

An Interface for Visualizing the Evolution of Enterprise Meta-models

Francisco Miguel Grilate Ferreira
Instituto Superior Técnico
Universidade de Lisboa
Av. Rovisco Pais, 1049-001 Lisboa, Portugal
Email: francisco.g.ferreira@ist.utl.pt

Abstract—The practice of Enterprise Architecture (EA) has proven its utility in the alignment of business and IT. Managing its evolution when new requirements arise is not a trivial task, thus, threatening the success and efficiency of EA development projects. In this paper, the authors used Design-Science Research (DSR) to propose an interface that provides a consistent environment for visualizing the evolution of enterprise meta-models. The interface used as demonstration an EA evolution project from an European financial organization, and was validated using specific interface evaluation techniques.

Index Terms—Enterprise Architecture, EA Management, EA Development Project, EA Tools, EA Transformations, Interface

I. INTRODUCTION

Enterprise Architecture (EA) is defined as a coherent whole of principles, methods, and models used in the design and realization of an organization’s structure, business processes, information systems (IS) and infrastructure (Lankhorst, 0013). EAs aim to capture the essentials of the business and the IT using enterprise meta-models and models, and does so by consolidating architecture decisions, along with migration and implementation plans to address strategic aspects of an organization.

A. Research Motivation

An effective EA Management (EAM) follows a consistent baseline of information concerning the *as-is* landscape of the organization, as well as a planning process of the envisioned *to-be* state. The planning and implementation of the organization’s *to-be* landscape from its *as-is* is done within the scope of an EA development project. An EA development project is supported by three central tasks: *documenting* the current state of the organization’s architecture, *envisioning* the future perspective of the architecture, and *planning* different intermediate architecture scenarios deciding on an architecture *road-map* (Buckl et al., 2009).

The organization’s architecture is defined by an enterprise meta-model, specifying the information regarding the elements and relationships used to represent and/or model the organization (Buckl et al., 2007). Starting an EA development project with a simplified enterprise meta-model and enriching it, while growing within the organization, sounds appealing. However, addressing these projects implies that enterprise meta-models are supported as living artifacts evolving according to the organization’s needs (Tribolet et al., 2014).

There are a variety of EA tools that support numerous EAM tasks. These tools support the EA architect in the design of enterprise models as structured, object-oriented models reflecting the organization’s *as-is* landscape stored in EA repositories. Existing EA Tools features are complete when defining and representing the *as-is* landscape, allowing EA architects to choose one according to specific organizational needs.

B. Research Problem

During an EA development project, the planning and implementation of the *to-be* landscape is performed iteratively. Each iteration may raise a number of issues that result from uncoordinated, *ad-hoc* decisions concerning EA components and their inter-relationships. Issues like the duplication of efforts and resources, poor coordination and control, management and business performance issues, and inefficiencies in operation are the outcomes of such decisions (Kaisler et al., 2005). Consequently, the meta-models and concepts used to model the first *as-is* are no longer enough to model the current *as-is* of even the *to-be*.

The lack of support from EA tools in regard to EA evolution as an iterative process, is one of the causes leading to inefficiency during the conduction of EA development projects. Considering all the above, the authors identify the research problem as the **inefficient implementation of EA development projects**. To address the identified problem, an enterprise meta-model evolution interface was proposed, featuring a set of EA transformations as an innovative, purposeful IS artifact.

C. Research Methodology

The research methodology used was Design Science Research (DSR) (Von Alan et al., 2004). This methodology was chosen due to being appropriate to researches that seek to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts. DSR is also active with respect to technology, engaging in the creation of technological artifacts that impact people and organizations (Von Alan et al., 2004).

D. Paper Structure

This paper is structured as follows: The “Related Work” section, presents the motivation behind the research (Identify

Problem & Motivate) by presenting an overview and discussion on the *state-of-the-art* concerning the topics of enterprise transformation, EA tools, and meta-model evolution, as found in literature. In the "Research Proposal" section, the authors provide a detailed description of the developed interface, corresponding to DSRM's Design & Development phase. Then, an enterprise meta-model evolution scenario is illustrated in the "Demonstration" section, showing the application of the proposed artifact, under the scope of an European financial organization's EA development project. Next, the "Evaluation" section outlines a metric-based validation of the proposed artifact as well as user tests. Finally, the conclusion of this research (the interface's limitations and themes for future work) is presented in the "Conclusion" section.

II. RELATED WORK

In the next sub-sections, the authors present a literature review with reference to contributions on the topics of enterprise transformation, EA tools, and meta-model evolution, as well as a discussion of the conclusions drawn from the *state-of-the-art* analysis.

A. Enterprise 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 (Sousa et al., 2009; Tribolet et al., 2014). These initiatives are part of the EAM practice within organizations.

To address the changes within the organization's domain, a set of principles were proposed concerning the evolution of architectural views (Sousa et al., 2009; Tribolet et al., 2014). Keeping up-to-date architectural views with organizations changes has brought up issues in EA development projects, such as being time-consuming, error-prone, and even cause some misalignments between both business and IT spectrum, thus, proving the inefficiency of conducting such EA development projects. To mitigate this, rather than using a versioning schema to handle architectural views evolution, was 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.

An organizational artifact is conceived as the future result of an EA development project, thereby entering the gestating state. An artifact remains in gestation throughout the project's time-line becoming alive after the project ends. The artifact dies after the completion of a decommissioning project or when a transformation project is canceled. In the end, the dead artifact is then retired when a retirement project explicitly removes it from the organizational infrastructure.

A viewpoint was presented for *road-mapping* the development of EA over time, complementing it with a conceptual model explaining the information demands for such *road-map* plans (Buckl et al., 2009). Both artifacts were drawn from a pattern-based approach for EA management (Denert-Stiftungslehrstuhl, 2008), meaning both can be integrated

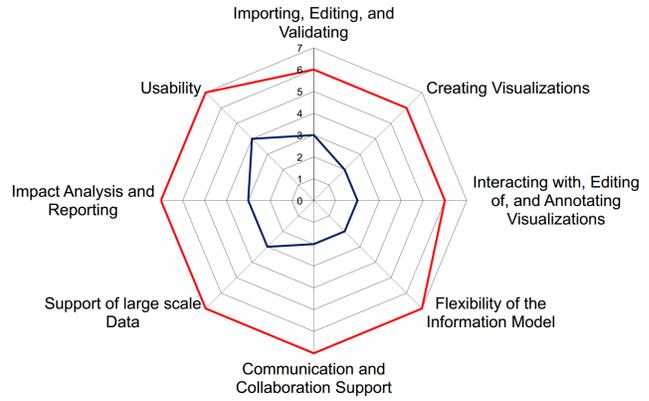


Fig. 1: Functional criteria spider diagram (from (Matthes et al., 2008))

into a multitude of organization-specific EA management approaches.

Also, a four-layered conceptual design of an interactive visualization to drill down and analyze model differences in EA meta-models and respective models was proposed (Roth and Matthes, 2014). The conclusions drawn were that enterprise models shared many similarities with software models with respect to complexity, the number of instances, etc. Therefore, an enterprise model could be perceived as a software model.

Another study identified the challenges of documenting EAs as being a manual, time-consuming and costly task (Buschle et al., 2012). 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.

B. EA Tools

In order to understand the actual state of progression regarding EA transformation support, analysis, and management within the EA industry, an assessment on the current EA Tool's features was made. (Matthes et al., 2008) performed an analysis to 9 EA tool, which were evaluated by 3 different teams. The evaluation process was based on the analysis of functional criteria and EAM task criteria. Figure 1 and Figure 2 illustrate the minimum and maximum achieved results from (Matthes et al., 2008) for each criteria, respectively. For each criteria, different test scenarios were created, complemented by on-line questionnaires. The findings were then processed and presented in spider diagrams, according to their ratings for each task. The ratings range from 0 to 7, being 7 the highest score possible. Later on, (Matthes et al., 2014) complemented the EA tool analysis, using the same evaluation process, by evaluating 4 additional EA tools.

Their functional criteria analysis concluded that improvements can be done in "Creating visualizations", since scenario generation is limited and flexible models are not supported. Moreover, "Importing, Editing, and Validating" has no stan-

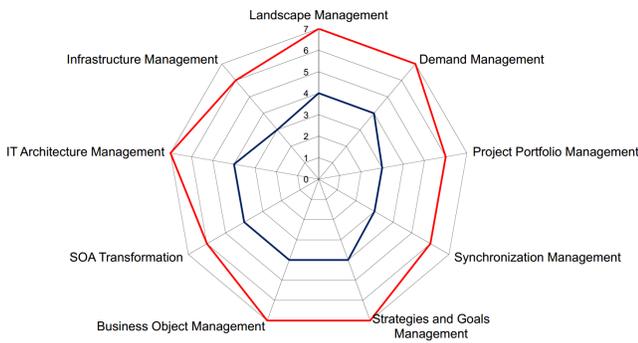


Fig. 2: Task criteria spider diagram (from (Matthes et al., 2008))

standard exchange format for enterprise models, and a common terminology to define information models is missing. Improvements can also be made regarding "Interacting with Editing of and Annotating Visualizations", "Editing of, and Annotating Visualizations" since semantic changes could provide an improvement towards graphical modeling.

More specifically, in the context of enterprise meta-model evolution, the functional criteria also assessed the "flexibility of information model". From the 13 analyzed tools, only 7 supported features to change or adapt the information model. From these 7 tools, only 6 were rated according to their functional support. The analysis showed that 4 tools scored above 5, incorporating a complete solution to manage the information model. Although most EA tools already show concerns about the information model flexibility, none of the tools mentioned any considerations regarding the migration of the enterprise models affected by those changes.

The task analysis showed that the tasks in "Landscape Management" could still be improved in EA Tools. More specifically, versioning application landscapes retains potential for improvement, and not all tools provide methods for deriving the planned landscape from the planned project portfolio. "Synchronization Management" is also limited, as no tool directly supports the concept of project delay. "SOA Transformation Management" lacks tool support for identifying services, and in "Infrastructure Management" not all tools provide concepts for life-cycle aspects of infrastructure components.

C. Meta-model Evolution

In the Model-Driven Development (MDD) area, software applications can be described through a set of models using Domain-Specific Modeling Languages (DSMLs). These DSMLs are subject to change given a set of changing requirements. DSMLs are defined by meta-models, i.e., models that define the structure of a modeling language used to describe a model (da Silva, 2015).

A formal framework (Mantz et al., 2015) based on graph transformations (Biermann et al., 2012; K Ehrig and G Taentzer, 2006) was proposed to model co-evolution of

meta-models and models. The research states that meta-model evolutions and migrations of their instance models are specified by coupled graph transformations.

Moreover, an evolutionary method was established to automate and secure modeling language evolution (Herrmannsdörfer et al., 2011). This method records the adaptations to the meta-model as coupled operations in a history model, which can later be used to automatically migrate models conforming to the meta-model.

Lastly, a transformational setting was defined for stepwise meta-model adaptation, in which each meta-model transformation implements a typical adaptation step performed manually (Wachsmuth, 2007). For each transformation, corresponding instant co-transformations (co-adaptations) of the meta-model instances are presented, always conforming to the current version of the meta-model. The latter approach was adapted to the Enterprise Engineering area of research, more specifically, to the subject of enterprise transformation (Silva et al., 2016). A set of EA migration rules were proposed in order to mitigate time and coherence issues revolving around the migration of enterprise models when changes were made within the organization's structure, i.e., enterprise meta-model changes.

D. Discussion

Based on the analyzed literature, the authors identified the need for an innovative artifact that enables the visualization and management of EA transformations over time, in a holistic manner.

After the EA tool analysis, some limitations were identified regarding the flexibility of information models in EAM. Most of current EA tools already address the concern of adapting the EA information model to the organization's specific needs, but do not comprise model migration scenarios. After performing changes to the information model, the affected enterprise models still need to be manually updated to ensure the information remains coherent. This concern still poses a limitation when addressing EA as an iterative process evolves over time. Therefore, proving the need for an innovative artifact that addresses these issues.

Integrating such artifact into an EA tool requires the specification and implementation of a structured EAM program. By coordinating the EAM program with the organization's needs as a continuous iterative process, a coherent informational environment can be reached. To achieve this alignment, an EA evolution meta-meta-model (see Section III-B) would give a better understanding of the evolutionary concepts (and relationships) that address the evolution of the organization's structure over time, thus, overcoming some of the problems associated to EAM failure (Denert-Stiftungslehrstuhl, 2008).

The EA evolution meta-meta-model incorporates any EA framework or notation, ensuring the EAM program success, as it can be adapted to each organization's individual needs. The considered meta-meta-model is aligned with Buckl's information model concepts to manage EA transformation (Buckl et al., 2009), representing an EA development project as a set of EA Transformations.

Finally, Roth and Matthes four-layered conceptual design to visualize model differences in meta-models accomplishes some of the authors proposed interface goals. Despite being thorough, the layered visualization has some limitations. The user can only visualize changes, in contrast to the interface the author proposes which aims to empower stakeholders, allowing them to perform changes in real time and oversee their impact. Another disadvantage is that the drill-down layered style is rather complex for everyday use, requiring navigation and filtering to visualize the changes. Hence, since some of the features already exist in current EA Tools, the proposed interface uses a single view to compare the differences between two meta-models (from the *as-is* to the *to-be* state).

III. RESEARCH PROPOSAL

To address the research problem (see Section I-B), the authors proposed an enterprise meta-model evolution interface as the outcome of the research Design & Development phase. First, *Objectives of a Solution* describes the interface's objectives. Then, the authors introduce an *EA evolution meta-meta-model* used as the interface's data schema, hence, defining the concepts and relationships for evolving enterprise meta-models. Finally, a detailed design specification of the proposed interface is given, in which some of (Silva et al., 2016) EA migration transformations were adapted.

A. Objectives of a Solution

The proposed interface aims to accomplish the following objective: **provide stakeholders with a consistent environment to visualize the evolution of the enterprise meta-model**, thus turning EA development projects more efficient.

B. EA Evolution Meta-meta-model

The EA evolution meta-meta-model, illustrated in Figure 3, defines a set of concepts and relationships used by the proposed interface.

An enterprise meta-model can be composed of any number of *EA Classes*, ranging from 1 to n , n being a finite integer. An EA Class has a variable list of *EA Properties*, each property being of a given *EA Type*. Enterprise models (instances of the enterprise meta-model) are composed of *EA Objects*, also ranging from 1 to n (n being also a finite integer), being instantiations of EA Classes that bound EA Properties to what each EA Object may hold at a given moment.

EA Values are always assigned to a pair \langle EA Object, EA Property \rangle and represent the actual value of an EA property from a given EA object at a given point in time. EA Properties can hold sets of EA Values, in particular, EA Properties of EA Type. Reference is a set of references to EA Objects of different EA Classes. The life-cycle states and their respective dates are kept for each concept.

Some of the concepts specifying the evolution of enterprise meta-models have a direct relation with the life-cycle states of each EA element (EA Class, EA Property, EA Object, etc.),

i.e., they influence the current state of a specific EA element. Each evolution concept also has its own life-cycle states.

An *EA Operation* represents a change made to the enterprise meta-model from a pre-defined set of operations. EA Operation defines a single transformation operation; for example, introducing an EA Property, eliminating an EA Class, changing an EA Property type, etc. An *EA Transformation* is composed of one or more EA Operations.

Since organizations often require several transformations to be applied, the concept of *EA Project* was incorporated. Each EA Transformation is related to a specific business requirement, thus, *EA Requirement* incorporates the reasoning behind any EA Transformation. Each EA Transformation is then associated to either an EA Class or EA property.

C. The Interface

The proposed interface addresses the management of EA evolution by using a set of EA transformations that alter the current state of an organization's meta-model. Those transformations are composed of specific operations that change the life-cycle stages of the meta-model elements: classes and properties. Afterwards, the impact of those modifications can be seen in the meta-model's viewpoint.

The interface was implemented within a vendor-specific EA tool. First, the EA evolution meta-meta-model was incorporated into the tool, creating a data type for each element, with the required properties. Each data type instantiation depicts a meta-model component. Figure 4 illustrates the proposed interface.

The interface allows creating, deleting and saving EA Projects, thus allowing the representation of different scenarios regarding the organization's future landscape. Each project is associated to EA Transformations and EA Requirements, presented in two data grid containers.

Each EA Transformation is associated to one of six EA Operations (Create EA Class, Create EA Property, Update EA Class, Update EA Property, Remove EA Class and Remove EA Property). Those operations are represented in the Point Chart (see Figure 5) at the top of the interface. The Point Chart lays out an overview of the number of EA Transformations that will be applied to the enterprise meta-model in a pre-determined date. The x-axis represents each transformation type, whereas the y-axis determines the date when each operation will take place.

Each data point is associated to a color, illustrating its purpose when applied to an EA Class or EA Property, i.e. red color means removal, green corresponds to creation and orange signifies an update. The data point's size pictures the number of EA Transformations with the same operation type that will occur on the same date.

In the right bottom, a 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 6) is dynamic, since adding or removing

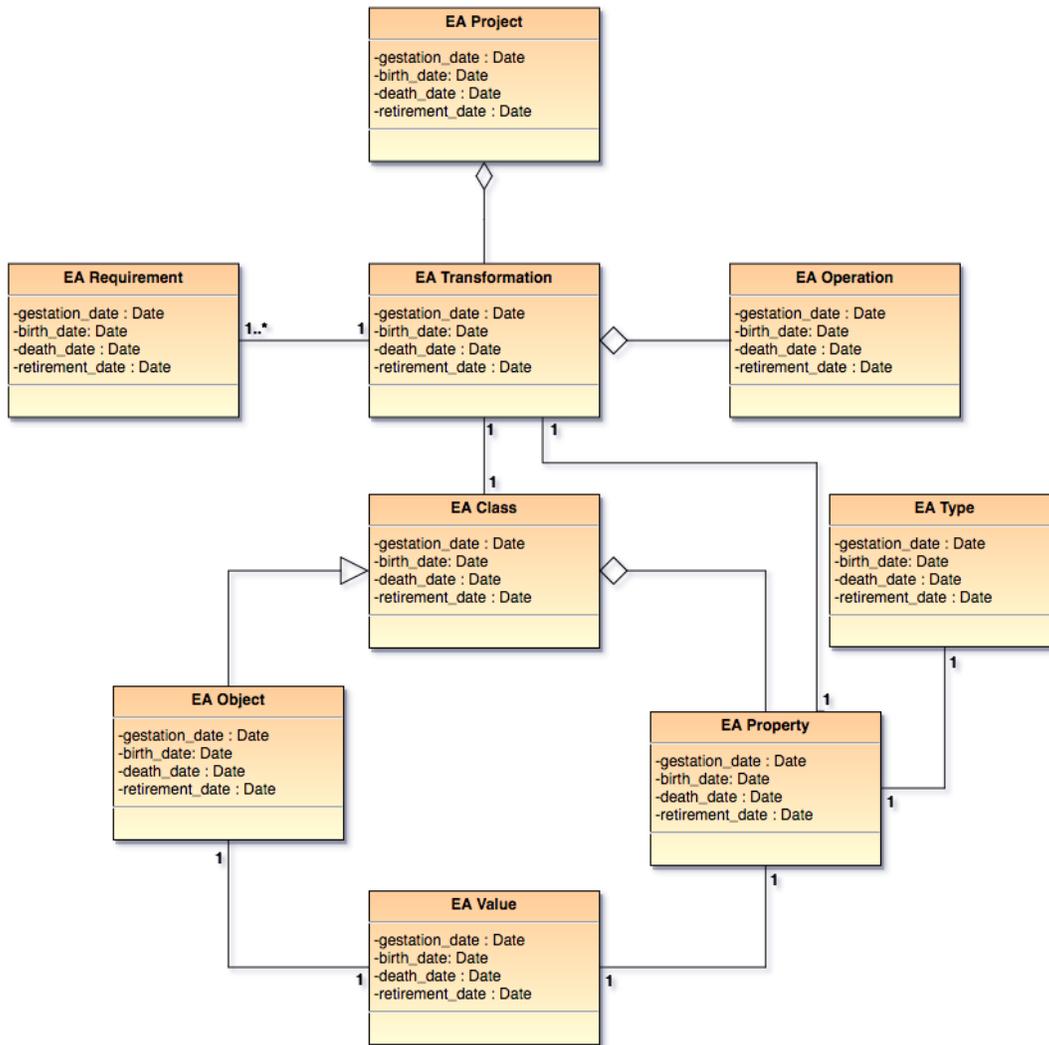


Fig. 3: The EA evolution meta-meta-model

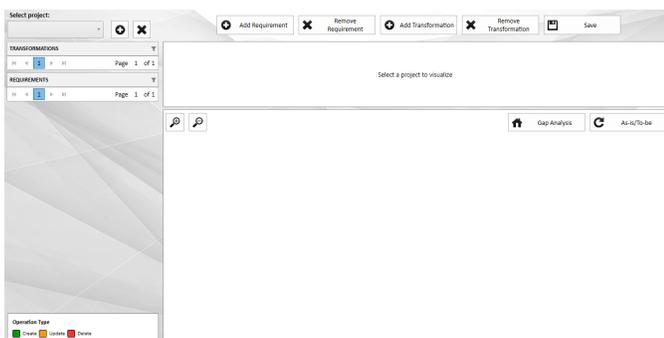


Fig. 4: The enterprise meta-model evolution interface (empty view)



Fig. 5: The interface's point chart

EA Transformations will update the viewpoint, displaying the impact of new transformations in real time.

The component's color is related to the EA Transformation operation type, using the same color scheme as described above. The viewpoint is also navigable. For example, a double click navigates to the class properties viewpoint that compares the *as-is* and *to-be* states. To add a different perspective to the comparison between the two states, the button "As-is/To-be" changes the side-by-side correlation to a single view, acting as a quick toggle button between them. The "Gap analysis" button redirects to the side-by-side view.

All interface components are related amongst them. By selecting one component's object, the affected objects in other components are also highlighted. This feature provides higher awareness to the stakeholders, making the interface usage and information analysis easier.



Fig. 6: The interface's enterprise meta-model viewpoint

IV. DEMONSTRATION

This section covers the Demonstration phase of the research method. After specifying and implementing the interface, the authors applied it in an EA development project of an European financial organization. The project's goal was to plan a first version of the organization's meta-model and from there iterate to newer versions according to specific requirements. The interface used the first version of the organization's meta-model as input for the set of EA transformations that were applied with respect to each requirement. Table I presents the number of EA transformations created by operation type that address each of the project's requirements.

EA Operations	EA Transformations
Create EA Class	1
Create EA Property	216
Update EA Class	-
Update EA Property	2
Delete EA Class	-
Delete EA Property	103

TABLE I. EA transformations by operation type (First iteration)

The EA transformations were handed out in an XML file input, derived from the gap between the original meta-model and its first iteration, using the tool's "Import data" feature. Then, the EA transformation and EA requirement instances were associated to the organization's project.

In the first iteration there was a need to redefine the classes' domain, by creating and removing properties according to the organization's needs. The viewpoint, depicted in Figure 7, clarifies the impact of the EA transformations to the meta-model.

The operation type can be deduced by analyzing the highlights on each meta-model component. The class highlighted in green (Usage Agreement) was added to the meta-model. All other classes are highlighted in orange, denoting changes in their properties. Correlating the Point Chart information

with changes visualized in the meta-model, the architect can conclude that the majority of these operations are "Create Property" and "Remove Property". This inference can be verified in Figure 8, displaying as an example the "Business Process" class properties viewpoint.

The EA project's second iteration changed the meta-model incorporating new requirements, derived from the organization's feedback after analyzing the organization's meta-model from the first iteration. The EA operations, depicted in Table II, were applied with resource to the interface's management features.

EA Operations	EA Transformations	
	First Iteration	Second Iteration
Create EA Class	1	-
Create EA Property	215	41
Update EA Class	-	-
Update EA Property	2	6
Delete EA Class	-	-
Delete EA Property	100	25

TABLE II. EA transformations by operation type (Second iteration)

Table II also represents the changes made to the first iteration. Besides adding or removing new EA transformations, several EA transformations were also undone. Figure 9 illustrates the interface after the second iteration's transformations were added. Each transformation and requirement is presented in the left-side data grids. The two iterations can be compared in the Point Chart by EA operation type, as Figure 10 shows. The data point's size shows the difference in the EA transformation's number by operation type. The operations "Create Property" and "Remove Property" comprise almost all of the required EA transformations. The meta-model's viewpoint did not suffer any changes from the first to the second iteration. This happened due to all executed operations affecting the existent classes' properties, not creating, updating or removing any classes.

The organization's meta-model became more complete and coherent in this second iteration, evolving alongside the organization's needs, as the Business Process properties viewpoint demonstrates in Figure 11. With the interface support, the required changes, in each iteration, to meta-model were fully represented, accomplishing a visual representation for the transformations performed, comprising an overview of the operations applied, and the reasoning behind them, defining the project's EA evolution.

The following section describes the evaluation phase of the research methodology, in which the authors validate the interface in order to assess the achievement of the solution's objective.

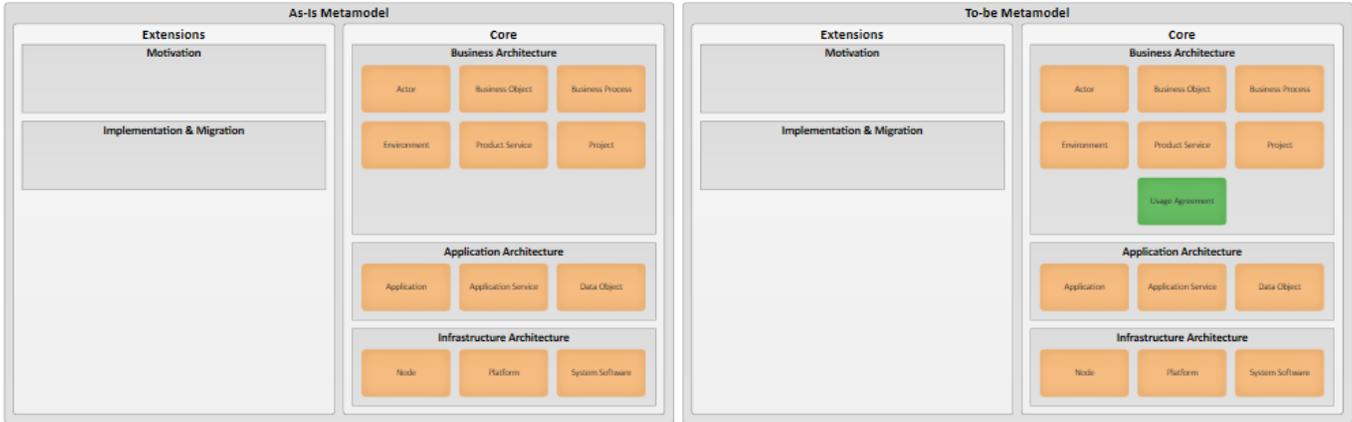


Fig. 7: Organization X meta-model viewpoint



Fig. 8: Business Process class properties viewpoint (first iteration)

V. EVALUATION

In order to validate the interface, a generic evaluation model for IS artifacts and evaluation criteria was used (Prat et al., 2014). Several evaluation criteria were chosen, taking into account different evaluation contexts. As an evaluation of an instantiation artifact, an analysis on the goal (efficacy), environment consistency with people (utility/ease of use), and environment consistency with the organization (utility) was made. Using the generic evaluation model, the following generic methods to evaluate the criteria were instantiated (see Table III).

From the interface’s demonstration, the authors concluded the artifact achieved the desired goal, providing a coherent visualization environment of the organization’s meta-model evolution through time. The contribution to better communicate relevant meta-model information changes between stakeholders verifies the goal efficacy, as exemplified with the

interface demonstration using a real organization (see Section IV).

To measure the quality of the artifact in practical use, the interface performance was measured with user tests. A sample of 20 students, researchers and practitioners was used, performing 16 tasks that validate the interface’s usability. The tasks incorporated the creation of requirements and different types of transformations, as well as exploring the interface using the supported Information Visualization features.

The average time to perform all tasks was 15 minutes and 5 seconds, the lowest time being 11 minutes and 37 seconds and the highest 17 minutes and 43 seconds. Similar task’s time reduced exponentially after the completion of the initial tasks. No user failed to perform any task. From these results, we can assess the utility of the interface regarding the consistency with people in the environment.

After the tests, the users answered a small questionnaire to

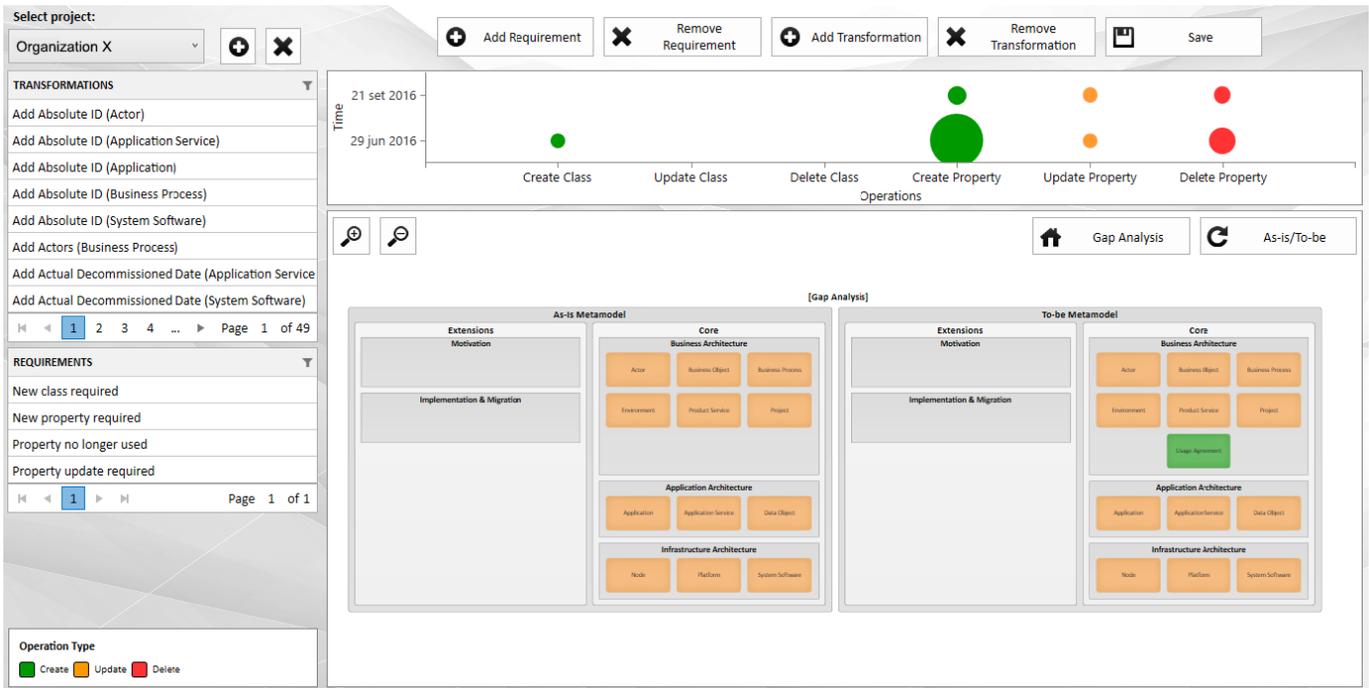


Fig. 9: Interface with Organization X EA development project

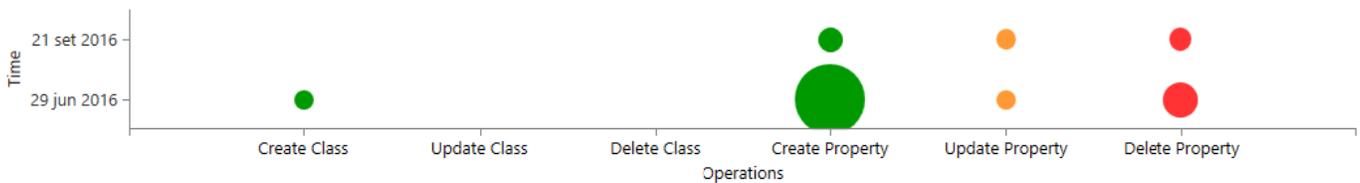


Fig. 10: Organization X EA development project point chart)

analyze the interface's usability. The questionnaire consisted of five questions, using a scale of 1 to 5, with 1 meaning that the user strongly disagrees and 5 meaning he/she strongly agrees with the statement.

The users understood the interface's goal (45% agree and 50% strongly agreed), found the interface ease to use (65% agreed and 30% strongly agreed), acknowledged that the features are well integrated in the interface (65% agreed and 30% strongly agreed) and disagreed that support was required from someone specialized to use the interface (25% strongly disagreed and 45% disagreed). From the feedback provided it is possible to validate the ease of use relative to the consistency with people in the environment.

In regard to the artifact consistency with the organization, the artifact introduces a missing feature to the organization's EA Tool, which improves the organization's EAM capabilities, verifying the artifact utility. The users agreed with that the interface would provide value for organizations (50% agreed and 45% strongly agreed), corroborating the author's statement.

Hence, the interface successfully met the research objective defined in III-A.

VI. CONCLUSION

In this paper, an EA meta-model evolution interface was proposed, using DSR, to address the lack of support in visualizing and managing changes within organizations. The proposed interface provides a link between the requirements motivating structural changes, the EA development projects that implement those changes, and the organizational assets defining the organization's structure.

Afterwards, a demonstration of the interface's usability was performed, under the scope of an EA development project, inside an European financial organization. The demonstration was followed by a user validation ranging from EA practitioners to students. The interface proved to enable a consistent EA evolution visualization environment of the organization's meta-model, hence, achieving the research objective.

A. Interface Limitations

The author's identified some limitations of the interface during the demonstration and evaluation phases. Due to the organization's meta-model recent development, the number of classes was limited; therefore, the demonstrated evolution scenario did not impact many classes with respect to creation

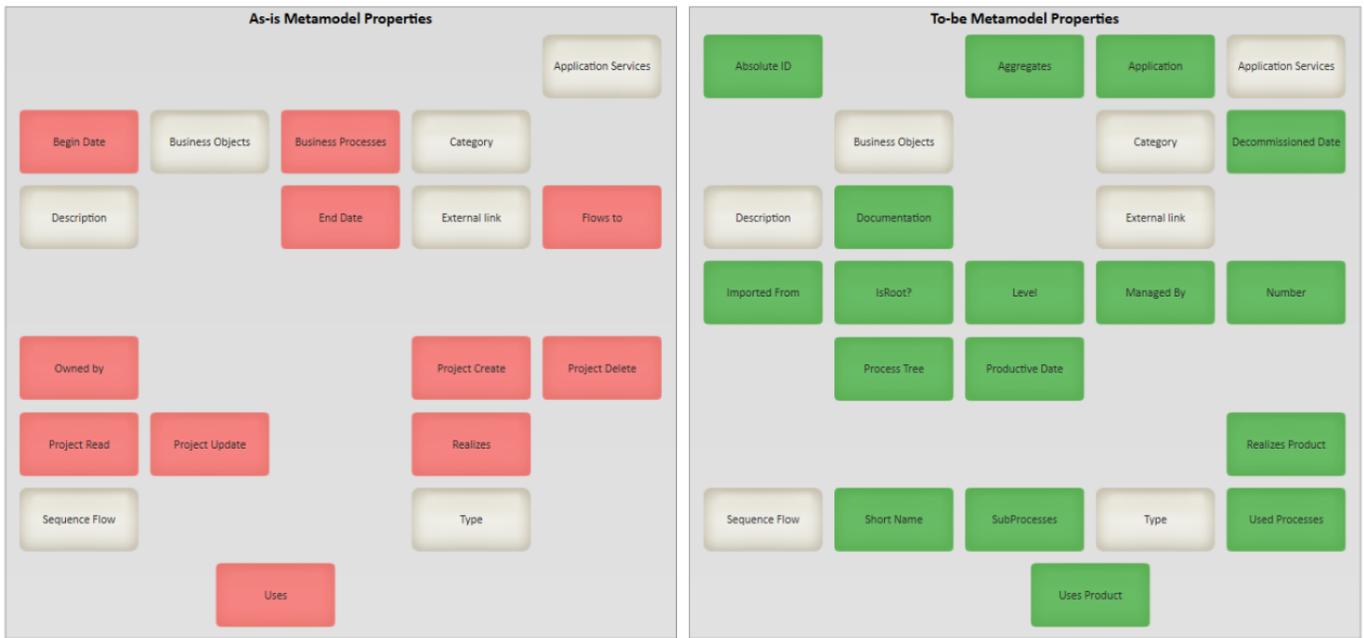


Fig. 11: Business Process class properties viewpoint (second iteration)

Description	Assessed criterion	Form of evaluation	Secondary Participants	Level of evaluation	Relativeness of Evaluation
Demonstration of the use of the artifact with a real example	Goal / Efficacy	Analysis and logical reasoning		Instantiation Real example	Absolute
Measurement of the performance of the artifact with user tests	Environment / Consistency with people / Utility	Quantitative	Students Practitioners Researchers	Instantiation	Absolute
Qualitative feedback after the user tests	Environment / Consistency with people / Ease of use	Qualitative	Students Practitioners Researchers	Instantiation	Relative to absence of artifact
Qualitative feedback after the user tests	Environment / Consistency with the organization / Utility	Qualitative	Students Practitioners Researchers	Instantiation	Relative to absence of artifact

TABLE III. IS artifact evaluation model instances (adapted from (Prat et al., 2014))

and removal. Furthermore, the absence of indicators reflecting the impact of meta-model changes in its instances (enterprise models) makes it difficult to assess the impact's degree of those changes. Also, not having the possibility of changing the meta-model to a previous state makes the effort of recovering previous meta-model versions higher. Finally, the application of the EA Transformations, defined using the interface during an EA development project, to the EA instances in the repository has to be done with another module in the EA tool.

B. Future Work

For future work more features will be considered to address the above-mentioned limitations, namely impact indicators of EA transformations with respect to enterprise models during the migration phase, as well as a roll-back feature concerning both the enterprise meta-model and enterprise models, and

to incorporate the ability to migrate the EA instances in the interface. Finally, more EA development project scenarios will be conducted to further validate the interface.

REFERENCES

- Biermann, E., Ermel, C., and Taentzer, G. (2012). Formal foundation of consistent emf model transformations by algebraic graph transformation. *Software & Systems Modeling*, 11(2):227–250.
- Buckl, S., Ernst, A. M., Lankes, J., Schneider, K., and Schweda, C. M. (2007). A pattern based approach for constructing enterprise architecture management information models. *Wirtschaftsinformatik Proceedings 2007*, page 65.
- Buckl, S., Ernst, A. M., Matthes, F., and Schweda, C. M. (2009). Visual roadmaps for managed enterprise architecture evolution. In *Software Engineering, Artificial Intel-*

- ligences, *Networking and Parallel/Distributed Computing, 2009. SNPD'09. 10th ACIS International Conference on*, pages 352–357. IEEE.
- Buschle, M., Ekstedt, M., Grunow, S., Hauder, M., Matthes, F., and Roth, S. (2012). Automating enterprise architecture documentation using an enterprise service bus.
- da Silva, A. R. (2015). Model-driven engineering: A survey supported by the unified conceptual model. *Computer Languages, Systems & Structures*, 43:139–155.
- Denert-Stiftungslehrstuhl, E. (2008). Enterprise architecture management pattern catalog.
- Herrmannsdörfer, M. et al. (2011). *Evolutionary metamodeling*. PhD thesis, Technische Universität München.
- K Ehrig, H. E. and G Taentzer, U. P. (2006). Fundamentals of algebraic graph transformation. with 41 figures (monographs in theoretical computer science. an eatcs series).
- Kaisler, S. H., Armour, F., and Valivullah, M. (2005). Enterprise architecting: Critical problems. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 224b–224b. IEEE.
- Lankhorst, M. (20013). Enterprise architecture at work: Modelling, communication and analysis (the enterprise engineering series).
- Mantz, F., Taentzer, G., Lamo, Y., and Wolter, U. (2015). Co-evolving meta-models and their instance models: a formal approach based on graph transformation. *Science of Computer Programming*, 104:2–43.
- Matthes, F., Buckl, S., Leitel, J., and Schweda, C. M. (2008). *Enterprise architecture management tool survey 2008*. Techn. Univ. München.
- Matthes, F., Nikolaus, K., and Matheus, H. (2014). *Enterprise architecture management tool survey 2014 update*. Techn. Univ. München.
- Prat, N., Comyn-Wattiau, I., and Akoka, J. (2014). Artifact evaluation in information systems design-science research-a holistic view. In *PACIS*, page 23. Citeseer.
- Roth, S. and Matthes, F. (2014). Visualizing differences of enterprise architecture models. In *Proc. International Workshop on Comparison and Versioning of Software Models (CVSM) at Software Engineering (SE)*.
- Silva, N., Ferreira, F., Sousa, P., and da Silva, M. M. (2016). Automating the migration of enterprise architecture models. *International Journal of Information System Modeling and Design (IJISMD)*, 7(2):72–90.
- Sousa, P., Lima, J., Sampaio, A., and Pereira, C. (2009). An approach for creating and managing enterprise blueprints: A case for it blueprints. In *Advances in enterprise engineering III*, pages 70–84. Springer.
- Tribolet, J., Sousa, P. M. A., and Caetano, A. (2014). The role of enterprise governance and cartography in enterprise engineering. *Enterprise Modelling and Information Systems Architectures*, 9(1):38–49.
- Von Alan, R. H., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 28(1):75–105.
- Wachsmuth, G. (2007). Metamodel adaptation and model co-adaptation. In *European Conference on Object-Oriented Programming*, pages 600–624. Springer.