAF outlines two main concept outputs: architecture road-map, and implementation and migration plan. Architecture road-map is the list of individual work packages in a timeline that will realize the target architecture. Implementation and migration plan is defined as schedules of projects that will realize the target architecture. Both definitions show that activities, either in the form of projects or work packages, will guide the realization of the target architecture.

TOGAF's ADM addresses EA transformation mostly with Phase E and Phase F. Phase E of TOGAF ADM, Opportunities and Solutions, describes the process of identifying the delivery vehicles, such as projects, programs, or portfolios, that will effectively deliver the target architecture. Since phase E focuses on how to deliver the architecture, it considers the complete set of gaps between the baseline and target architectures in the enterprise's multiple domains: business, data, application, and technology. Phase E is an effort to build a best-fit road-map based upon the stakeholder requirements, the enterprise's business transformation readiness, identified opportunities and solutions and identified implementation constraints.

Phase F, Migration Planning, shows how to move from the baseline to the target architecture by finalizing a detailed implementation and migration plan, including the architecture road-map. The objectives of migration planning phase are to finalize the architecture road-map; to ensure the coordination between implementation and migration plan with the enterprise's approach to manage and implement change in change portfolio, and to ensure that key stakeholders understand the business values and costs of work packages and transition architectures.

Another study identified the challenges of documenting EAs as being a manual, time-consuming and costly task [30]. A vendor-specific enterprise service bus was reverse-engineered to understand the degree of coverage to which data of a productive system could be used for EA documentation. From there, a set of model transformations for automating EA documentation was derived.

Finally, a set of principles were proposed with respect to the evolution of architectural views as a mean to address changes within the organization's domain [11, 25]. Keeping up-to-date architectural views with organizational changes have brought up issues in EA development, as well as some misalignment between both business and IT spectrum. To address this, instead of using a version schema that handles the evolution of architectural views, Sousa et al. proposed that all organizational artifacts have four fundamental invariant states (gestating, alive, dead, and retired) in their life-cycle as if each organizational artifact was a living entity [25].

In their approach [25], an enterprise is defined as a graph G of artifacts in which their relationships can be expressed and altered by two fundamental types of the enterprise type space:

- **Blueprint:** "whose instances contain references to other artifacts. A given *artifact* is represented on a given blueprint if graph G holds as a relation between them"; and

- **Project:** ”whose instances contain references to artifacts related to the project”.

A `state_of_existence` regarding all artifacts is also defined for all artifacts representing the EA other than *Blueprint* as one of the following states:

- **Conceived:** ”if it is only related with *blueprints*”;
- **Gestation:** ”if it is related with alive *projects* and is not related with any other artifact other than *blueprints*”;
- **Alive:** ”if it is related with other artifacts in the alive state. This means that it may act upon other artifacts in conceived, gestation or alive states”;
- **Dead:** ”if it is no longer in the alive state”.

Sousa et al. considered the IT project and life-cycle to be two important concepts concerning the evolution of EA. Nonetheless, their approach lacks a more detailed rationale on how the life-cycle of EA model elements changes, i.e., which transformations were applied to the elements.

3.2 EAM Tools

EAM handles a multitude of dependencies between the various domains of EA, making the EA process difficult without the assistance of EAM tools. When architectural requirements arise, EAM tools have to evolve and add new features, with various approaches to solve the same problems, each approach with its individual strengths and weaknesses.

To understand the state of EAM tools in supporting EA management, [22], [23] and [3] performed analysis to a total of 26 EAM tools. The following sections will resume all three surveys, divided by EAM support and visualization analysis.

3.2.1 EAM tools EAM support analysis

An analysis of 13 EAM tools was made by [22, 23] (9 tools by [22] and 4 by [23]) in which various aspects of EAM were considered in order to understand the state of different tools on the market. The analysis consisted of 2 different phases. On the first phase, 10 scenarios were created, each one to test different functionality. On the second phase, the focus was on the analysis of different EAM tasks, in which 10 scenarios were also created.

The analysis concluded that EAM tools present some limitations regarding EAM tasks. For example, no tool supports the concept of project delay, lack of concepts for life-cycle aspects of the infrastructure,

and lack of support for SOA management. Another limitation regarding EAM was a limited generation of visualizations that despite not being an EAM issue, allows a better EA overview.

As for co-evolution of EA meta-model and EA models, from the 13 analyzed tools, 10 supported features to change or adapt the information model, but only some allow the user to change defined elements, attributes, etc., maintaining their predefined information model. The analysis showed that 4 tools scored above 5 (of a maximum of 7), incorporating a complete solution to manage the information model. Although most EA tools already show concerns about the information model flexibility, there is lack of support of the tools allow the user to automatically migrate the EA models affected by changes in the information model. Regarding this theme, the survey does not mention the possibility of an impact analysis concerning any of the changes.

Focusing on co-evolution, and complementing with the information from [3] the following problems arise:

1. Inflexible Information Models;
2. Absence of a feature to perform gap analysis, to govern road-maps and trace the impact to the business architecture;
3. Difficulty or impossibility in propagating changes made through visual interactions to real data.

A final point is the lack of simulations' possibility, on EAM tools, complemented with methods for quantifying certain properties. These simulations will allow better governance of the EA dynamics.

3.2.2 EAM tools Visualization analysis

A visualization analysis was made to 18 EA tools by Roth et al. [3], with the goal of understanding which visualization configuration adaptation capabilities the tools have, and which of these are the most used by practitioners.

The analysis was divided into 2 phases. On the first phase, 50 EA vendors of different EA tools were asked to answer a survey about the capabilities that their EA tools dispose to users. On the second phase, with the goal of understanding current EA practices, market demands, trends and pain points, 109 practitioners described, via a survey, which EA tool features they actually use or plan to use.

All the visualizations received from EA vendors in the first phase were resumed to 26 (see Figure 3.1), the 5 most used by EA practitioners are Matrix, Cluster Map, Timeline, Flow Diagram, List.

Besides the usage of visualizations, some of them are becoming a trend in the market and are planned to be used by practitioners. Dashboard visualization which more practitioners intend to use is just one example (see Figure 3.2).

Although there are a lot of different types of visualization, some problems were encountered:

Figure 3.1: 26 visualizations provided by EA tools (from [3])

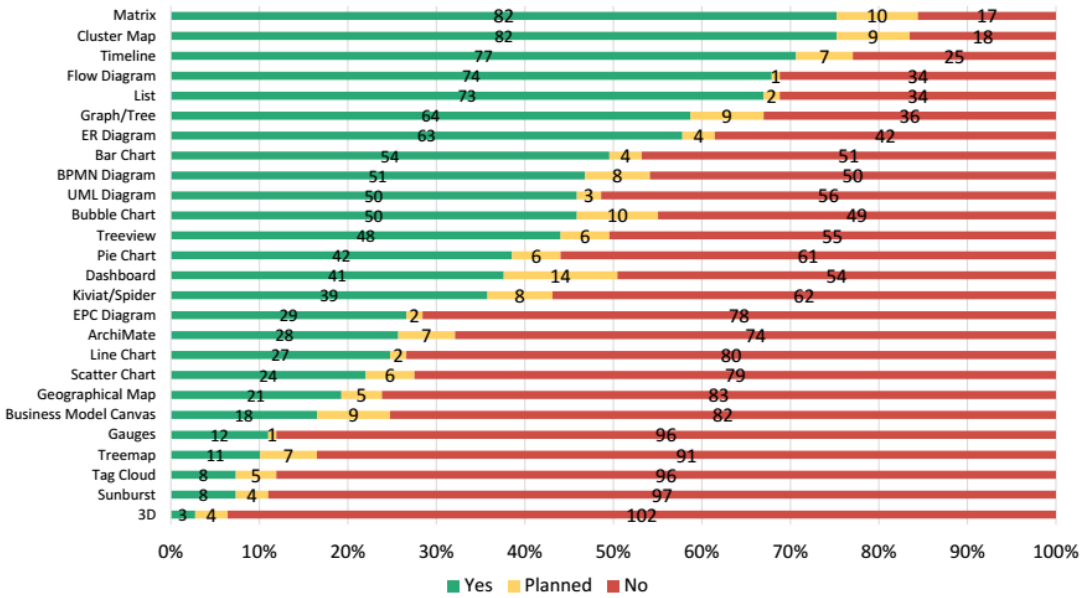

Figure 3.2: Usage of visualization types by the EA experts(from [3])

- Low interaction with visualizations, making them just a document of EA instead of making visualizations EA live-data, propagating the changes to the repositories;
- Lack of visual impact of EA transformations;
- EA tools are unable to provide a holistic picture;
- The nonexistence of historical visualization in some tools.

Regarding the theme of EA meta-models, 43% of the participants use a predefined schema (EA meta-model) which comes with their EA tool. In contrast, 57% of the participants use an organization EA meta-model. Most vendors confirmed that their tool supports an adaptation of the EA meta-model, but most of them did not support graphic visualization of the meta-model and in-place interaction, thus becoming a necessity to EA practitioners.

EAMS is another tool presented in the market, mainly in the Portuguese one. This tool presents various features as support for EA evolution by proving life-cycle management to all modeled artifacts, edition, and creation of an EA meta-model and visualization of the various states of the EA models, from the past to the present. Besides all the features presented on this EA tool, it still does not support co-evolution of EA meta-models and models.

A master thesis [24] was developed based on the EAMS tool. This thesis solves mainly the problem of the edition of EA meta-model and after the edition creates a file with a list of migration rules to apply to the EA models in order to complete the co-evolution. The main fault of the thesis is the lack of impact analysis on the evolution and the necessity of manual work to make complete the co-evolution.

3.3 EA Model's Changes Visualization

Enterprise architecture is periodically changing and those changes are usually implemented by several teams. Communication of the teams implementing changes requires the visualization of those changes. Visualization of changes in enterprise architecture languages is not a simple task as such languages reflect the static structure of an enterprise. For this purpose, two contributions were analyzed and discussed in the following sections.

3.3.1 Visualizing Differences of Enterprise Architecture Models by Roth, Sascha and Matthes, Florian

In Visualizing Differences of Enterprise Architecture Models [4], the proposed visualization is based on 4 layers (Figure 3.3). In the first layer, changes to the meta-model are shown in a graph where:

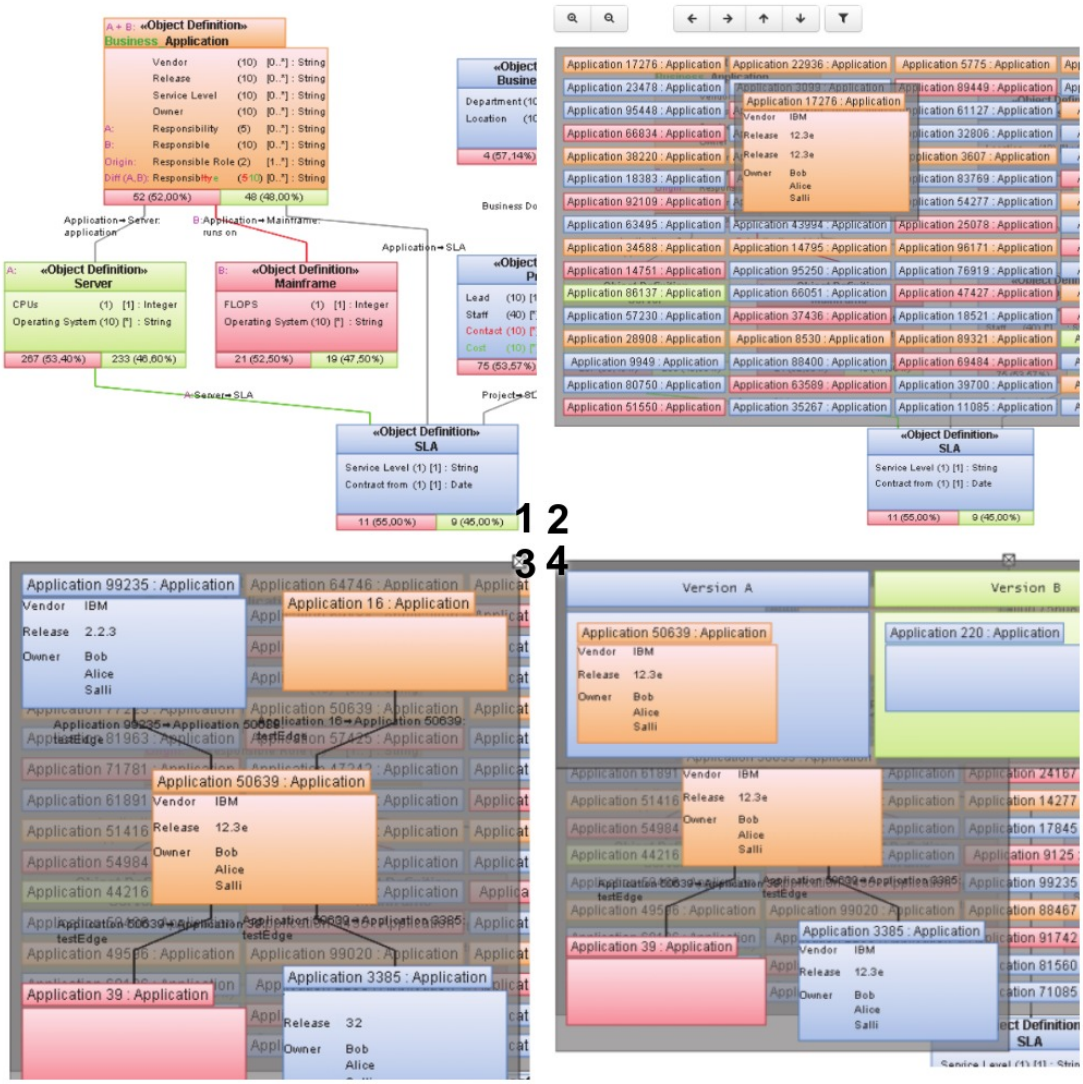

Figure 3.3: 4 layers (1-First Layer, 2-Second Layer, 3-Third Layer, 4-Fourth Layer) from [4]

- New Object Definitions (Object Definitions are treated as EA classes in this document) are shown in green;
- Altered Object Definitions are shown in orange;
- Deleted Object Definitions are shown in red;

The changes on names of Object Definitions tint the differences on the name with green to the added parts, and red to the deleted parts. Changes in relations between Object Definitions are also shown and follow the same logic, green to new relations, red to deleted relations and change ones with orange.

On the second layer an overview of objects (meta-model instances) following the previous logic is shown. New instances in green, changed ones in orange and deleted ones in red. Besides the

visualization is possible to filter, zoom in/out to facilitate the visualization. The third layer, shows the instance neighborhood, i.e., the neighbors of a chosen object, but focusing on relations of Objects instead of on attributes, because the expert states that Links (instances of relationships) between objects are far more interesting for an analysis than changes of attributes [4]. The color code is the same as on other layers. On the fourth and last layer, the user can choose an object and see the different versions of an object, comparing the different versions with the original one.

Analyzing the proposed visualization, some problems could be observed on this approach:

- The second layer became very confusing and with a lot of information in small space, as it is possible to see in Figure 3.3.

- If there exists, is imperceptible to the user the notion of time and historical data regarding the changes made to the models.

3.3.2 Visualization of changes in ArchiMate by Bakelaar, Robert and Roubtsova, Ella

Bakellar and Roubtsova [31] discuss an approach to show the visualization of changes within the Application and Technology Layers of ArchiMate 2.1 [32], focusing on the gap of AS-IS and TO-BE models.

In this approach, there is only one layer where all the changes should be visualized. The possible visualized changes are the following:

- Create, change or remove objects;

- Create, change or remove relations;

A color code was applied to observe changes, as well as labels. To new objects/relations the color is green and the label $<new>$, for changed ones orange and $<changed>$ and finally for relations or objects remove the color is gray and the label $<obsolete>$. To make the visualization scalable unchanged relations could be omitted. The omitted relations are the ones that are generic elements that appear on the model just to make the model understandable, so they are not really necessary.

When analyzing the approach, some problems emerge. Visualization of past states is not allowed, thus making impossible to see changed attributes and the possibility of the visualization becomes confuse when the users start to use it with more domains of ArchiMate.

3.4 Meta-model evolution & Model co-evolution

In Model-Driven Engineering (MDE), meta-models create a semantic and symbolic manner that has to be respected when representing the models. Often models become non-conformant with the evolved version of the meta-model, hence having to adapt accordingly [13]. Some changes in meta-model are additive changes, and this way, they don't break the respective models, but others introduce incompatibilities and cross-version inconsistencies, therefore invalidating the models.

Some authors have contributed to meta-model evolution and model co-evolution. In the next subsections, some of those contributions are described.

3.4.1 Metamodel adaptation and model co-adaptation by Wachsmuth, Guido

Wachsmuth, Guido [7] proposes a stepwise meta-model adaptation. Being a stepwise adaptation, each transformation applied to the meta-model has a corresponding co-transformation. Thereby, the [7] provides instant co-adaptation of models, allowing at each step that models conform to the actual meta-model version.

The transformations that are applied to meta-models are divided into 3 groups, Refactoring, Construction and Destruction (see Table 3.1).

Table 3.1: Proposed Transformations by [7]

Group	Transformation
Refactoring	Rename Element Move Property Extract/Inline Class
Construction	Introduce Class Introduce Property Generalize Property Pull Property Extract Superclass
Destruction	Remove Class Remove Property Restrict Property Push Property Flatten Hierarchy

The proposed transformations (see Table 3.1) are applied to meta-models and have co-transformations patterns to co-evolute the respective models. The co-transformations were divided in Co-Refactoring, Co-Construction, and Co-Destruction according to the transformation applied to the meta-model.

In the paper are shown some examples of how co-transformations should work, but there isn't any formal description on how to apply these transformations and co-transformations.

3.4.2 Automating the migration of enterprise architecture models by Silva, Nuno

The approach of Silva, Nuno [8] is based on [7] migration rules, already applied to EA models and meta-models. [8] proposes the migration rules shown in Table 3.2.

Table 3.2: Migration rules proposed by [8]

Group	Migration Rule
Refactoring	Change EA Object Class Change EA Property Type Move EA Property Inline EA Class Association to EA Class EA Class to Association
Construction	Introduce EA Property
Destruction	Eliminate EA Class Eliminate EA Property

Every migration rule corresponds to a modification on EA meta-model that needs to be propagated to the respective models. To do this propagation, every transformation has some properties, parameters, pre-conditions, post-conditions, and statements. The parameters are all the information needed to propagate the transformation. The pre-conditions are conditions that need to be met before the propagation is made. The post-conditions are the conditions that need to be met to guarantee that the propagation was well done. The statements are the real propagation, i.e., the algorithm to make the propagation.

The migration rules, when applied, follow the principle of Enterprise Transformation that *All organizational artifacts can be classified as being in one of four invariants – Gestating, Alive, Dead, and Retired.* [11]. That means that when applying any co-destruction transformation no artifacts are removed, instead, their state changes to *dead* or *retired*. This principle allows the saving of changes made to meta-models and its models, i.e., having a trace of changes.

3.4.3 Towards synchronizing models with evolving metamodels by Gruschko, Boris

Gruschko, Boris [5] presents an approach to address the model migration problem, implementing some transformations to migrate models when meta-models are changed. This approach is based on Meta Object Facility (MOF) meta-modeling architecture. MOF introduce a four-layer division, where each layer represents a different level of abstraction. The first layer, M3, represents the meta-meta-models, that are used to define M2, the second layer. M2 describes instances' models that are instances of M3. M2 models represent meta-model. The third layer, M1, describes instances of M2 and finally, M0 represents instances of M1.

The focus of this approach is the migration of M1 models when M2 changes. For that, 5 steps are

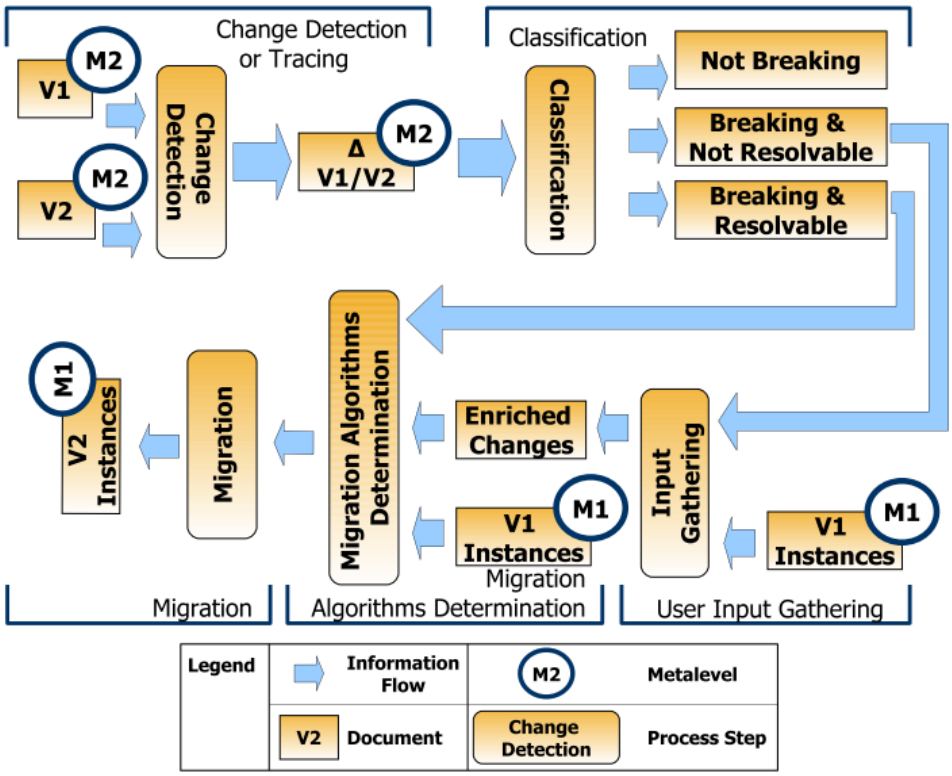

Figure 3.4: Example of M1 migration when changes in M2 happens (from [5])

needed (see Figure 3.4):

1. Changes detection or tracing;
2. Classification;
3. User input gathering;
4. Algorithms determination;
5. Migration.

On the first step exists two options. The first option is the direct comparison between two different versions of M2 models, where the output will be the differences between them. On the other hand, the second option is tracing the differences, i.e., take an older version of the M2 model and a trace of change operations as input. These trace of change operations represent the operations that allow the older version of an M2 model to get to the desired M2 model version. The different options have advantages and disadvantages, see Table 3.3.

The second step, named Classification, refers to the classification of the detect changes in step 1. Changes can be classified into 3 groups:

Table 3.3: Advantages and disadvantages of change detection methods (from [5])

	Direct Comparison	**Trace of Changes**
Advantages	Work With any modelling tool	Simpler Algorithms Accurate changes output
Disadvantages	Complex Algorithms Changes not always accurate	Needs a modelling tool that keeps trace of change on models

1. Not breaking changes;
2. Breaking and resolvable changes;
3. Breaking and unresolvable changes.

According to the authors of this approach, the first group is not relevant because these changes don't change the instances of M2 models (M1), so no migration is needed. The second group is the group of changes that can be propagated to M1 models without user input. Finally, the third group is a group of changes that although they can be propagated to M1 models, some user input is needed to execute the propagation.

The third step execution depends on of the type of changes. If there isn't any change of "Breaking and unresolvable changes" type, this step is not needed.

The four-step, Algorithms determination, will generate changes necessary to do the migration, i.e., generate a list of necessary steps to turn the old M1 models in new M1 models aligned with the new version of the M2 model.

The fifth and last step is the migration of the M1 model itself.

3.4.4 Automation Process for Co-evolution of EA Meta-Models and Models by Silva, Nuno

Another approach to address the co-evolution is presented by Silva et. al [6] (see Figure 3.5). In this approach, a 5-step process for semi-automated process for co-evolution was created, considering automated and non-automated activities.

The process consists of five activities and requires minimum user intervention. The *Select Change* activity allows the user to choose one or more operations from a change catalog to be applied to one or more EA meta-model elements. These operations can be constructive, destructive or refactoring [8]. The *Propagate Change* activity "supports propagation of the effects of changing a meta-model element to other elements that are strongly connected to it, i.e., elements having a single connection to elements they depend on." This step is required when operations of a destructive nature are applied [6]. The *Check Conformance* activity provides a comparison between the newest version of the EA meta-model and the current EA model version by assessing the model's compliance with the new meta-model

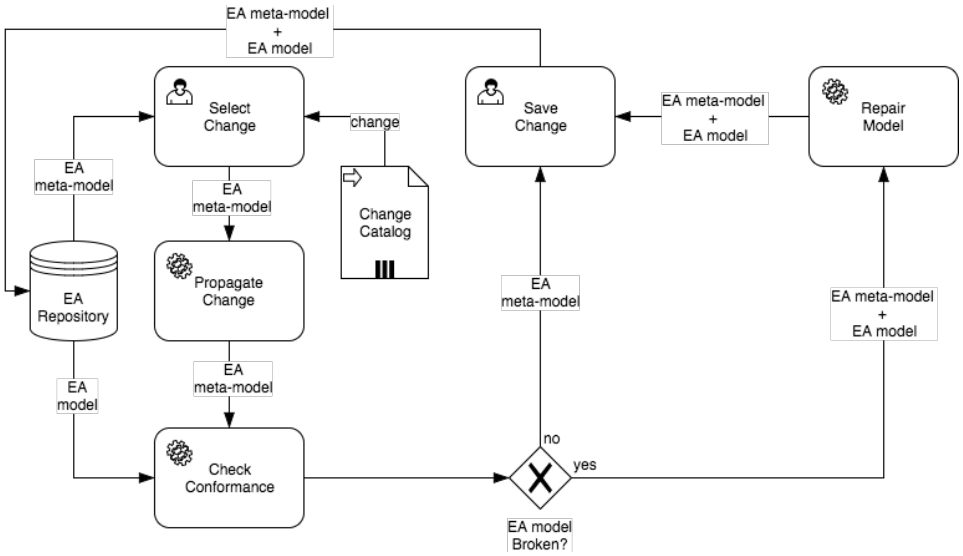

Figure 3.5: The semi-automated co-evolution process (from [6])

version. If inconsistencies are found, the *Repair Model* activity repairs the EA model, ensuring meta-model conformance. Finally, the user can then opt for either saving the changes in the EA repository or discarding the changes (*Save Change*) once both the EA meta-model and EA model are properly updated.

3.5 Discussion

The last sections mentioned previous research on EA Transformation (Section 3.1), followed by an analysis of 2 surveys on EAM tools (Section 3.2). From there, an analysis of visualization of changes on EA models (Section 3.3) is presented, as well as, an analysis of meta-model and model evolution (Section 3.4).

The most significant impact from the first section (Section 3.1) on this research is the application of the life-cycle concept to all the existent EA elements. On Section 3.2, the analysis of the surveys provide essential information on the problems presented on EAM tools, as well as, understand the used visualizations on them and existent interactions with the visualizations. One of the problems stated in this section is the low interaction with the EA visualizations. To complete the analysis of the EAM tools, an analysis of 2 papers on EA changes visualization is also presented, where two approaches are described. From this approaches, the main contribution is the visualization of the changes using color patterns to codify states of EA elements. To finalize, the last section (Section 3.4), approaches of the co-evolution process and implementation of migration rules are presented. The co-evolution processes presented in the last section represented the basic steps of the presented tool workflow. These approaches become the main contributions to this research, being fundamental to the artifacts

presented.

4

Research Proposal

Contents

This chapter describes the software tool developed during this research, starting with the definition of objectives and then the description of each DSR iteration.

4.1 Objectives

This section identifies the objectives that the proposed solution aims to achieve. For that purpose, and in addition to the existing literature, the following requirements were also identified from a set of interviews with practitioners as input for identifying the objectives:

1. A solution should allow the saving of change's motivation;
2. A solution should allow the migration of the model, after meta-model changes;
3. A solution should provide the description of change's impact;
4. A solution should allow the undo and redo of EA meta-model operations;

Considering both the requirements above and the existing literature, the tool aims at achieving three main objectives:

1. EA practitioner support on EA co-evolution using a simple and interactive EA meta-model editor and visualizer;
2. EA practitioner support on EA co-evolution by providing change impact analysis features of specific model changes;
3. EA practitioner support on EA co-evolution concerning the migration of EA model.

4.2 Tool's Core

Following the objectives stated above, the tool's architecture must be grounded in a set of concepts and relationships that define the evolutionary aspects of an EA. The conceptual model on Figure 4.1 was implemented as the tool's base entity-relationship model, with the required properties for each created entity.

The remainder of this section presents the different elements of the conceptual model. An EA Project acts as the enabler of EA evolution by applying a set of EA Transformations that transform the AS-IS state of an EA description into a TO-BE state. Each transformation can be decomposed into a set of EA Operations, each one altering the EA Life-Cycle of an EA Class, EA Property, EA Object or EA Type.

Also, the tool's co-evolution process started based on the process defined by Gruschko et al. [5], although, in the second DSR iteration the proposed tool implements part of Silva et al. process [6] (see

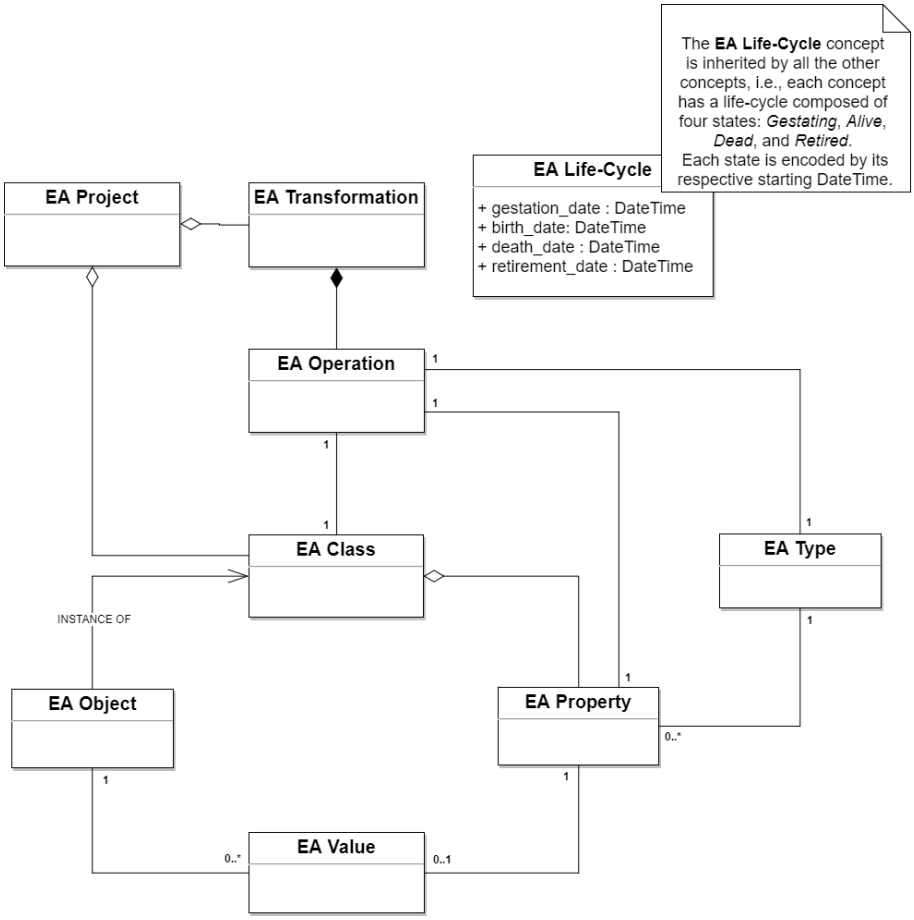

Figure 4.1: Tool Conceptual Model

Figure 4.2) since it was more explicit and simple. The tool will support migration rules proposed by Silva et al. [8] as well as the proposed by Wachsmut et al. [7]. In detail the migration rules that have been implemented are:

- Change EA Property Type;
- Eliminate EA Class;
- Eliminate EA Property;
- Introduce EA Class;
- Introduce EA Property;
- Move EA Property;
- Rename EA Class;
- Rename EA Property.

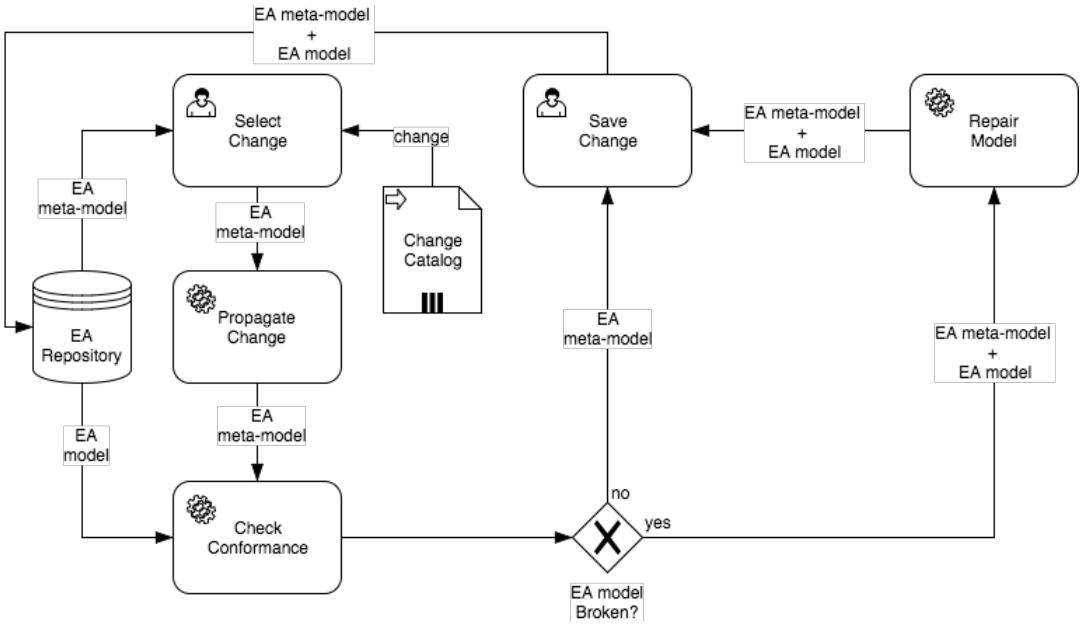

Figure 4.2: The (semi-)automated co-evolution process (from [6])

The tool was implemented as a future feature of a vendor-specific, highly configurable EAM tool[1]. The tool features a set of EA Transformations that alter the EA meta-model's state. Afterwards, one can visualize the impact of those changes on the meta-model's viewpoint.

The tool allows for the creation, deletion, edition, and submission for approval of EA Projects, thus enabling the creation of different evolution scenarios portraying the organization's future landscape.

[1] http://www.linkconsulting.com/eams/

Initially, after creating an EA project, since no transformations exist, the viewpoint presents the *AS-IS* state of the EA meta-model, without showing any changes, as expected.

Each EA Transformation is associated to one or more of ten EA Operations (Create EA Class, Rename EA Class, Move EA Class, Create EA Property, Rename EA Property, Rename EA Property, Change EA Property Type, Move EA Property, Remove EA Class and Remove EA Property).

Each operation, to maintain environment coherence and consistency, has no immediate impact on neither the production version of the EA meta-model nor EA model elements. The changes will be propagated to the production meta-model and model on the EAM tool only when the users decides to.

These operations can be composed of more complex transformations, for example, Copy EA Class or Copy EA Property.

4.3 First DSR iteration

On the first DSR iteration, after a set of interviews with practitioners a low-fidelity prototype was designed in order to create a first approach to a tool that supports the co-evolution of EA meta-model and model.

4.3.1 Prototype Workflow

The EA meta-model and model co-evolution process on the prototype is based on a 7-step workflow as Figure 4.3 illustrates. This process is based on 7 activities described on the following points.

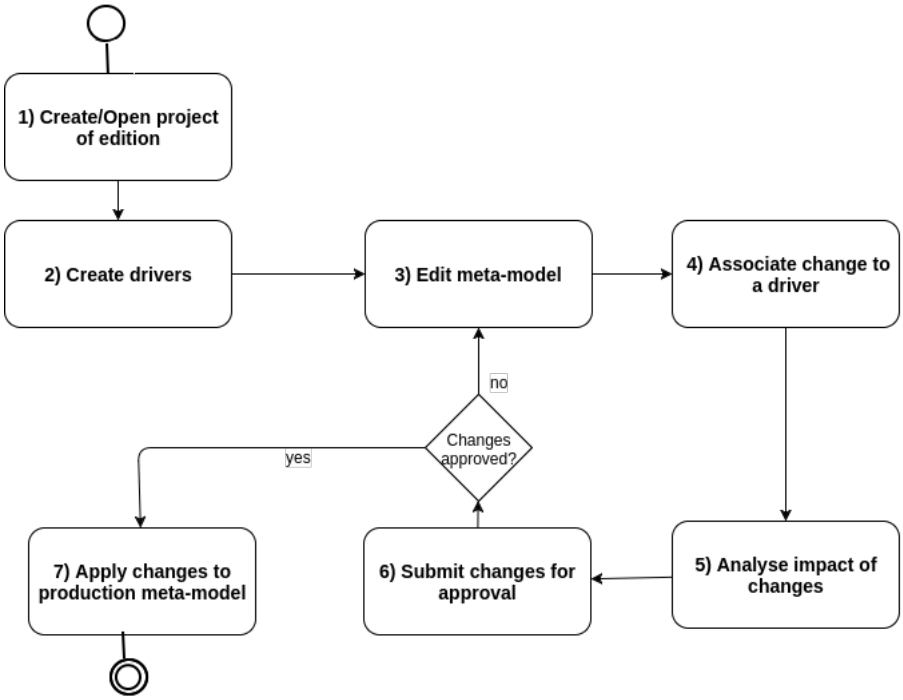

Figure 4.3: Prototype usage workflow

Create/Open project of edition - First, the user creates or opens a saved project for meta-model edition. Only inside the project scope is possible to edit the EA meta-model. This concept of ”Project” is mapped into the concept *EA Project* of the tool's conceptual model.

Create driver - In order to edit the meta-model, from both retrieved practitioners and researchers data, the reason behind a specific change must exist. To do this the concept of *Driver* has been created and associated with one or more changes made to the EA meta-model.

Edit meta-model - Editing the meta-model is mostly applying a set of predefined transformations to it.

Associate transformation to driver - A transformation to driver association must also exist, explaining the motivation behind a meta-model change. The association could be made while adding a transformation to the meta-model or after the transformation has been added.

Analyze impact of transformation - In this step, the user is presented with analysis-specific features regarding the impact of a meta-model change to the respective model. The analysis is based on indicators of incoherences and problems with relations between classes, supporting the user in the decision-making task of which evolution scenarios would be preferred.

Submit changes for approval - Meta-model (and consequent model) changes must be approved before updating the organization's EA repository. Therefore, the tool also allows the user to submit the evolution scenario for approval.

Apply changes to the EA repository - After the changes have been approved, they must be implemented, i.e., be applied to the current version of the EA repository. In this step, the user selects an approved project and applies the editions made to the EA repository. In this step the modifications are also associated with life-cycle dates. Is also at this step that the migration of the model is automatically done following the migration rules defined by Silva et al. [8] and by Wachsmut et al. [7].

4.3.2 Prototype's UI Structure

In order to support the process specified above, the prototype's UI (see Figure 4.4) is structured in three items: a tree-like data container with the associated drivers and transformations, EA meta-model visualizer, and transformation window.

The user can see which drivers and transformations were applied to the meta-model and also analyze the impact of applying the transformations (and changes) to both the meta-model and model. This feature allows the user to filter for transformations that (s)he wishes to see. Also, the future state of the EA meta-model, i.e., the new meta-model version after applying the changes can be seen on the meta-model visualizer with colors representing what changes were applied to a specific meta-model concept or relation type following the approach of Roth et al. [4].

Besides the graphical representation of the meta-model presented by the meta-model visualizer, one can choose how (s)he wants to view the meta-model, i.e., view AS-IS and TO-BE side by side, or as an integrated view in which both the AS-IS and TO-BE are incorporated into a single view. Another view option is the *Domain*, an option that works as a filter in which the user can choose, from the existent

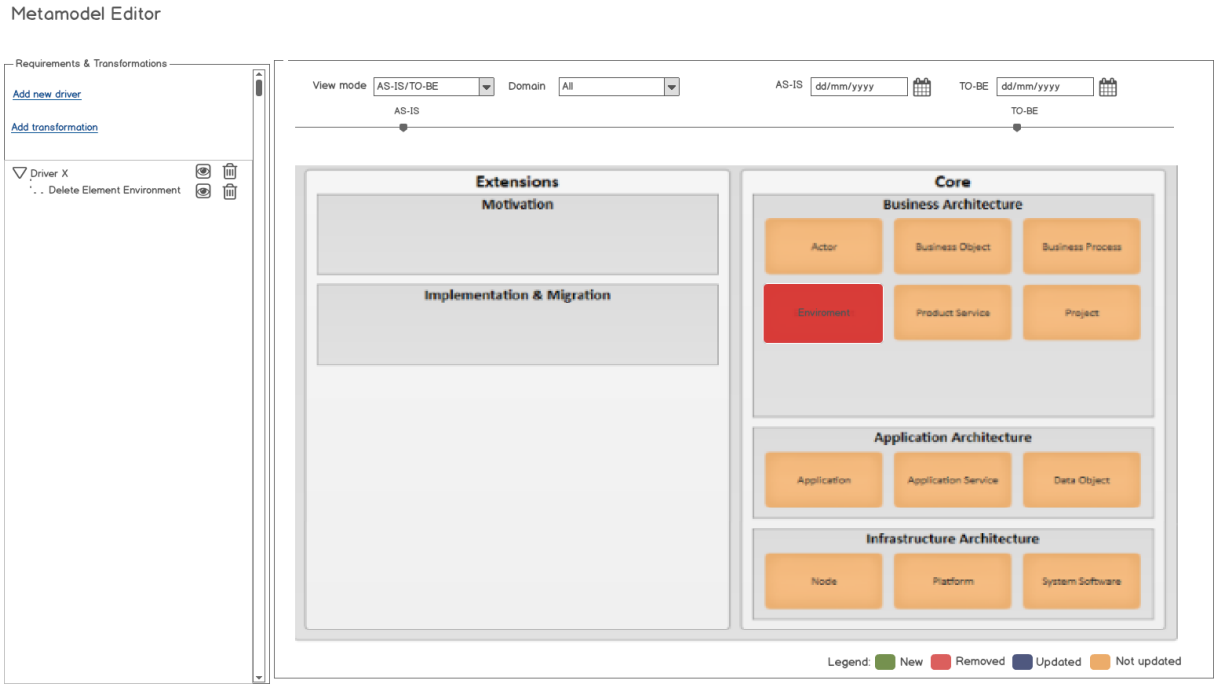

Figure 4.4: Prototype's UI

EA meta-model domains, the type of architecture he wants to focus the visualization on, thus, allowing either a holistic or domain-specific view of the EA meta-model.

An AS-IS/TO-BE date options, as well as the AS-IS/TO-BE bar, work as a time navigation feature, allowing the user to navigate to a previous version of the EA meta-model and observe which changes were made between the two chosen dates.

The transformation screen allows the user to (1) fulfill the properties of a transformation, (2) apply the transformation to the EA meta-model, and (3) observe the impact of such transformation on the model. To view the properties screen the user has two options:

1. An interactive view in which the user goes to the meta-model visualizer and right-clicks an element or architectural domain to identify the applicable transformations;

2. Via the *Add transformation* button option.

The first option allows the user to interact with the model and also filters all the possible transformations. The second one is more conventional, where the user has to fill in both the element and the transformation to be applied to that element. Both options navigate the user to the same screen where he or she has to fill some properties (for example, the association with a driver or creation of a new one at that moment, as well as seeing the change impact of that transformation).

Concerning the change impact analysis, it covers quantitative aspects as number/percentage of instances impacted, number/percentage of relations between classes lost in a depth chosen by the

user.

4.4 Second DSR Iteration

The second DSR iteration was divided in 3 user iterations that create the final version of this research. The objective of 3 user iterations was achieve a better UI and human-computer interaction, starting with the creation of 3 different screens and iterating until the achievement of a final version.

4.4.1 First User Iteration

The results from the first DSR iteration (see Chapter6) addressed the research objectives, however, to add more value and soundness to the tool, new interviews with users were made. The interviews had the goal of gathering information and create different versions of the same tool's UI to test other approaches. From this iteration 3 different versions (see Figures 4.5, 4.6, 4.7) were created based on the different information of the interviews.

The first version (see Figure 4.5) was created based on Archi[2] UI because of the users' acquaintance with the Archi modelling tool.

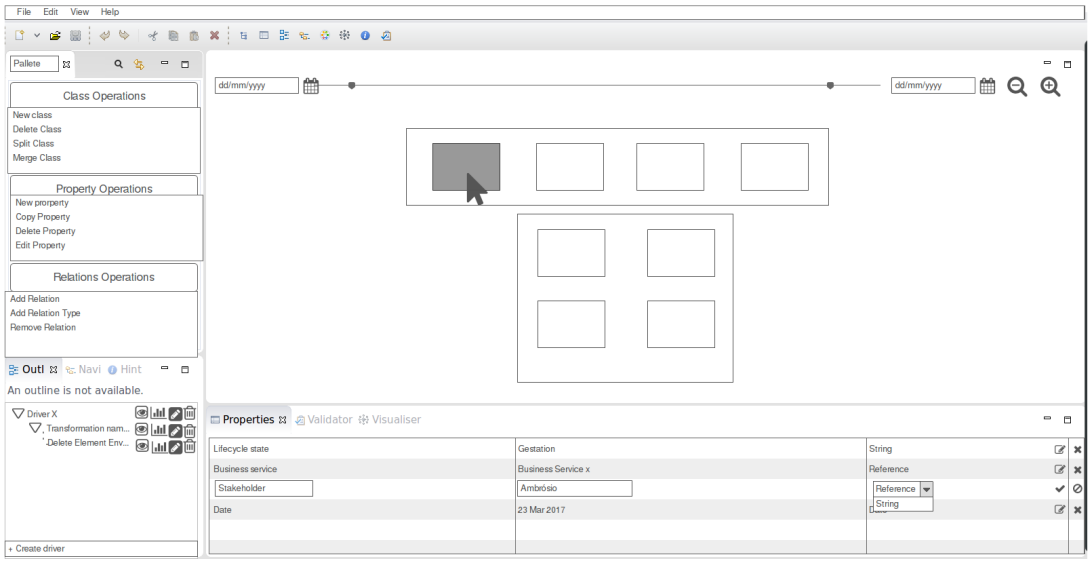

Figure 4.5: First user iteration, UI's first version

The second version was an improved version of the first DSR iteration with more features, and changes in human-computer interaction, resulting in the UI on Figure 4.6.

The third version is distinct from the others, having an almost completely different interaction. In this version (see Figure 4.7) the concept of "Requirements" was removed, since users pointed that when

[2]https://www.archimatetool.com/

Figure 4.6: First user iteration, UI's second version

meta-modeling the focus needs to be mainly on meta-modeling act. Hence, by fulfilling requirement forms they are detracting effort from the meta-modeling task, making the process of co-evolution too long. To simulate the concept of ”Requirement” the concept of ”Save” was created. Using this concept the user could create different saves during the process, almost as the well-known concept of ”Commit” in version control software for software development. This concept of ”Save” does not mean that the tool would not save automatically all the changes, but only that the user could create groups of automatically saved changes.

Comparing the three versions, the main differences between them are the following:

- On the first version, a change catalog is shown to the user, and no iteration is presented on meta-model visualization, except the selection of classes to edit.

- On the second version, the add transformation button was created. A transformation is a change that could be made to the meta-model, and from there the user could choose which change will be made. Interaction with meta-model's visualization is also possible using the right mouse click. Using this last method, the tool will filter the possible transformations and only show that ones to the user.

- On the third version, the ”Requirement” concept is removed, and added the concept of ”Save”. Moreover, all the changes to the meta-model should be made interacting with meta-model's visualization.

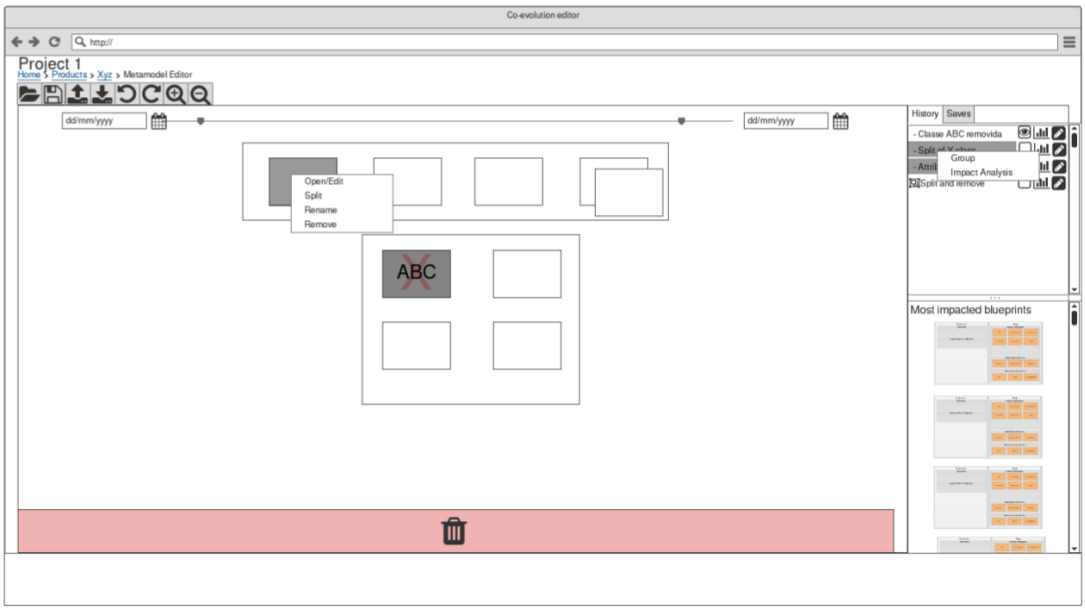

Figure 4.7: First user iteration, UI's third version

4.4.2 Second User Iteration

This iteration (see Figure 4.8) was made to join the three versions from the second iteration and make it more seamless with a web application.

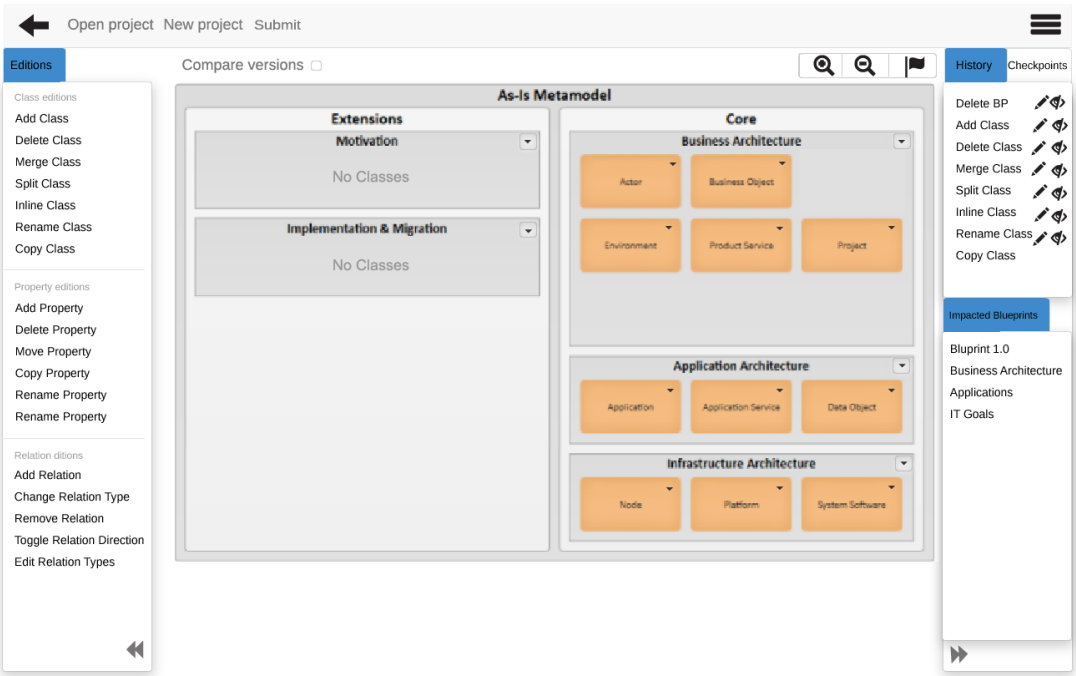

Figure 4.8: Second user iteration UI

This version received the best from each of the three versions of the Section 4.4.1, like the human-

computer interaction, the ”Save” concept from the third one (see Figure 4.7) now renamed to ”Checkpoint”, the interaction with meta-model’s visualization elements from the second (see Figure 4.6) and the change catalog from the first one (see Figure 4.5). Now, the impact blueprints do not show a preview of the blueprint, but only its name since practitioners usually know what it represents.

From this version, a final version was created based on insights from the users’ interactions as well as technology constraints.

4.4.3 Final User Iteration

After gathering insights from the users, and analyzing all the information gathered, a final version of the tool was developed. Figure 4.9 illustrates the tool’s UI using an example EA meta-model before its edition (*AS-IS*).

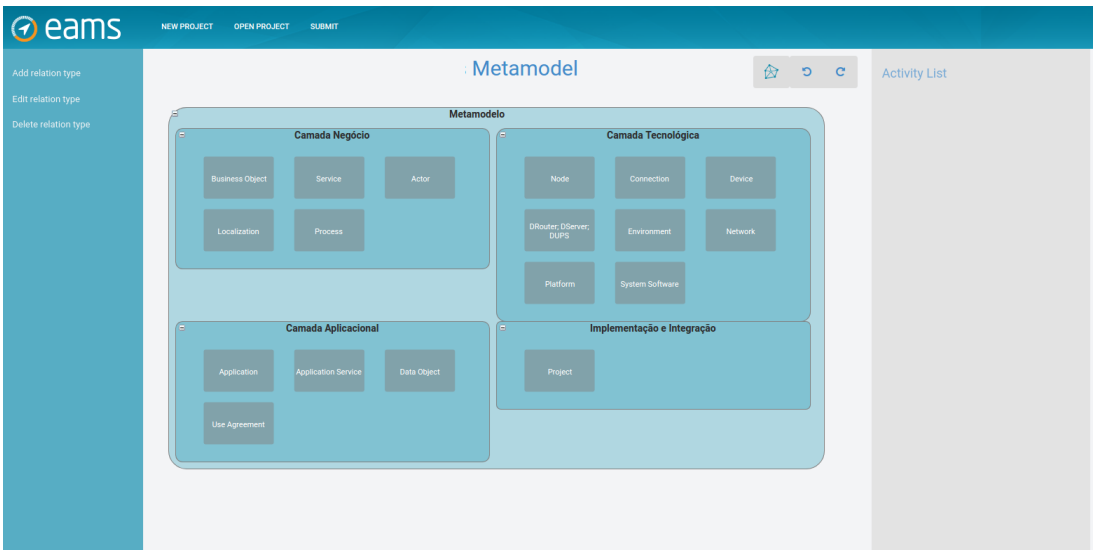

Figure 4.9: The tool’s enterprise meta-model initial screen

The proposed tool implements part of Silva et al. co-evolution process [6] illustrated in Figure 4.10.

Tool’s Workflow

As stated previously, the interface is based on Silva et al. co-evolution process [6], illustrated in Figure 4.10. However, a few changes were made to allow the user to create various projects on the same project and test various the changes. The workflow implements is illustrated in Figure 4.11.

To get a holistic perspective of the user’s iteration with the tool, we describe each activity in detail:

Create/Open project of edition - If the user is creating/opening a copy of the production meta-model on a certain repository to edit.

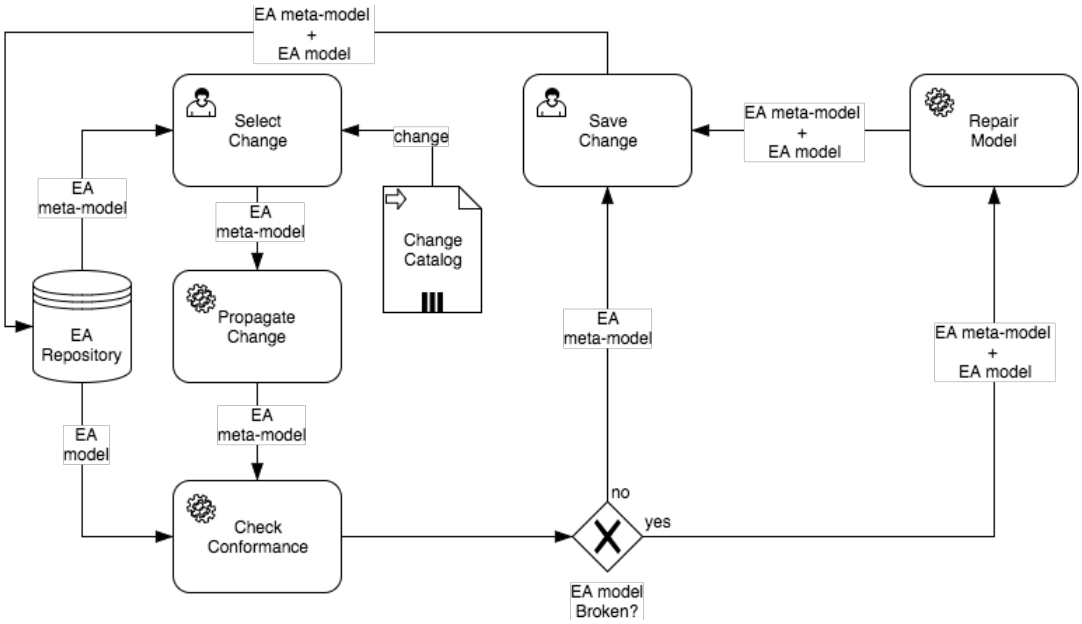

Figure 4.10: The semi-automated co-evolution process (from [6])

Edit meta-model - The user changes the EA meta-model applying the provided transformations and operations. This edition is only made on the copy of the meta-model made in step 1.

Analyze impact of changes - Analyzing how many objects were impacted or the classes that were impacted by a certain change, the user could decide to accept the change or not, undo could be made to the changes. This step will also contemplate the impact on the EA model entities.

Submit changes for approval - If the user has done all the necessary changes to the EA meta-model and wanted to apply them to the EA meta-model in production, he/she must submit the changes for administration approval.

Apply changes to the EA repository - If the changes were approved the user could then apply the changes to the EA repository by clicking on the ”Send to EAMS” option. Afterwards, the system updates the EA meta-model in production and migrates the EA model to ensure meta-model conformance. This final migration step is currently under development.

In order to have an interactive EA meta-model visualization, the user must apply all the editions to the EA meta-model, except relation type’s related editions, via meta-model’s visualization iteration. For example, to add a new class to a specific domain, the user right clicks on the domain and clicks on the ”Add Class” option. To move a class from a domain to another, the user drags a class from its previous domain and drops it on the target domain.

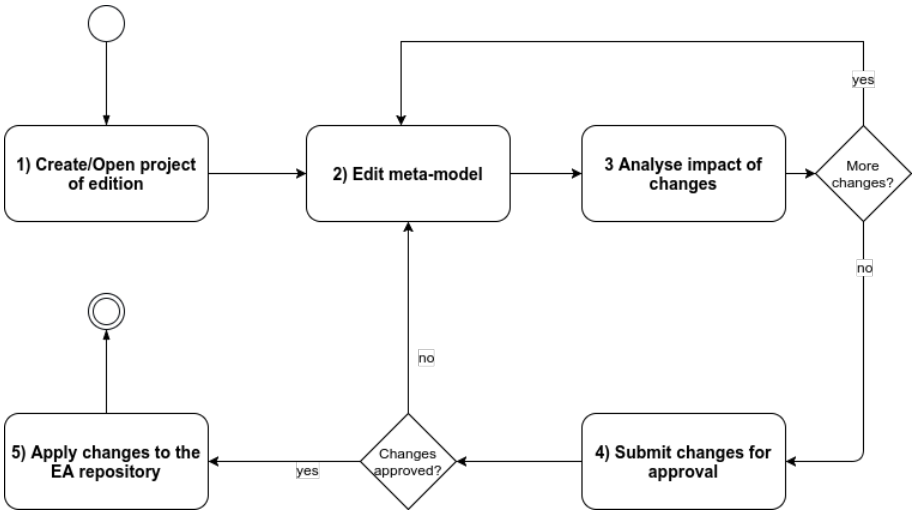

Figure 4.11: The tool's workflow

The component's color (see Figure 4.12) is related to the state of an EA Class as stated by Roth et al. [4]. The viewpoint itself is also interactive. For example, a double-click on a specific EA Class navigates to the class properties edition window (see Figure 4.13).

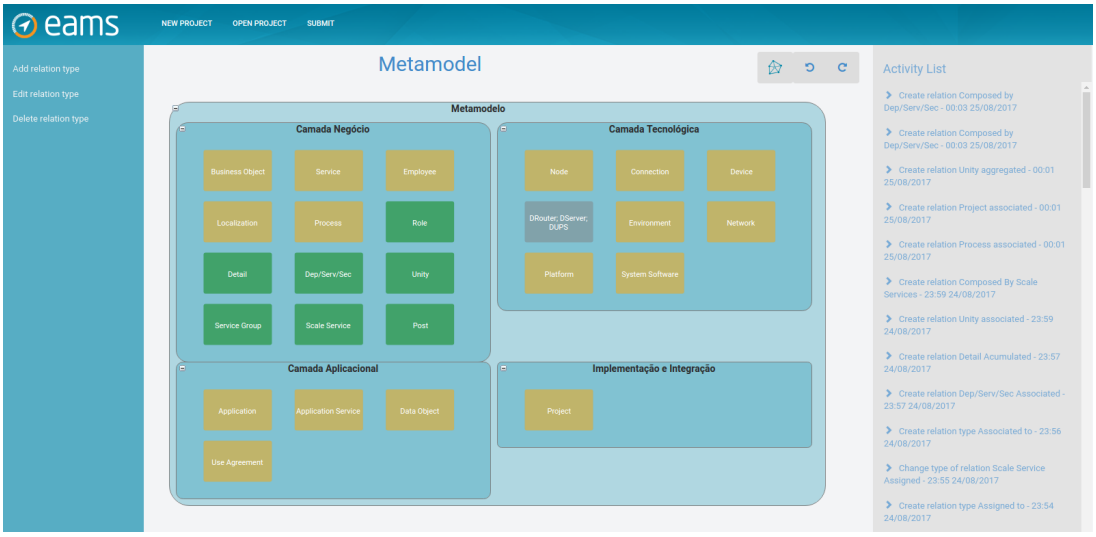

Figure 4.12: The tool's modified EA meta-model screen

The "Submit" option followed by "Send to EAMS" option migrates the EA meta-model in production to its new version, as well as the EA model, hence maintaining meta-model and model conformance.

Tool's UI Structure

To support the previous described workflow the interface tool is divided in 2 different screens:

- **Projects Screen** (see Figure4.14) with a table where each row represent a different project created

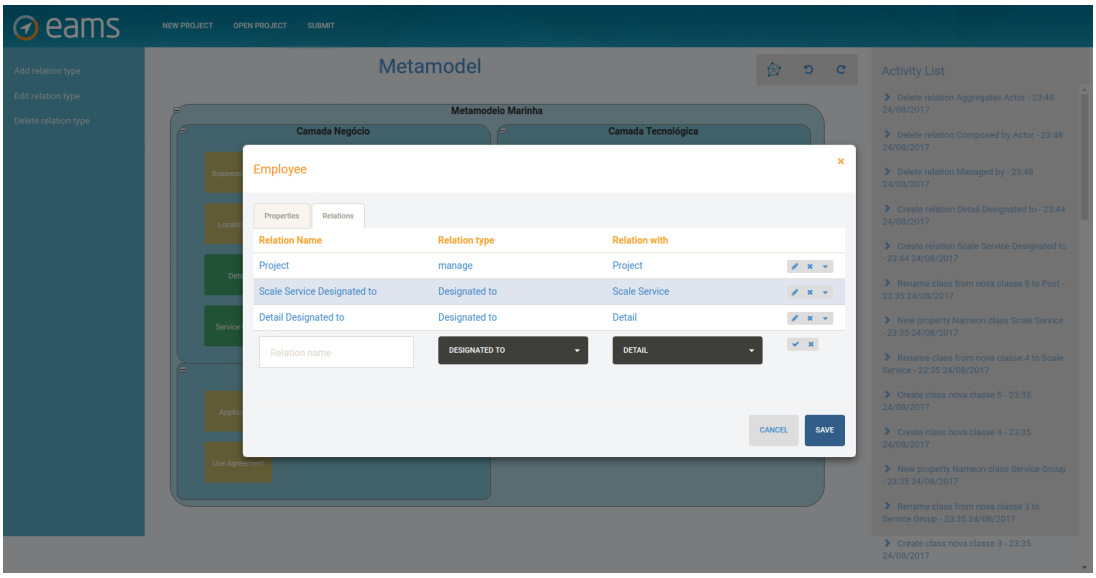

Figure 4.13: The tool's edit class window

by the user with the different properties of a project. In this screen, the user has feedback about the state of the project. The different states possible are:

- Not Submitted - no editions had been submitted to revision of the administrator;
- Submitted - meaning that the project had been submitted to administrator revision;
- Accepted/Rejected - if the administrator accepts the editions, the state of the project becomes "Accepted", otherwise becomes "Rejected".
- Sent - if accepted the user could send the editions to the base EAM tool clicking on a button.

- **Edition Screen** (see Figure4.9) on top-left corner, project functions as "New project", "Open project" and "Submit" are shown, allowing the user to create a new project, switch between projects or submit the actual changes for administration approval respectively.

  On the left side, one can apply one of three different operations: create/edit/delete relation types. Each operation allows the user to create the relation types which are best suited for the organization's needs.

  On the center, an interactive viewpoint illustrates a comparison between the enterprise meta-model's AS-IS state, i.e., before the execution of the EA Transformations, and the EA meta-model's TO-BE state, i.e., after applying the EA Transformations. The viewpoint (see Figure 4.9) is dynamic, since adding or removing EA Transformations will update the viewpoint, displaying the impact of new transformations in real time. Here it is also possible to switch the view to a graph-based visualization of the EA meta-model (see Figure 4.15), however, this visualization is not interactive like the previous one.

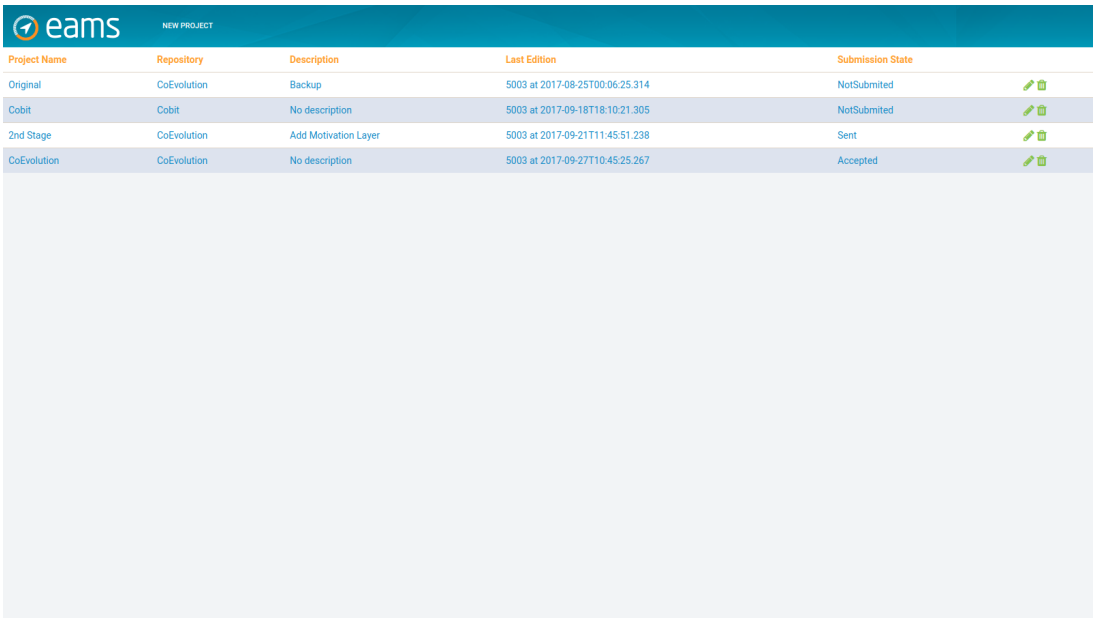

Figure 4.14: The tool's project screen

On the UI's right side, an activity list of all performed actions is shown as an edition history, aggregating all the operations and transformations made by the user. This activity list is shown as a tree, where inside each element it is possible to observe all the operations made regarding that transformation.

Tool's Modules

On this section, an overall view of the tool's modules is presented in order to have a better understanding of the artifact. The tool is divided into three main modules:

- **The EAM tool** module is the tool used that contains all the EA data in a repository. In this research, EAMS was used, but others could be used if they provide a Representational State Transfer (REST) Application Programming Interface (API), for meta-model and model edition.
- **The Back-end** module was created using Spring Boot Framework[3], a Java Framework used on the EAM tool where this tool should be integrated. This module is also divided into modules, in this case, in this case on the following modules:
    1. Controllers;
    2. Entities;
    3. Managers;

[3]https://spring.io/

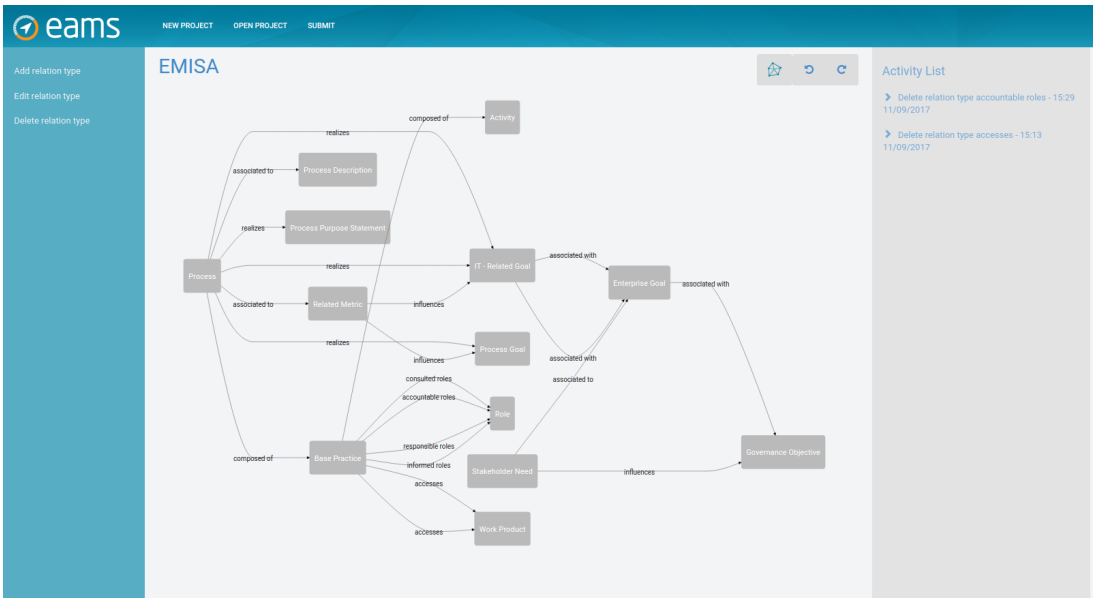

Figure 4.15: The tool's meta-model relations view

The first module of the Back-end module, **Controllers**, is responsible for the controlling the flow of the application execution. When a request is made, a controller is responsible for returning the response to that request. The controller can return a view (a Thymeleaf[4] template) to the Front-end module or return a JavaScript Object Notation (JSON) array/object, in this last case, is called a REST controller, since it represents an endpoint for the tool REST API.

The **Entities** module is the module with all the existing tool's entities. The existing entities are mapped to the tool's conceptual model (see Figure 4.1 on Section 4.2) on the Table 4.1

As is possible on Table 4.1, the mapping is not complete due to implementation choices (in case of absence of mapping an information model's concept to an entity) or to the addition of more features (if the absence of entity mapping on an information model's conception). A special case occurs in the mapping of the EA Property information model's concept since it is mapped to three different entities. This case arises since presented tool on this research distinguishes relations from properties. Relations are connections between one or two EA classes and properties are attributes of a certain EA class. Finally, the **Managers** module is responsible for managing all the operations on the entities, i.e., the application of all the migration rules chosen by the user on the EA elements.

- **The Front-end** module is responsible for all what is seen by the user. The Front-end provides views that are sent to the user by the Controllers module of the Back-end module. The front-end allows the user to make all the interactions with the visualizations of the meta-model and perform

[4]http://www.thymeleaf.org/

Table 4.1: Tool Conceptual Model mapped on Entities

Information Model Concept	Entity
EA Project	EAProject
EA Object	EAObject
EA Value	
EA Class	EAClass
EA Property	EAProperty EARelation EAObjectRelation
EA Type	EARelationType
EA Operation	-
EA Transformation	-
-	EAAction
-	EAHistory
-	EACategory
-	EAUser
-	EAMetamodel

all the actions provided by the tool. The two different visualizations of the meta-model are provided by two different JavaScript (JS) libraries:

- mxGraph[5];
- dagre-d3[6].

The mxGraph is responsible for the first visualization of the meta-model (see Figure 4.9), as well as, the interaction with the visualization and triggering of different events (Move, Right-Click, Left-Click, Double-Click, etc.). The events after triggered are handled by other JS functions that perform the tool's workflow on the Front-end, or even, the communication of that events to the Back-end (for example, when moving an element from a domain to other domain). The position of the elements, contrasting with the dagre-d3 library, is implemented in this research since with the algorithms presented in the library was not possible to achieve the expected result.

The dagre-d3 is responsible for the relation visualization, performing an algorithm for positioning of the different elements taking into account the relations between the different elements on the visualization. The algorithm is derived from Brandes and Köpf, "Fast and Simple Horizontal Coordinate Assignment" [33] and already implemented in the library.

[5]https://github.com/jgraph/mxgraph
[6]https://github.com/cpettitt/dagre-d3

5

Demonstration

Contents

This chapter covers the demonstration phase of the research method from both DSR iterations. The first iteration describes a low-fidelity prototype applied in the context of an EA transformation and on the second iteration the application of the developed tool in an EA project for the Portuguese Navy.

5.1 First DSR iteration

On the first iteration, a prototype (see Figure 5.1) was applied in the context of an EA Transformation phase with the purpose of measuring the artifact's efficacy on supporting EA practitioners evolving an EA meta-model, as well as showing the impact on the EA model.

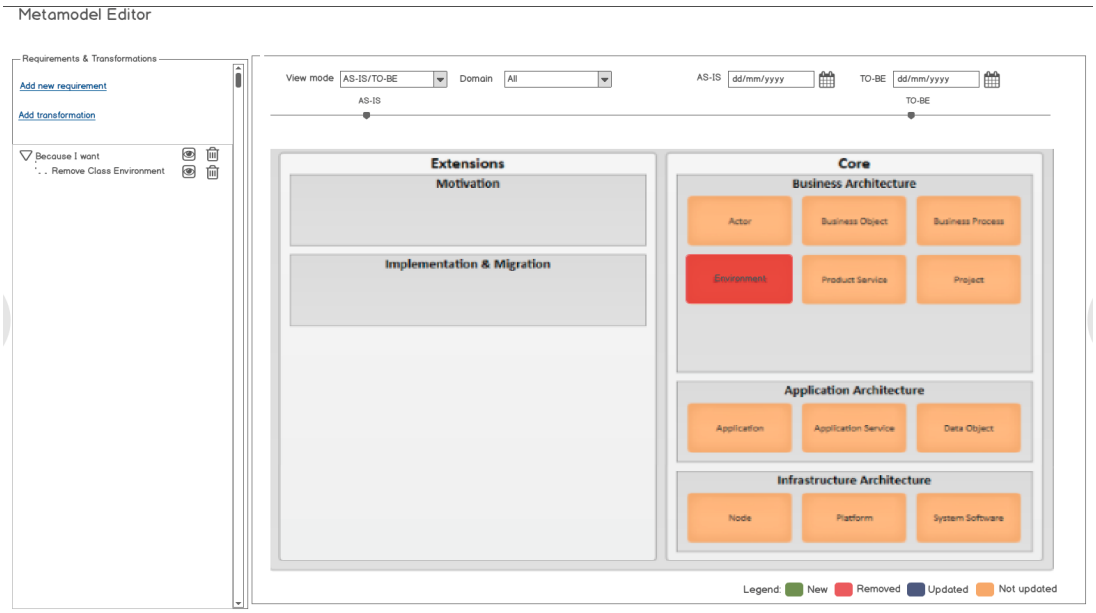

Figure 5.1: The interface's modified EA meta-model screen

The transformation phase included 3 simple main tasks from the creation of a requirement to analyze the impact of an EA meta-model change to the EA model. The full list of the tasks is the following:

- Task 1
  1. Create new requirement on the open project with the name "Because I want".
- Task 2 (see Figure 5.1)
  1. Apply transformation "Remove" to the Environment class and associate it with the requirement "Because I want";
  2. Mention which queries will be automatically changed and which will not.
- Task 3

1. Hide the impact of changes on the visualization (see Figure 5.2);

2. Show the impact of changes on the visualization;

3. Delete the transformation done on task 2;

4. Delete the requirement created on task 1.

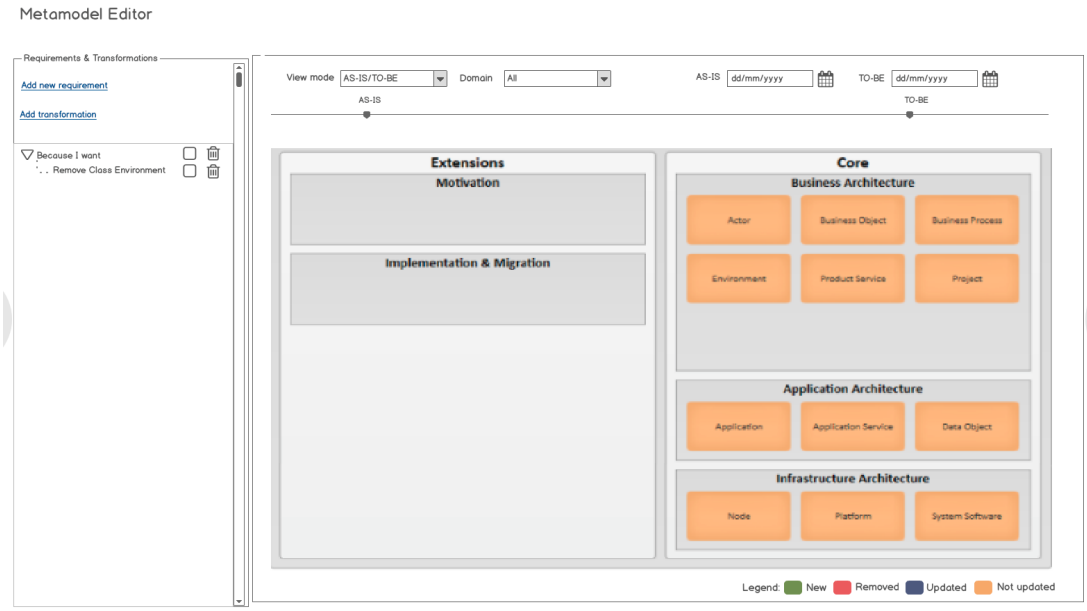

Figure 5.2: First DSR iteration demonstration step 1 of task 3

For this demonstration, a set of 18 academics and 2 EA practitioners were approached to perform a co-evolution scenario using the proposed interactive prototype. The 18 academics were former EA students, some of them EAMS users in the context of their master thesis. The approached EA practitioners approached were part of the EAMS team. None of the academics or EA practitioners failed to do any of the tasks.

The prototype interaction was made using the InVision App[1], creating an interactive UI that allows a straightforward UI testing. All the input required from the user was already pre-filled on the prototype, only mouse interaction was needed. This demonstration allowed to gather information about the interaction of the users with the tool, mainly testing the usability of it and the concepts presented.

After the demonstration all the users answered a questionnaire, in order to evaluate the interface usability on academics case, and utility on EA practitioners case. In the next chapter, the evaluation and it's results are described.

[1] https://www.invisionapp.com

5.2 Second DSR iteration

In the second iteration, after developing the tool, to assess the tool's effectiveness in addressing the research problem, a field experiment for the Portuguese Navy was made in the scope of an EA project.

The EA project, divided into two main phases, consisted of merging three different EA meta-models used by different sectors of the organization, keeping the biggest amount of information presented in the meta-model and also extend it with new information.

The first phase was composed of two steps:

1. Map the common concepts of the three EA meta-models; and

2. Create a unique EA meta-model containing all the concepts and relationships presented in each one.

In this phase, a repository containing the EA meta-model with more data was used as input. Table 5.1 contains all the transformations and/or operations applied to the EA meta-model.

Table 5.1: First phase EA Transformations/Operations results

EA Transformations/Operations	#
Create EA Class	6
Copy EA Class	1
Rename EA Class	8
Create Relation Type	1
Create EA Property	12
Create EA Relation	-
Delete EA Relation	16

Table 5.2: EA transformations by operation type (Second iteration)

EA Operations/-Transformations	#	
	First Stage	Second Stage
Create EA Class	6	7
Rename EA Class	8	10
Copy EA Class	1	-
Create EA Property	12	14
Create EA Relation	2	45
Delete EA Relation	16	3
Create EA Relation Type	1	1
Create EA Domain	-	1
Rename EA Domain	-	1

Once the *to-be* EA meta-model was approved, these changes were sent to the EA Repository, updating the meta-model in production.

The second stage consisted of analyzing and discussing the updated version of the EA meta-model and all the concepts needed to cover every part of the organization that was not previously contemplated by any of the three EA meta-models. The EA operations, depicted in Table 5.2, were applied using the tool's features.

Figure 5.3 illustrates the tool's UI after adding the second iteration's transformations.

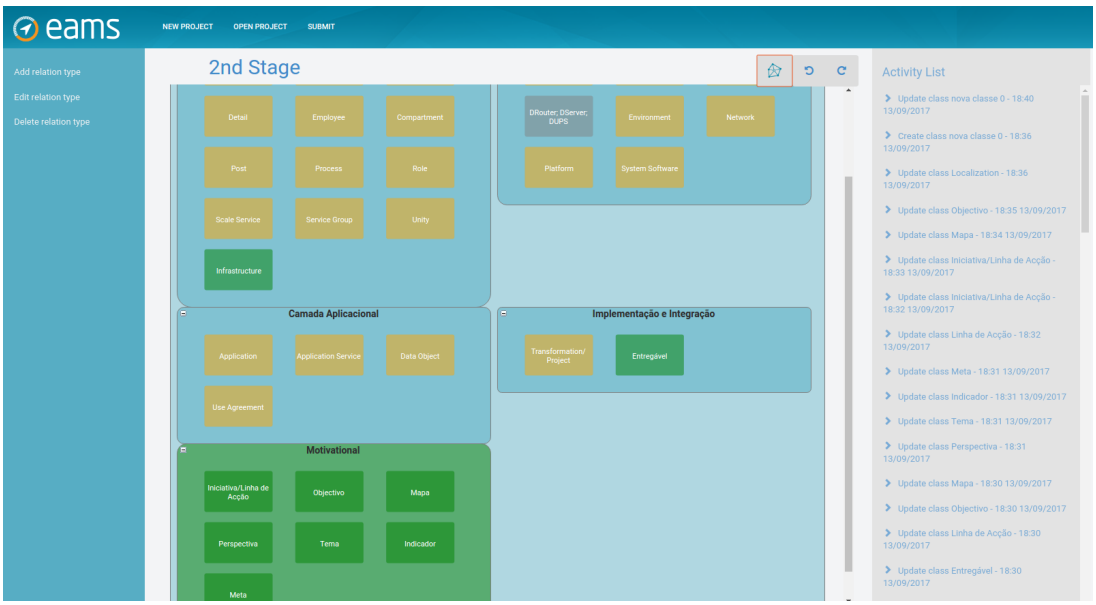

Figure 5.3: The interface's modified EA meta-model screen

The tool enabled a visual and interactive evolution management approach concerning the Portuguese Navy EA meta-model, thus attending to its needs, by applying the transformations and operations that together changed the state of the existing EA meta-model elements.

6

Evaluation

Contents

In DSR, artifact validation is done by applying design evaluation methods. Possible evaluation methods are *Observational* (by means of case studies or field experiments), *Analytical* (applying static analysis, architecture analysis, optimization, and dynamic analysis), *Experimental* (through controlled experiments and simulation), *Testing* (such as functional - Black Box - and structural - White Box - testing), or *Descriptive* (using informed arguments or scenarios) [17].

Observational, *Analytical*, and *Experimental* design evaluation methods were applied using a generic DSR evaluation model for IS artifacts [9].

The chosen evaluation criterion applied to the artifacts was as follows:

- **Observational:** Field experiment (first DSR iteration);
- **Analytical:** Dynamic Analysis with interviews, focus groups (second DSR iteration) and user tests;
- **Experimental**: Controlled Experiment with user tests.

Hence, the field experiment using a problem instantiation (EA evolution scenario of a real organization), interviews and focus groups (using a live demo of the proposed tool), assess the artifact's applicability and its goal (efficacy), whereas the user tests (from an experimental perspective) the usability of the interface. The instantiated generic DSR evaluation model [9] is presented in Table 6.1.

To assess its usability, UI performance was validated with user tests for both DSR iterations. As for the second DSR iteration, to assess tool's efficacy and utility 2 focus group with EA practitioners and 10 interviews also with EA practitioners were made.

6.1 First DSR Iteration

On the first DSR iteration, the UI was evaluated through questionnaires made to 16 students knowledgeable of EA and 2 EA practitioners after a demonstration of 3 tasks state on Section 5.1. The questions results, for academics, can be seen on Table 6.2 and for practitioners in Table 6.3.

Table 6.1: IS artifact evaluation model instances (adapted from [9])

Description	Assessed criterion	Form of evaluation	Secondary Participants	Level of evaluation	Relativeness of Evaluation
Demonstration of artifact's applicability with a field experiment	Goal/ Efficacy	Analysis and logical reasoning		Instantiation/ Real examples	Absolute
Qualitative feedback after the user tests	Environment/ Consistency with people/ Ease of use Environment/ Consistency with people/ Utility	Qualitative	Students/ Researchers/ Practitioners	Constructs/ Instantiation	Absolute
Qualitative feedback from practitioners on the utility and efficacy of the artifact.	Environment / Consistency with people /Utility Goal / Efficacy	Qualitative	Practitioners	Instantiation	Absolute

Table 6.2: Scores confidence intervals with 95% confidence level

Question	Range of scores
1) The interface communicates in a simple way	[5,79-6,34]
2) The interface uses the same language as the user	[5,26-6,37]
3) How do you consider the user mental effort?	[5,04-5,96]
4) How do you consider the consistency of the interface?	[5,04-6,33]
5) How do you consider the help features presented on the interface?	[4,04-5,33]

To draw significant conclusions from the questionnaire results, the confidence intervals statistical method was used based on the given result sample data. Each question had an average score higher than 4, with 7 being the best possible score (see Table on on AppendixA) . With an average score above 3.5 and a confidence level of 95%, it is acceptable to conclude that all target users will answer these questions in the range of scores, presented by the confidence intervals, on Table 6.2. The results of the questionnaire prove both the system's simplicity and interactivity were good and that the prototype evaluated should be taken into account in future developments.

Table 6.3: Practitioners results

	Practitioner 1	Practitioner 2
Using the system in my job would enable me to accomplish tasks more quickly	4	6
Using the system would improve my job performance	5	6
Using the system in my job would increase my productivity	6	7
Using the system would enhance my effectiveness on the job	6	6
Using the system would make it easier to do my job	5	5
I would find the system useful in my job	6	5
Learning to operate the system would be easy for me	6	6
I would find it easy to get the system to do what I want it to do	7	6
My interaction with the system would be clear and understandable	7	4
I would find the system to be flexible to interact with	6	6
It would be easy for me to become skillful at using the system	6	6
I would find the system easy to use	7	6

Due to the sample size (2 practitioners), a statistical analysis was not feasible. Although, based on the obtained results, it is possible to infer that the proposed prototype has the potential to solve the research problem.

6.2 Second DSR Iteration

After evaluated the first DSR iteration, 3 user iterations were made on the second DSR iteration. The final iteration resulted in the implementation of the tool and partial integration with an EAM tool. In order to evaluate the tool in terms of efficacy, interviews and focus were made. In terms of usability, user tests to 20 EA students and 4 EA practitioners were made.

6.2.1 Efficacy and Utility

To test the tool's efficacy three different methods were used, a field experiment, focus groups and interviews, in this case, semi-structured interviews, both of them with the same structure. The focus group allowed discussion between the present practitioners. Interviews, despite not adding discussion between practitioners, were made since joining practitioners from different organizations in the same room was impossible to make. In the next sections, a description of the evaluation, as well as, the obtained results, is presented.

Field Experiment

In Chapter 5 the instantiation of an EA project is shown. The project consisted on joining three different meta-models used by the Portuguese Navy. A set of EA Transformations and Operations were used to change the meta-model. After that, an explanation of the steps done was made to the Portuguese Navy's EA team, from whom the approval of the evolution was received. Afterwards, the changes were sent to the EAMS system, migrating the production meta-model and model. From this field experiment, the author can conclude that the artifact achieves the desired goal of providing an environment for the managing the co-evolution of EA meta-model and respective model.

Focus Groups

The evaluation with focus groups was divided into two different focus groups. The first was made with 2 members of Link Consulting[1] EAMS team, and the second with 3 members from the Portuguese Navy, 2 of them responsible for the Navy's EA.

First Focus Group From the first focus group, the result was positive, saying that the developed tool was a good starting point for the development of a fully functional one since this one is a tool concept and not ready for production. This focus group suggested that a better impact analysis is needed before this tool could be associated with a final product. The impact analysis should be more explicit and have more information about the implication of the deletion of a relation and a class. It should also be possible to understand the reason behind a particular model change (or set of changes), like an explanation of what is impacting what. Besides that, the focus group mentioned that the tool in the future should also change all the other artifacts presented on EAMS like blueprints and reports.

Despite the needed improvements, the tool facilitates the work and decreases the number of errors of the creator/editor of the EA meta-model on the EAMS. Most of the users edited the meta-model using

[1] http://www.linkconsulting.com/pt-pt/

a text file with the meta-model described in a text-based language, something that could create a lot of mistakes a lot of time debugging, and that is solved using the tool.

Second Focus Group The second focus group, composed by Navy's EA experts gave positive feedback. Like in the previous focus group, this focus group mentioned that the tool should have better impact analysis, allowing, for example, the visualization of the objects that will be impacted by the different changes. Nevertheless, the tool was extremely useful for them to edit the meta-model, allowing them to edit it easy in a safe environment and with much fewer errors than before. The possibility of creating a new meta-model from scratch was something the group also found to be useful.

Conclusion The tool's lack of expressiveness concerning the impact of meta-model changes to the EA model was the main conclusion from each focus group. Notwithstanding, the feedback was positive, and both groups concluded that the tool would facilitate their job, making it less error-prone. The second focus group, less accustomed to EAMS tool find it useful.

Interviews

The interviews were made with 10 different EA practitioners (not all have used EAMS, but the system was explained to them during the interview) from different companies and organizations, from consultancy and audit to telecommunications companies. The interviews have taken an average time of 28 minutes. They started with an introduction of the thesis, followed by the collection of opinions from the practitioners regarding the complexity of the research problem. Afterwards, a live-demo of the tool was made to all interviewers followed by a discussion on the various tool's features.

The research problem was recognized by the interviewees as an occasional rather than regular issue. It is indeed an issue that is typically associated to the acquisition of a new company or the necessity to meet the last notation standard of EA. One of the interviewers mentioned that the EAM tool used by him already allow the co-evolution of the EA meta-model and respective model, and it was easy to use.

After the demonstration, the interviewees gave their opinion, in which 10 out of 10 interviewers provide a positive feedback, despite the note of a more extensive impact analysis being a future improvement. In particular, 8 out of 10 acknowledged that this module is of practical relevance and facilitates the task of EA meta-model evolution. The user that already uses an EAM tool that allows the co-evolution said that the tool developed in this research is easier use and that for 70% of the cases it will be better to use this tool instead of the tool already in use. To achieve more percentage of cases, the tool should allow the definition of rules for the properties and relations. After this information, a trial of the tool mentioned by the interviewee was tested, and no signal of meta-model evolution was encountered. From what was possible to understand, there was only the possibility to extend objects already existent with

new information, not changing the underlying meta-model.

The tool's pros mentioned by the interviewers were:

- The importance of the activity list, working as a log of all changes made;
- Good user interaction;
- Fulfills necessity of EAMS users;
- Diagram with the meta-model relations;
- Workflow with administration review, thus reducing the probability of errors;
- Number of impacted objects when deleting a class;

The cons of the tool were the following:

- Lack of a more expressive model impact analysis;
- Visualization of impacted objects;
- Not keeping a standard color pattern associated with class changes to all elements;
- Inability to resume and quantify changes;
- Unable to filter by class' neighbors or domain in the meta-model relationship view.

Conclusion

Regarding the two focus groups, both provided a positive feedback by claiming the tool's capacity of reducing execution time and error-proneness.

As for the 10 interviewees, 8 argued that the evolution of the meta-model besides not being a regular task, it can be necessary and therefore, it is critical to perform safely. The 2 interviewees that did not agree with the remaining said that "the meta-model rarely changes after is defined", something that could be caused by the use of standard notations on the interviewers' projects, where the data is associated with the existent classes of the notation. Of the 8 interviewees, 100% agreed that the proposed tool has the potential to facilitate their job since it gives them a safe environment to make all the required changes.

In conclusion, taking into account the information retrieved from the focus groups and interviews, the author argues that the developed tool provides relevant features that aid towards EA meta-model and model co-evolution, hence being a possible design solution for addressing the problem of EA meta-model and model co-evolution.

6.2.2 Usability

In order to test the usability of the developed tool's UI, a sample of 20 Enterprise Architecture students, as well as 4 practitioners, performed 12 tasks. Before performing the user tests, a simple demonstration of important EAMS's concepts was shown to every user, as well as, a simple introduction of the research and finally all the users signed a consent form allowing the recording of images (see the Figure B.1 on Appendix B)

The 12 tasks are the following:

1. Create new project with name ”Demo-UserName” in order to change the meta-model of ”UserTesting-TiagoRechau”;
2. Visualize meta-model in relationship mode;
3. Go back to the structural mode;
4. Delete Class ”Process Purpose Statement” and mention how many objects will be deleted,
5. Create new class on ”Enabling Processes” domain;
6. Rename the created class to ”Indicators Measurements”;
7. Move ”Process Goal” class to ”Goals Cascade” domain;
8. Add new property named ”Type” in ”Activity” class of type ”Text”;
9. Remove ”Input” relation from ”Base Practice” class;
10. Change property type of property ”Type” from ”Activity” class from ”Text” to ”Numeric”;
11. Submit changes to administrator approval;
12. Delete your project.

The user tasks incorporated the use of different types of EA transformations/operations, as well as exploring the tool by using the supported information visualization features. The average time to perform all tasks was 4 minutes and 24 seconds, the lowest time being 2 minutes and 54 seconds and the highest 9 minutes and 3 seconds if we discount the time the users were expressing their opinions. Also, future time should be in the interval of [0:03:50, 0:5:00] with 95% of confidence level.

From the analysis of time concerning all tasks, tasks with similar interaction had their time highly reduced after the completion of initial tasks.

To assess usability after the tests, users were asked to answer a questionnaire regarding the tool's acUI usability. The questionnaire consisted of 10 questions using a scale of 1 to 5 (1 meaning that

the user strongly disagrees and 5 meaning he/she strongly agrees with the statement) and one more question to evaluate the overall user perception concerning the interface's usability. This questionnaire was a fully based on System Usability Scale (SUS) questionnaire [34] with the addition of a seven-point adjective-anchored Likert scale question [35], where from 1 to 7 we have the following adjectives: 1) Worst Imaginable, 2) Awful, 3) Poor, 4) Ok, 5) Good, 6) Excellent, and 7) Best Imaginable. The results of the questionnaire are presented on Table C.1 on Appendix C.

The results of the questionnaire result on 75% of users giving a score above 80 points with an average score of 81.56, hence and considering the work of Bango et al. [36] the interface gets a traditional school grade of B. Analyzing each question in detail we can conclude that:

- 87.5% users agree or strongly agree that they would like to use the interface;
- 83.3% users disagree or strongly disagree that the interface is unnecessarily complex;
- 79.1% users agree or strongly agree the interface was easy to use;
- 91.6% users disagree or strongly disagree they would need support of a technical person to be able to use this interface;
- 87.5% users agree or strongly agree on the various functions in the interface were well integrated;
- 95.8% users disagree or strongly disagree there was too much inconsistency on the interface;
- 87.5% users agree or strongly agree that most people would lean to use the interface quickly;
- 83.3% users disagree or strongly disagree the interface is very cumbersome to use;
- 70.8% users agree or strongly agree they fell very confident using the interface;
- 91.6% users disagree or strongly disagree they would need to learn a lot of things before they could get going with the interface;
- 75% of the users rated the user-friendliness of the interface as "Excellent" or "Best Imaginable".

Overall, the evaluation method provided satisfactory results that corroborate the fulfillment of the solution's objective. The worst point was the confidence on using the interface, which could be caused by the first use of the interface.

6.2.3 Conclusion

Since the results from SUS questionnaire reported an average score of 81.56, the objective 1 (Chapter 4) was achieved. From the demonstration presented in 5, the objective 3 was met since a completely

EA project in the context of EA meta-model and model co-evolution was done solely by using the proposed tool. The objective 2 was poorly met since the impact analysis present on the tool did not provide sufficient information to the interviewed EA practitioners.

Overall, and taking the following objectives of a solution into consideration:

1. EA practitioner support on EA co-evolution using a simple and interactive EA meta-model editor and visualizer;
2. EA practitioner support on EA co-evolution by providing change impact analysis features of specific model changes;
3. EA practitioner support on EA co-evolution by automated model migration.

Despite one of the three objectives being poorly met, the tool could solve the problem stated in Chapter 2 since the interviews and focus group show that the developed tool has the potential to help the EA practitioners on the co-evolution process.

In the next chapter, a conclusion of the research is presented by discussing the lessons learned, design and prototype limitations, and topics for future work, namely addressing the identified limitations.

7

Conclusion

Contents

This research presented a tool for visualization and management of EA meta-model evolution to address the issue of insufficient tool support regarding the co-evolution of the EA meta-model and respective model. The tool implements a set of features addressing the structural changes to the EA meta-model, impact analysis and migration of those changes to the meta-model and model on an EAM tool.

Two DSR iterations were done to the tool's design and implementation, as well as the respective demonstration and evaluation methods. The first DSR demonstration was a simple case of a transformation phase, in order to verify the objectivity of the tool, followed by an evaluation with user testing. The second DSR iteration was used as a field experiment in the scope of an EA project within the Portuguese Navy. Afterwards, we performed a DSR evaluation model assessing the tool's objective, efficacy, with focus groups and interviews, and usability with user tests.

7.1 Discussion

We identified the need for a solution that could mitigate the gap in the current EAM tool spectrum regarding visualization and management of EA transformations (see Chapter 3). This need also corroborates the challenge revolving the evolution of the EA meta-model, hence reinforcing the need for possible design solutions. Further interviews with practitioners clarified the identification of the following requirements as being of relevance to the design of the proposed tool:

- A solution should allow the saving of change's motivation;
- A solution should allow the migration of the model, after meta-model changes;
- A solution should describe change's impact;
- A solution should allow the undo and redo of EA meta-model operations;

Despite existing EAM tools offering EA meta-model adaptation, none apparently showed considerable concerns towards a feature set focused on EA evolution. Whether or not the change frequency of the EA meta-model can be considered regular in practice is arguable. Nonetheless, when changes to the EA meta-model occur, existing tools do not offer the necessary automation mechanisms to evolve the EA meta-model and conforming model accordingly. Overall, tools do not consider model migration scenarios. Hence, the affected EA model still needs to be manually updated to ensure model conformance. Therefore, EAM tools could benefit from an information model that addressed the evolutionary concepts concerning the evolution of the enterprise's structure over time, thus, overcoming some of the problems associated with EAM failure [37].

The tool's conceptual model not only considers the EA meta-model but also its respective model as two highly cohesive EA description artifacts. The model also expresses the concepts that, as analyzed in existing literature, describe enterprise transformation. Furthermore, due to the conceptual model's generic nature, is possible to incorporate any EA modeling language, thus promoting EA model design flexibility according to the organization's individual needs.

In contrast with Roth's four-layered conceptual design [4], the proposed tool (besides visualizing the changes done to the EA meta-model) represents the UI of an EAM evolution tool that aims to empower practitioners, by allowing them to perform changes in real time and oversee their impact.

Finally the evaluation outcome confirmed the achievement of the solution's objectives by enabling an EA meta-model evolution visualization environment in which editions to the EA meta-model could be done in an intuitive and timely manner and then propagated to the EA model.

7.2 Limitations

From the interviews and focus groups of second DSR iteration, some limitations of the tool were found:

1. Absence of indicators reflecting the impact of meta-model changes in the model makes it difficult to assess the degree of impact concerning meta-model changes;
2. Lack of filtering on the visualizations;
3. Lack of a resume and quantification of changes made to the EA meta-model;
4. The absence of the definition of mandatory properties and relations.

These limitations have no implications with the main objective of this research since the evaluations made reveal that the tool enables the co-evolution of EA meta-model and respective model.

7.3 Communication

In order to communicate the research, four papers have been submitted submitted, and two of them published in EMCIS 2017[1]:

- Rechau, Tiago, et al. "A Tool for Managing the Evolution of Enterprise Architecture Meta-model and Models." European, Mediterranean, and Middle Eastern Conference on Information Systems. Springer, Cham, 2017. [38];
- Silva, Nuno, et al. "Automation Process for Co-evolution of Enterprise Architecture Meta-Models and Models." European, Mediterranean, and Middle Eastern Conference on Information Systems. Springer, Cham, 2017. [6];

In these papers, the authors present a first approach prototype for co-evolution of EA meta-models and respective models, as well as, a process for Co-evolution of EA meta-models and models.

Another research paper has been submitted to EMISA[2] with the final version of the tool, focusing on usability and ease of use of the UI, and a last one submitted to ECIS 2018[3]. This last paper also presents the proposed tool but addressing the topic and solution from the industry perspective.

7.4 Future Work

Future efforts are required to create a fully functional and ready for production tool that should be implemented on an EAM tool. These efforts are mainly resolving the problems stated on the limitations Section 7.2, with focus on extend the impact analysis expressiveness. Afterwards, the integration of the following EA transformations stated on [8] and on [7] is also needed:

- Inline EA Class;
- Association to EA Class;
- EA Class to Association;
- Extract Class;
- Inline Class;
- Generalize Property;
- Pull Property;

[1] http://http://emcis.eu/
[2] https://www.emisa-journal.org/emisa
[3] http://ecis2018.eu/

- Extract Superclass;
- Restrict Property;
- Push Property;
- Flatten Hierarchy;

To finish, a total integration with the EAMS EAM tool is necessary.

Bibliography

[1] "Togaf version 9.1," http://pubs.opengroup.org/architecture/togaf9-doc/arch, (Accessed on 14/11/2016).

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

[3] S. Roth, M. Zec, and F. Matthes, "Enterprise architecture visualization tool survey 2014," sebis, Technische Universität München, Tech. Rep., 2014. [Online]. Available: https://wwwmatthes.in.tum.de/pages/o790x7rho1te/EAVTS-2014-Final-Report

[4] S. Roth and F. Matthes, "Visualizing differences of enterprise architecture models," in *Proc. International Workshop on Comparison and Versioning of Software Models at Software Engineering*, 2014.

[5] B. Gruschko, D. Kolovos, and R. Paige, "Towards synchronizing models with evolving metamodels," in *Proceedings of the International Workshop on Model-Driven Software Evolution*, 2007.

[6] N. Silva, T. Rechau, M. M. da Silva, and P. Sousa, "Automation process for co-evolution of enterprise architecture meta-models and models," in *Proceedings of the 14th European, Mediterranean, and Middle Eastern Conference*. Springer International Publishing, 2017, pp. 655–661.

[7] G. Wachsmuth, "Metamodel adaptation and model co-adaptation," in *European Conference on Object-Oriented Programming*, 2007, pp. 600–624.

[8] N. Silva, F. Ferreira, P. Sousa, and M. M. da Silva, "Automating the migration of enterprise architecture models," *International Journal of Information System Modeling and Design (IJISMD)*, vol. 7, no. 2, pp. 72–90, 2016.

[9] N. Prat, I. Comyn-Wattiau, and J. Akoka, "Artifact evaluation in information systems design-science research-a holistic view." in *PACIS*, 2014, p. 23.

[10] M. Lankhorst, *Enterprise architecture at work: modelling, communication, and analysis*. Springer, 2009.

[11] J. Tribolet, P. Sousa, and A. Caetano, "The role of enterprise governance and cartography in enterprise engineering." *Enterprise Modelling and Information Systems Architectures*, vol. 9, no. 1, pp. 38–49, 2014.

[12] S. Kaisler, F. Armour, and M. Valivullah, "Enterprise Architecting: Critical Problems," *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, vol. 00, no. C, pp. 1–10, 2005.

[13] F. Mantz, G. Taentzer, Y. Lamo, and U. Wolter, "Co-evolving meta-models and their instance models: A formal approach based on graph transformation," *Science of Computer Programming*, vol. 104, pp. 2–43, 2015.

[14] IEEE-SA Standards Board, "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems," *IEEE Std*, vol. 1471-2000, pp. 1–23, 2000.

[15] H. Jonkers, M. M. Lankhorst, H. W. L. Ter Doest, F. Arbab, H. Bosma, and R. J. Wieringa, "Enterprise architecture: Management tool and blueprint for the organisation," *Information Systems Frontiers*, vol. 8, no. 2, pp. 63–66, 2006.

[16] H. Shah and M. El Kourdi, "Frameworks for enterprise architecture," *It Professional*, vol. 9, no. 5, pp. 36–41, 2007.

[17] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[18] S. Lucke, Carsten; Krell and U. Lechner, "Critical issues in enterprise architecting – a literature review." AMCIS, 2010, p. 305.

[19] S. Buckl, A. M. Ernst, F. Matthes, and C. M. Schweda, "Visual roadmaps for managed enterprise architecture evolution," in *Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, 2009. 10th ACIS International Conference on*. IEEE, 2009, pp. 352–357.

[20] S. Aier, B. Gleichauf, J. Saat, and R. Winter, "Complexity levels of representing dynamics in ea planning," in *Advances in Enterprise Engineering III*. Springer, 2009, pp. 55–69.

[21] M. Schaub, F. Matthes, and S. Roth, "Towards a conceptual framework for interactive enterprise architecture management visualizations." in *Modellierung*. Citeseer, 2012, pp. 75–90.

[22] F. Matthes, S. Buckl, J. Leitel, and C. M. Schweda, *Enterprise Architecture Management Tool Survey 2008*, 2008, vol. 19.

[23] F. Matthes, M. Hauder, and N. Katinszky, “Enterprise Architecture Management Tool Survey 2014.”

[24] F. Ferreira, “An Interface for Visualizing the Evolution of Enterprise,” Master’s thesis, Técnico Lisboa, 2016.

[25] P. Sousa, J. Lima, A. Sampaio, and C. Pereira, “An approach for creating and managing enterprise blueprints: A case for it blueprints,” in *Advances in enterprise engineering III*. Springer, 2009, pp. 70–84.

[26] F. J. Armour, S. H. Kaisler, and S. Y. Liu, “Building an enterprise architecture step by step,” *IT professional*, vol. 1, no. 4, pp. 31–39, 1999.

[27] S. Buckl, A. Ernst, F. Matthes, and C. M. Schweda, “An information model for landscape management–discussing temporality aspects,” in *International Conference on Service-Oriented Computing*. Springer, 2008, pp. 363–374.

[28] J. A. Zachman, “A framework for information systems architecture,” *IBM systems journal*, vol. 38, no. 2/3, p. 454, 1999.

[29] J. Zachman, “The zachman framework for enterprise architecture,” *Zachman International*, vol. 79, 2002.

[30] M. Buschle, M. Ekstedt, S. Grunow, M. Hauder, F. Matthes, and S. Roth, “Automating enterprise architecture documentation using an enterprise service bus.” in *AMCIS*. Association for Information Systems, 2012.

[31] R. Bakelaar and E. Roubtsova, “Visualization of changes in archimate,” no. August, 2016.

[32] “Archimate® 2.1 specification,” http://pubs.opengroup.org/architecture/archimate2-doc/, (Accessed on 20/5/2017).

[33] U. Brandes and B. Köpf, “Fast and simple horizontal coordinate assignment,” in *Graph drawing*, vol. 2265. Springer, 2001, pp. 31–44.

[34] J. Brooke *et al.*, “Sus-a quick and dirty usability scale,” *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.

[35] A. Bangor, P. Kortum, and J. Miller, “Determining what individual sus scores mean: Adding an adjective rating scale,” *Journal of usability studies*, vol. 4, no. 3, pp. 114–123, 2009.

[36] A. Bangor, P. T. Kortum, and J. T. Miller, “An empirical evaluation of the system usability scale,” *Intl. Journal of Human–Computer Interaction*, vol. 24, no. 6, pp. 574–594, 2008.

[37] E. Denert-Stiftungslehrstuhl, “Enterprise architecture management pattern catalog,” 2008.

[38] T. Rechau, N. Silva, M. M. da Silva, and P. Sousa, “A tool for managing the evolution of enterprise architecture meta-model and models,” in *European, Mediterranean, and Middle Eastern Conference on Information Systems*. Springer, 2017, pp. 68–81.

A

Appendix A

Table A.1: Students Questionnaire results from First DSR evaluation

	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	User 12	User 13	User 14	User 15	User 16
The interface communicates in a simple way	6	6	6	6	6	6	6	7	6	7	6	6	7	5	5	6
The interface uses the same language as the user	4	6	7	7	7	5	7	7	6	4	6	5	7	4	5	6
How do you consider the user mental effort?	6	7	6	6	6	3	6	6	5	5	6	5	6	4	5	6
How do you consider the consistency of the interface?	7	6	7	6	4	5	7	7	4	6	6	7	7	3	4	5
How do you consider the help features presented on the interface?	6	6	5	6	5	2	7	5	6	4	3	4	5	4	4	3

B

Appendix B

Co-Evolução de Meta-modelos de modelos em Arquiteturas Empresariais

Testes com Utilizadores

Tomada de conhecimento

Eu, _____,
declaro que tomei conhecimento de que nos testes de utilizador para o estudo *A Tool for Managing the Co-Evolution of Enterprise Architecture Meta-models and Models* em que irei participar serão gravadas as minhas interacções com o sistema, bem como retirados comentários sobre a minha interacção. Declaro também que autorizo o uso desses dados para fins educativos e de investigação, sem mais nenhum fim alternativo fora dos mencionados.

(assinatura)

_____

Figure B.1: Consent form

C

Appendix C

Table C.1: System Usability Scale Questionnaire Results

Question/ # Practitioner Result	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
I think that I would like to use this system frequently	5	4	5	5	4	3	3	5	5	4	5	4	4	5	3	4	4	5	4	4	5	4	4	4
I found the system unnecessarily complex	2	3	1	1	1	2	2	4	4	1	2	2	1	2	2	1	2	1	5	2	1	1	2	1
I thought the system was easy to use	5	2	4	5	5	3	3	5	4	5	4	5	4	4	4	5	4	5	3	4	5	3	4	4
I think that I would need the support of a technical person to be able to use this system	1	2	1	2	2	2	3	1	1	1	2	1	2	2	1	2	2	1	2	3	2	2	1	1
I found the various functions in this system were well integrated	4	5	4	5	4	4	3	4	4	5	4	4	4	5	3	4	4	5	4	5	5	3	5	5
I thought there was too much inconsistency in this system	1	1	1	1	1	1	2	2	2	2	1	2	2	1	3	1	1	1	1	1	1	1	2	1
I would imagine that most people would learn to use this system very quickly	5	3	5	5	4	4	4	5	5	4	4	5	5	5	3	4	3	5	4	5	5	4	5	5
I found the system very cumbersome to use	1	3	1	1	2	2	2	1	3	1	1	3	2	1	2	1	2	1	3	1	1	1	2	1
I felt very confident using the system	5	3	3	4	4	3	3	5	3	4	5	4	4	4	4	4	3	4	4	3	5	4	4	4
I needed to learn a lot of things before I could get going with this system	1	2	1	1	1	4	2	1	2	2	2	1	2	2	1	1	2	2	3	2	2	1	2	1
Overall, I would rate the user-friendliness of this product as:	7	4	6	6	6	6	5	6	6	6	6	6	6	6	4	6	5	6	5	6	7	4	6	6
SUS Result	95	65	90	95	85	65	62,5	87,5	72,5	87,5	85	82,5	80	87,5	70	87,5	72,5	95	62,5	80	95	80	82,5	92,5