Towards an Enterprise Architecture Model Evolution

Simon Hacks and Horst Lichter[1]

Abstract: One central aim of Enterprise Architecture (EA) is to keep the EA model up-to-date to provide recent information to its stakeholders. Based on EA information, projects develop their results, called solutions, which possibly affect the EA model. As the interaction between projects and EA is not always coordinated, wrong decisions can be taken and, consequently, solutions can be developed that do not conform to the EA.

Therefore, we propose an enterprise architecture roundtrip process that guides the coordinated and project-driven distributed evolution of an EA model.

Keywords: Enterprise Architecture Management, Model Merging, Enterprise Architecture Roundtrip, EAM, Project Alignment, Distributed Evolution

1 Introduction

Since it beginnings in the 1980's, Enterprise Architecture (EA) has developed to an established discipline. The ISO 42010:2011 defines architecture 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" [III11]. As this definition implies, the EA model, comprised by the organization's elements and relationships, is a central artefact of EA. Additionally, EA has to provide important and up-to-date information of the organization to its stakeholders.

There are a lot of different sources for changes of the EA model [Fa12], which contribute to a continuous evolution of the EA model. As our research assumes a project-driven environment, we will refer to projects as the main source of changes. However, proposed changes by the projects can be contradictory, e.g., because the interaction between projects and the EA is not coordinated, leading to a state where the EA model and the solutions developed by projects are not consistent.

To overcome the EA model inconsistency problem rooted in distributed evolution scenarios, an architecture roundtrip process is proposed. As existing research focuses solely on single aspects of such a roundtrip process, like model merging issues [Ke15], quality related questions [NP13], or shed light on the maintenance of the application layer [Pi17], we phrase our research question associated as follows:

[1] RWTH Aachen University, Research Group Software Construction, Ahornstr. 55, 52074 Aachen, Germany, {hacks, lichter}@swc.rwth-aachen.de

How does a complete and effective architecture roundtrip process looks like supporting a distributed EA evolution?

Obviously, this process has to define roles as well as the needed process steps. Knowing these steps, we can investigate which steps can be automated to avoid error prone manual work [Mo09]. Our research contributes to the domain of EA by developing a holistic evolution process for EA models and categorizing related research. Figure 1 illustrates the relation between existing research and our research.

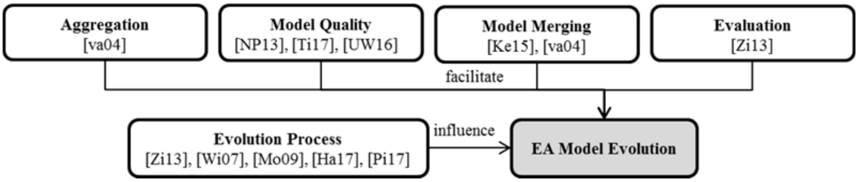

Fig. 1: Relation of Existing Research and EA Model Evolution

This paper is structured as follows: First, we detail the identified problem to motivate our research. Second, we present the proposed architecture roundtrip process. Last, we look at related research considering a complete roundtrip, or supporting single process steps of a roundtrip, before we conclude and propose possible future work.

2 Problem Statement and Motivation

The projects of an organization and its EA model are related in different ways, cf. figure 2. When it starts, each project receives its needed information based on the current EA model version. All changes performed by a project affecting the EA model are collected and documented in a so called EA change set (denoted $\Delta P_x(EA_y)$). For instance, project P_1 receives information based on version EA_1 and subsequently creates its change set $\Delta P_1(EA_1)$. At the end of P_1, this change set has to be integrated into EA_1 to create the succeeding version EA_2.

As projects do not necessarily know about the architecture decisions that were taken by other projects, we assume that projects develop their solutions independently resulting in a distributed evolution of the EA. This leads to different challenges: e.g., the change sets of projects may not be completely distinct; they share some common EA information. When these change sets are integrated, the union of the change sets with the EA model may cause conflicts [KWN05].

Therefore, the integration process of change sets has to ensure that the EA model remains always consistent. This becomes more difficult if the different timespans of projects are considered, too. E.g., project P_3 starts when EA_1 is the current version and ends after version EA_2 already exists, integrating the change sets of projects P_1 and P_2. Hence, there could be data contained in the change set of project P_3, which is based on outdated EA information.

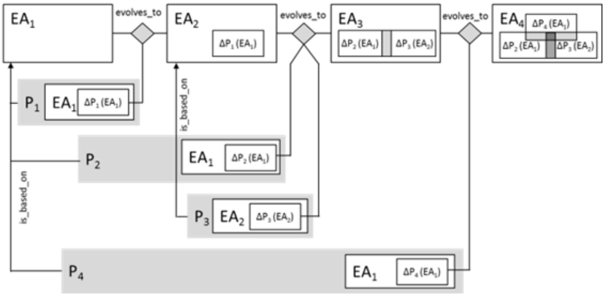

Fig. 2: Dependencies between Different Projects Elaborating on the EA Model

3 Architecture Roundtrip Process

The roundtrip process involves two EA roles: *enterprise architects*, and *solution architects*. Enterprise architects are responsible to maintain the EA model and keep it up-to-date. In contrast, solution architects are responsible to create project specific solutions based on the overall EA model. The developed solutions may affect the EA model and have to be integrated appropriately.

As an EA model is a descriptive representation of the EA, a revealed deviation between the EA and the EA model is the trigger for enterprise architects to evolve the EA model as presented in figure 3.

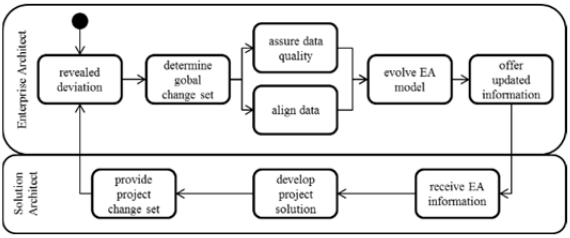

Fig. 3: Visualization of the Roundtrip Process

In the first step, the global EA change set, aggregating all project-specific change sets, for the next roundtrip needs to be determined. As this is a known problem in model-driven software development (cf., [Ke15]), we reuse the respective technique to elaborate on our issue. A change set is defined as a group of atomic model changes belonging logically together. Facilitating this approach to the domain of the architecture roundtrip, the solution architects have to deliver project-specific change set besides the modeled solution architecture containing all atomic changes they made. Furthermore, the enterprise architects have to consult with the solution architects about the changes while determining the global EA change set to ensure the consistency of the EA model if conflicts arise.

After determining the global EA change set, the EA model can be evolved. But at first, the data of this change set may be aligned to yield the right abstraction level, as usually the architecture models provided by the solution architects are more detailed than needed for EA. The respective alignment consists of two steps: First, too detailed elements need to be deleted and, second, derived relations need to be insserted.

Identifying too detailed elements is challenging, since the right level of abstraction depends on the organization and even in the research community there is no common agreement on the right level of abstraction [ARW08]. Indications can be aggregation or composition relations between elements where the aggregated/composed elements may be needless.

If elements are removed, transitional relations may get lost, which should not happen as this information is still important and needed. Van Buuren et al. [va04] suggest a method to derive a relation between two non-connected ArchiMate elements by analyzing the path between these elements and choosing the most general relation. This technique can be applied in order not to lose this information.

After aligning the change set, a quality assurance is necessary to ensure the overall quality of the EA model, including, for example, the detection of duplicates, the correction of typing errors, the amendment of semantic misuse of modeling elements, or checking for certain characteristics of the EA [UW16]. As we already did a first step on the path to ensure EA model quality, we refer for further details to our previous work [Ti17].

After determining the change set, aligning the data, and assuring data's quality, the enterprise architects can import the processed global EA change set into the EA model. This step needs to be done mostly manually, because conflict resolution cannot be solved fully automatic. Those conflicts arise from the problems stated in chapter 2. The enterprise architects have to keep the project-specific change sets in mind, delivered by the solution architects. If a conflict arises, the enterprise architects and the solution architects who have issued the conflict have to solve the conflict.

Providing the updated information to EA's stakeholders is straight forward. All reports have to be created newly and pushed into their communication channels. An area of special interest in the context of the architecture roundtrip is the project-specific view on the EA model. This view is the basis on which the solution architects model the solution architecture and define needed changes of the EA model.

4 Related Work

Zimmermann proposes a reference process model to guide the Business-IT-Management (BIM) reusing the concepts IT governance, strategic management, multi-project management, and EA [Zi13]. Within the process of project mentoring of the EA, he outlines the added value of providing up-to-date information regarding the EA model to the projects [Zi13, pp. 176-178]. Moreover, he highlights that the additional effort to keep the EA up-to-date at the end of projects pays back [Zi13, pp. 176-178. Unfortunately, Zimmermann neither elaborates on this update nor discusses the problems may arise.

In contrast to Zimmermann [Zi13], Wittenburg [Wi07] outlines an architecture roundtrip where the EA and projects interchange sequentially. But, there exist several interchanges, which can be understood as small roundtrips. Unfortunately, Wittenburg focuses on the representation of the application landscape and does address the issues we are facing.

Hartmann [Ha17] illustrates not the complete roundtrip but the alignment between software development and EA. He embeds the EA governance process into the software development process and, though, enforces the governance. His intention was to place EA needs also in an agile development process.

5 Conclusion and Future Work

Within this work, we have presented a process to overcome the problems related to a distributed EA model evolution. However, there are some shortcomings which are situated in the additionally generated effort for the solution architects and the enterprise architects. Whereby, this depends strongly on the realization within an organization. Thus, the enterprise architects may accept every change supposed by the solution architects and only react on conflicts. In this case, the additional effort should be not too high for the enterprise architects. Unlike, the enterprise architects may check every single change causing additional effort. The additional effort for the solution architects cannot be neglected, as they have to model the changes. But this investment will lead to a better EA model [Zi13, pp. 176-178].

Furthermore, our work offers still some fields of improvement. As our work is only partly evaluated so far, further case studies for the single steps of the process and for the whole architecture roundtrip process are necessary. Moreover, we took only research from Software Engineering and Information Systems into account. Probably, there are other interesting trends in other fields of research which can add value to our research.

Future work should elaborate on these limitations. Additionally, we will work on the single steps of the roundtrip process: First, we will deepen our work on a quality framework for EA. Second, we like to introduce probabilities into our EA model, to persist different evolution scenarios. Last, we like to transfer the change set determination and all related means (e.g., [Ke15])to the EA domain, i.e., to ArchiMate.

References

[ARW08] Aier, S.; Riege, C.; Winter, R.: Unternehmensarchitektur. Literaturüberblick und Stand der Praxis. In Wirtschaftsinformatik, 2008, 50; pp. 292–304.

[Fa12] Farwick, M. et al.: On Enterprise Architecture Change Events. In (Aier, S. et al. Eds.): Trends in Enterprise Architecture Research and Practice. Springer, Berlin, Heidelberg, 2012; pp. 129–145.

[Ha17] Hartmann, A.: Enterprise Architecture als Katalysator zwischen Qualität, Effizienz und Governance. In (Eibl, M.; Gaedke, M. Eds.): INFORMATIK 2017. Gesellschaft für Informatik, Bonn, 2017; pp. 2121–2126.

[III11] ISO; IEC; IEEE: Systems and software engineering -- Architecture description. ISO; IEC; IEEE, 2011.

[Ke15] Kehrer, T.: Calculation and Propagation of Model Changes Based on User-level Edit Operations. A Foundation for Version and Variant Management in Model-driven Engineering. Dissertation, Siegen, Germany, 2015.

[KWN05] Kelter, U.; Wehren, J.; Niere, J.: A Generic Difference Algorithm for UML Models. In Software Engineering, 2005, 64; pp. 4–9.

[Mo09] Moser, C. et al.: Some Process Patterns for Enterprise Architecture Management. In (Münch, J.; Liggesmeyer, P. Eds.): Software Engineering 2009 - Workshopband. Gesellschaft für Informatik e.V., Bonn, 2009; pp. 19–30.

[NP13] Niemi, E.; Pekkola, S.: Enterprise Architecture Quality Attributes. A Case Study: 2013 46th Hawaii International Conference on System Sciences. IEEE, 2013; pp. 3878–3887.

[Pi17] Piontek, T.: Kollaboratives Management von Softwareprodukten im Rahmen von EAM. In (Eibl, M.; Gaedke, M. Eds.): INFORMATIK 2017. Gesellschaft für Informatik, Bonn, 2017; pp. 2095–2105.

[Ti17] Timm, F. et al.: Towards a Quality Framework for Enterprise Architecture Models. In (Lichter, H.; Anwar, T.; Sunetnanta, T. Eds.): Proceedings of the 5th International

Workshop on Quantitative Approaches to Software Quality (QuASoQ 2017) co-located with APSEC 2017. CEUR-WS.org, 2017; pp. 10–17.

[UW16] Ullrich, A.; Weber, E.: Einsatz stilisierter Fakten zur Bewertung wandlungsfähiger Unternehmensarchitekturen. In (Mayr, H. C.; Pinzger, M. Eds.): Informatik 2016. Gesellschaft für Informatik e.V, Bonn, 2016; pp. 785–798.

[va04] van Buuren, R. et al.: Composition of Relations in Enterprise Architecture Models. In (Ehrig, H. et al. Eds.): Graph Transformations Second International Conference, 2004.

[Wi07] Wittenburg, A.: Softwarekartographie. Modelle und Methoden zur systematischen Visualisierung von Anwendungslandschaften. Dissertation, München, Germany, 2007.

[Zi13] Zimmermann, K.: Referenzprozessmodell für das Business-IT-Management. Vorgehen, Erstellung und Einsatz auf Basis qualitativer Forschungsmethoden. Dissertation, Hamburg, Germany, 2013.