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

Tiago Rechau
Técno Lisboa
Universidade de Lisboa
Lisbon, Portugal
tiago.rechau@tecnico.ulisboa.pt

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 Enterprise Architecture Management (EAM) 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 metamodel 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.

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

INTRODUCTION
Enterprises are complex adaptive socio-technical systems that continuously consider and pursue fundamental changes to maintain or gain competitive advantage [21].

The Enterprise Architecture (EA) discipline defines 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 [11]. EA helps to address the strategic aspects of the enterprise by consolidating architecture decisions along with migration and implementation plans.

EA Models capture and share the necessary knowledge to the development of a system [20]. A model captures the concepts and relationships within a given domain and allows reducing and mastering the complexity of a related system design, enhancing the communication among its stakeholders [17]. Hence, EA models are used to express, through diagrammatic descriptions, the essentials regarding the organization, business, application, and IT domains of the enterprise.

The EA model must conform to an EA meta-model, which specifies the concepts, relation types, and relations between concepts used to structure and represent the enterprise in all its domains [6]. Furthermore, EA meta-models are perceived as mutable artefacts that change according to the needs of the enterprise [28].

RESEARCH PROBLEM
EA projects define and implement the strategies that guide the enterprise in its evolution [30]. An issue that compromises the success rate of such projects is the inherent complexity of EA models [30], thus making the task of reading and updating these models also complex [26]. Many of today’s enterprises struggle with transformation management from a baseline architecture to a target architecture via intermediary planned architectures [7]. Several causes regarding this issue were identified [1], such as missing practical methodologies for architecture road-mapping, inadequate representation of the concept of time in EA models and insufficient tool support for architecture planning. Furthermore, Lucke et al. have also identified and categorized several issues in the EA process retrieved from the body of research, in particular [12]:

- Complexity - the large number of application’s in today’s organizations and the existing dependencies between the different layers and segments portrayed in architectural descriptions raises the complexity of EA endeavors;
- Rapidly changing conditions - due to the rapidly changing conditions in both technology and business, developing an EA becomes a rather effortful process. Architects have to deal with high dynamics and constraints caused by different life-cycle phases of systems and applications;
- Insufficient tool support - unsatisfying tool support is described in the literature as a general issue. Many of the existing tools are oriented towards the modelling of software architectures and do not address the representation of the system dynamics.

Existing EAM tools provide EA model visualizations and support strategic decisions concerning the enterprise [22]. Nevertheless, despite the multitude of available tools in the market (Troux, Aris, Enterprise Architect, System Architect, ABACUS, etc.), few tools address EA meta-model evolution containing the past, present, and future states of the EA [14, 15, 20]. This insufficient tool support for dealing with the rapidly
changing conditions, resulting in the evolution of EA meta-models, hinders the success rate of EA projects.

Overall, the problem of this research is the lack of support from EAM tools for co-evolution process of the EA meta-model and conforming model.

RELATED WORK
In the next sub-sections, we explore the body of research regarding EA evolution, meta-model evolution, EAM tools, and visualization of changes in EA. A database-driven literature review was conducted using the AIS Electronic Library (AISeL) and Google Scholar with search terms as “enterprise architecture” and “evolution” or “maintenance” or “changes” or “meta-model” or “model” or “tools” were used in both databases. Mainly IS journals and conference proceedings with focus on enterprise architecture and model-driven development domains were considered from the search results. Recent approaches (dating from 2012) were prioritised in detriment of older ones; nonetheless, due to the somewhat scarce body of knowledge referencing the domain of research, the authors opted for also consider older approaches (before 2012).

EA Evolution
The evolution of EA is an iterative and incremental process [2, 27]. Throughout the EA process, one or more transition architectures are defined. A transition architecture is the EA state between the baseline and target architectures. These transition architectures are planned using road-maps, which are abstracted plans for business or technology change, typically operating across various disciplines and over multiple years [27]. Architecture road-maps are used to describe the transformation path, i.e. transition architectures, over a period of time, from the baseline architecture to the desired target architecture. Sousa et al. considered the evolution of EA models as a means to address change within the organization’s domain [26]. In their approach, an enterprise is defined as a graph G of artefacts 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 artefacts. A given artefact is represented on a given blueprint if graph G holds as a relation between them"; and
- **Project**: "whose instances contain references to artefacts related to the project".

A state of existence regarding all artefacts is also defined for all artefacts 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 artefact other than blueprints";
- **Alive**: "if it is related with other artefacts in the alive state. This means that it may act upon other artefacts 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.

Meta-model Evolution
In the model-driven area of research, software applications can be described through a set of models using Domain-Specific Modelling 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 modelling language used to describe a model [23].

A formal framework [13] based on graph transformations 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 modelling language evolution [10]. This method records the adaptations to the meta-model as coupled operations in a history model, which can later be used to migrate models automatically con-forming to the meta-model.

Gruschko et al. presented an approach [9] that applies transformations to migrate a model when changes occur to the meta-model. Their approach is based on the Meta Object Facility (MOF) meta-modeling architecture. MOF introduces a four-layer division, where each layer represents a different level of abstraction. The first layer, M3, represents the meta-meta-model, used to de-fine M2, the second layer. M2 describes models that are instances of M3. M2 models are identified as meta-models. The third layer, M1, describes instances of M2 and finally, M0 represents instances of M1.

The focus of their approach is the migration of M1 models when M2 changes and is defined as a 5-step method.

The first step (Change detection or tracing) is comprised of two options. The first option is the direct comparison between two different versions of M2 models, in which the output will be the differences between them. On the other hand, the second option traces the differences, i.e., takes an older version of the M2 model and a tracing of change operations as input. This trace of changes operations represents the operations that allow the older version of an M2 model to get to the desired M2 model version.

The second step, Classification, refers to the classification of the detect changes in step 1. The changes were classified into three groups: 1) Not Breaking Changes; 2) Breaking and Resolvable Changes; 3) Breaking and Unresolvable Changes. Gruschko argues that the first group is not relevant since these changes do not change the instances of M2 models (M1), thus 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 to execute the propagation is needed.

The third step (User input gathering) execution depends on the type of changes. If there isn’t any change of "Breaking and unresolvable changes" type, this step is not required.

The fourth step, Algorithms Determination, generates the necessary changes to migrate, i.e., creates a list of necessary steps to turn the old M1 models into 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.

Lastly, a transformational setting was defined for stepwise meta-model adaptation, in which each meta-model transformation implements a typical adaptation step performed manually [29]. For each transformation, corresponding instant co-transformations 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 EA transformation [24]. A set of EA migration rules were proposed to mitigate time and coherence issues revolting around the migration of enterprise models when changes were made within the organization’s structure, i.e., EA meta-model changes.

Another approach to EA meta-model evolution process is presented by Silva et al [25]. 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 catalogue to be applied to one or more EA meta-model elements. These operations can be of a constructive, destructive, or refactoring nature [24]. The Propagate Change activity "sup-ports 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 [25]. The Check Conformance activity provides a comparison between the newest version of the EA me-ta-model and the current EA model version by assessing the model’s compliance with the new meta-model version. If inconsistencies are found, the Repair Model activity repairs the EA mod-el, thus 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.

**EAM Tools**

Matthes create a report of review and analysis made to nine EAM tool evaluated by three different teams [14]. The evaluation process was based on the analysis of functional criteria and EA management task criteria. For each criterion, different test scenarios were created, complemented by online questionnaires. Later the same analysis was complemented, using the same evaluation process, with four additional EAM tools [15].

The functional criteria analysis concluded that improvements could be made in "Creating visualizations" since scenario generation is limited and flexible models are not supported. Moreover, "Importing, Editing, and Validating" has no standard exchange format for enterprise models, and a common terminology to define information models is missing. Improvements could also be made regarding "Interacting with Editing of and Annotating visualizations" since semantic changes could provide an improvement towards graphical modelling.

In the context of EA meta-model evolution, the functional criteria also assessed the flexibility of meta-model. From the thirteen analyzed tools, seven supported features to change or adapt the meta-model. From these seven tools, six were rated according to their functional support. The analysis showed that four tools incorporate a solution to manage the meta-model. Nevertheless, despite most EAM tools showing concerns regarding the meta-model flexibility, none mentioned any considerations towards the migration of the EA model affected by those changes.

The task analysis showed that the tasks in "Landscape Management" could still be improved. In particular, versioning application landscapes retain the potential for improvement, and not all tools provide methods for deriving the planned landscape from the planned project portfolio. Finally, not all tools provide concepts for life-cycle aspects of infrastructure components.

**Visualization of Changes in EA Models**

Buckl et al. identified the challenge in the visualization of the development of business support provided by the application landscape over time [7]. In their approach both a Gantt chart inspired graphical viewpoint for supporting EA transformation documentation and a conceptual model explaining the information demands that need to be satisfied with the creation of road-map plans were introduced. Their approach allows the analysis and visualization of the business application’s life-cycle given an IT project. However, there is no evidence of the viewpoint’s expressiveness in representing both the motivation and nature of each transformation.

Ross et al. focused on the transformation procedure also from a visual perspective [18,19]. They proposed a high-level maturity model for EAs providing starting points for the design of the transformation process to enhance the used EAM process. However, neither a meta-model nor visualizations supporting the transformation process were discussed in their approach.

Roth et al. contribution is based on 4 layers [20]. In the first layer, changes to the meta-model are shown in a graph where new classes are shown in green, updated classes are
shown in orange, and deleted classes are shown in red. Changes involving the creation of class names are tinted with green. The same class names are tinted with red when deleted. Changes in relations between classes are also shown following the same logic, green expresses new relations, red concerning deleted relations, and changed ones portrayed in orange. The second layer provides an overview of objects (class instances) following the previous logic: new instances are presented in green, changed ones in orange and deleted ones in red. Besides the visualization, filter and zoom in/out, features are available to facilitate the visualization. The third layer shows the instance neighborhood, i.e., the neighbors of a chosen object, but focusing on the relations of Objects instead of attributes since the expert acknowledges links (instances of relationships) between objects to be a far more interesting metric for analyzing changes rather than attributes [20]. In this case, the color code is the same as in the other layers. In the fourth and final layer, the user has the option to choose an object and then view the different versions of that object, thus comparing the different versions with the original one.

**RESEARCH PROPOSAL**

To address the research problem, a tool that provides an environment to visualize and manage the evolution of the EA meta-model, as part of an EA project execution is presented. The objectives of the solution were grounded on the reviewed literature and also according to the following requirements, gathered from two rounds of seven interviews with practitioners and researchers:

- 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 provide the description of change’s impact;
- 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.

**Information Model**

The information model in Figure 1, illustrates the set of concepts and relationships defining the interface’s data schema. Despite the migration of the EA model (ensuring conformance with the new EA meta-model version) not being the focus of this research, the information model also expresses the concepts and relationships concerning the EA model for future efforts on migrating EA model data.

![Figure 1. Information Model](image)

**User Interface**

The interface 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 interface 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. 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 "Submit" option followed by "Send to Production" 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.

**Interface Structure**

On the interface’s top-left corner, project functions are shown as "New project", "Open project" and "Submit" allowing the user to create a new project, switch between projects or submit the actual changes for administration approval respectively.

On the interface’s 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 his organization’s needs.
On the center of the interface, 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 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 2).

On the interface’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.

**Interface Interaction**

As stated previously, the interface is based on Silva et al. co-evolution process [25]. However, few changes were made to allow the user to create various projects on the
same project and test all the changes he wants to make. The workflow done by the user is illustrated in Figure 3.

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 visualization interaction. For example, to add a new class to a specific domain, the user right clicks on the domain and then 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.

The color scheme used by Roth et al. [20] was chosen to represent the EA Class state. The viewpoint itself is also interactive. For example, a double-click on a specific EA Class navigates to the class properties edition window.

DEMONSTRATION
To assess the interface’s effectiveness in addressing the research problem a field experiment was done in the scope of an EA project for the Portuguese Navy. The project 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 stage was composed of two steps: 1) mapping the common concepts of the three EA meta-models, and 2) creating a unique EA meta-model containing all the concepts and relations presented in each one. In this stage, a repository containing the EA meta-model with more data was used as input. 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 were not previously contemplated by any of the three EA meta-models. EA operations were applied with resource to the interface’s management features.

The interface 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 altered the state of the existing EA meta-model elements.

EVALUATION
To test the tool’s efficacy two different methods were used, 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 other hand, 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.

In the next sections, a description of the evaluation, as well as the obtained results, is presented.

User Tests
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 UI 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 fully based on SUS questionnaire [5] with the addition of a seven-point adjective anchored Likert scale question [4], 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. From the questionnaire 75% of users gave a score above 80 points with an average score of 81.56, hence and considering the work of [3] 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.

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

The focus group suggested that the impact analysis should be more explicit and have more in-formation 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 artefacts 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 tool that they usually use. Most of the time they edited the meta-model using a text file with the meta-model described in a text-based language, some-thing that could create a lot of mistakes a lot of time debugging, and that is solved using this tool.

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.

Interviews
The interviews were made with 10 different EA practitioners 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 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. The main fault of the tool following the information gathered from the interviews and focus groups was that the tool’s lack of expressiveness concerning the impact of meta-model changes to the EA model. 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,
CONCLUSION
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. The tool 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.

Discussion
The need for a solution that could mitigate the gap in the current EAM tool spectrum regarding visualization and management of EA transformations was identified. 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 provide the description of 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 [8].

The tool’s information 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 information 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 [5], 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.

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.

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, with focus on extend the impact analysis expressiveness. Afterwards, the integration of other EA transformations stated on [24] and on [29] is also needed. To finish, a total integration with the EAMS EAM tool is necessary.

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

In these papers, the authors present a first approach to 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 also been submitted to EMISA 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. This last paper also presents the proposed tool but addressing the topic and solution from the industry perspective.

REFERENCES


