Providing EA Decision Support for Stakeholders by Automated Analyses

Dierk Jugel[1,2], Stefan Kehrer[1], Christian M. Schweda[1] and Alfred Zimmermann[1]

Abstract: Enterprise architecture management (EAM) is a holistic approach to tackle the complex Business and IT architecture. The transformation of an organization's EA towards a strategy-oriented system is a continuous task. Many stakeholders have to elaborate on various parts of the EA to reach the best decisions to shape the EA towards an optimized support of the organizations' capabilities. Since the real world is too complex, analyzing techniques are needed to detect optimization potentials and to get all information needed about an issue. In practice visualizations are commonly used to analyze EAs. However these visualizations are mostly static and do not provide analyses. In this article we combine analyzing techniques from literature and interactive visualizations to support stakeholders in EA decision-making.

Keywords: Enterprise Architecture, decision support, visual analytics, automated analyses

1 Introduction

Enterprise Architecture Management (EAM) is a commonly accepted method to support enterprises in their continuous transformation processes. These processes are necessary, as enterprises need to continue growing in order to be sustainable in a competitive environment. Enterprises are large and complex systems consisting of business processes, organizational units, applications and other elements. EAM provides a systematic approach to enhance transparency, to support business and IT-alignment and to enable the strategy-driven development of the Enterprise Architecture (EA) as a whole [Ha12]. Key to EAM is the systematic evolution of the EA over time. Modifications of business processes, but also applications have to take place, and affect plenty of other elements throughout the EA. Frameworks like The Open Group Architecture Framework (TOGAF) [TOG09] assist organizations with a holistic approach for EAM. The TOGAF Architecture Development Method (ADM) [TOG09] details EAM tasks stating that analyses of the EA have to be undertaken prior to decision-making. EA analysis is used to derive and extract information relevant for decision-making from the available information about Enterprise Architecture [Ra14]. In this vein, as Johnson et al. describe in [Jo07], EA analysis is the "application of property assessment criteria on enterprise architecture models". The complexity of these models and the impacts of a change over different layers of the EA require the involvement of different stakeholders. In particular, exchange between these stakeholders is necessary to agree on EA planning, as Lucke et

[1] Reutlingen University, Herman Hollerith Zentrum, {dierk.jugel, christian.schweda, alfred.zimmermann}@reutlingen-university.de, stefan.kehrer@student.reutlingen-university.de
[2] Rostock University, dierk.jugel@uni-rostock.de

al. identify in [LKL10] the stakeholder's communication is a critical issue in the field of EAM. Summarizing, one can say that EA analysis and planning processes are skill-intensive and dependent on the competence and the decision-making ability of the engaged and often diverse team of stakeholders.

In general EA analysis uses visualizations of relevant information to support stakeholders in their individual tasks [Ma08]. EA tools provide different visualization techniques, which are usually static and do not target an interactive style of collaboratively working on a decision. Therefore, stakeholders seeking to make a decision regarding the evolution of the EA need assistance of "EA analysts" to develop an EA-based argumentation representing their concerns contributing to an architectural effort. Florez et al. [FSV14] propose to automate analysis methods in order to facilitate decision-making. By automating and standardizing analysis processes, EA activities get more reliable and repeatable without manual errors.

In this article we extend the ideas of Florez et al. from [FSV14] towards a visual analytics approach for EAM. In the sense of Keim et al. [Ke08], who define visual analytics as the combination of automated analyses and interactive visualizations, we combine automated analyses of the EA structure with an interactive visualization mechanism. Visual analytics can be regarded a novelty for the field of EAM, as the survey of Roth et al. [RZM14] inquiring visualization capabilities of current EAM tools shows.

First we put the base from existing approaches for EA analysis in Section 2. In Section 3 we describe how structural analyses of the EA can be automated by our novel approach. In Section 4 we link the analysis results to our approach to support interactive visualizations. Final Section 5 concludes our results and gives an outlook.

2 Related Work

TOGAF [TOG09] is the de facto global standard for EAM and outlines a general method for project-like development and evolution of the EA. The Architecture Development Method (ADM) is the core of TOGAF and describes a sequence of development phases starting with "A – Architecture Vision" and concluding with "H – Architecture Change Management". The three phases "B - Business Architecture", "C – Information Systems Architecture" und "D – Technology Architecture" each encompass activities of EA analysis building on the available information. This information is organized in different EA models pragmatically. TOGAF [TOG09, p. 95] recommends to "gather and analyze only that information that allows informed decisions to be made relevant to the scope of this architecture effort". As a general framework TOGAF does not make specific recommendations on the analysis procedures and techniques. It contrariwise offers only high-level guidance and needs to be complemented with techniques for analysis [KW07]. In Section 2.1 we reflect best practice analyses described in literature. Section 2.2 addresses related work in the field of automated EA analyses.

2.1 Best Practice Analyses in EAM

Johnson et al. recommend in [JE07] a goal-driven approach for EAM. The authors especially address how to derive decision-relevant information from EA models. Johnson et al. discuss that architecture-related goals have to be operationalized to provide a foundation for the decision-making processes. Therefore, the authors propose a mapping of goals and necessary viewpoints. They stress the specific creation of viewpoints to support stakeholders in their decision-making tasks. Automated assessment and corresponding tools are only mentioned for the analysis of influence diagrams, respectively to their goal-oriented approach. The goal-driven approach provides a systematic framework for understanding and performing EA analysis, but only offers limited guidance for the identification of architecture improvements.

Hanschke presents so-called analysis patterns in the appendix A of [Ha13]. These analysis patterns are described as practice-proven and generalized templates to find needs for action and potential improvements concerning the EA. Hanschke identifies five different categories: 1) redundancy, 2) inconsistency, 3) organizational need for action, 4) implementing business requirements and 5) technical need for action and potential improvements. Each analysis pattern is structured using a canonical form including the following characteristics: id, name, version, description, context, dependencies, result, and example. Each pattern provides a textual prescription on how to identify shortcomings and derive improvements in the architecture. This prescription gives guidance to an experienced enterprise architect, but is not translated to a formal, i.e. algorithmic, manner. The context of the pattern states necessary requirements concerning the underlying EA model, but does not provide an explicit model. If there are any dependencies to other patterns they are described in the dependencies section.

Matthes et al. [Ma11] present quantitative, metrics-driven EA analyses resulting in a set of EA key performance indicators (KPIs). These KPIs provide necessary measurement capabilities for EAM, needed to aid planning and controlling the EA. The EAM KPI Catalog contributes in presenting ten common EA management goals and 52 KPIs including the underlying EA model. Each KPI is described by the following characteristics: description, underlying EA model, goals, calculation, code, sources, organization-specific instantiation and the affected layers of the EA. The EAM KPI Catalog also provides a good basis for implementation purposes. Especially the calculation section describes the algorithm for calculation of each KPI concisely. While KPIs, like the project performance index, can be used to detect need-for-action, optimization potentials in the EA itself are not identified. An experienced enterprise architect would be needed to identify the elements of the EA causing the value of the KPI.

2.2 Automated EA Analyses

Buschle et al. present a tool, which supports the Predictive, Probabilistic Architecture Modeling Framework (P^2AMF) in order to perform analysis on EA models [BJS13]. The authors especially promote the capability of the tool 1) to analyze EA models without hardcoded analyses and 2) the ability to handle model incompleteness. Basically they use two methods to address the aforementioned capabilities: 1) a model transformation using extended Eclipse Modeling Framework (EMF) containing the so called Class Modeler and the Object Modeler, and 2) P^2AMF derivations to reconstruct missing attributes using a model-based dependency structure. The assessment framework built in the Class Modeler can be loaded into the Object Modeler to instantiate the predefined classes. Additionally, the evidence for missing attribute values can be supplied by sampling according to P^2AMF [BJS13]. The underlying concept is further detailed in [Jo07]. The automated approach of Buschle et al. provides a framework for quantitative analysis under uncertain and incomplete information. This approach can be applied to different quantitative best-practice analyses, e.g. the KPIs defined by Matthes et al. [Ma11]. The results of the analyses are presented in simple graph-like visualizations or in a tabular manner, covering only a subset of the visualization types prevalent in EAM.

Florez et al. [FSV14] discuss the motivations and requirements for EA model analysis automation and the requirements of analysis methods. Depicted advantages of automated analysis procedures are the possibility to work with larger models and the lesser chance of manual errors. On the downside, Florez et al. identify the following three problems with automated EA analysis: 1) Higher model requirements would lead to more complicated and hence more expensive modelling processes. 2) Analysis methods are selected late in projects whereas meta-models have been already designed and corresponding EA models constructed. 3) In general EAM tools do not support flexible meta-models. Such flexibility would – according to the authors – be required to store analysis results along the analyzed elements of the EA. Florez et al. [FSV14] derive seven requirements for future EAM tools and present an approach called SAMBA to realize these requirements. A key point of their approach is the capability to change the meta-model by analyses.

Ramos et al. propose in [Ra14] a characterization of analysis functions including a specification of the algorithms. The authors consider the explicit documentation of information requirements that every analysis method has as important. Therefore, the structure for each analysis function contains the following information: name, description, dimension, type, layer, entities and relations, structural attributes and algorithm. Whereas the dimension reuses the classification concept by Lankhorst [La13], the type references one of the types, developed according to the different concerns. The layer depicts the concerned ArchiMate layer and the entities, relations and structural attributes characterize the underlying EA model. Finally, the algorithm describes the information extraction of the analysis function from the model. The analysis process itself is depicted as a cyclic procedure of querying and enriching the EA model until the result is achieved. Thereby, the authors also use the aforementioned method of Florez et al. [FSV14] to store results of analysis processes in the EA model. Additionally they mention the possibility to use a set of newly generated data in a visualization without transforming the EA model. The authors of [Ra14] are currently working on a catalog of so called analysis functions and a conceptual framework on top of ArchiMate.

Naranjo et al. describe in [NSV14] a flexible and configurable graph-based approach that makes automated EA analysis methods applicable in a pipeline. This holistic approach is grounded on the Visual Analysis framework PRIMROSe. Contrary to Florez et al. [FSV14] and Ramos et al. [Ra14] the authors use the so called Conceptual Meta-model to store the initial EA model and the Analysis Meta-model, as extended version enriched with facts generated by the EA analyses. The analysis is performed by a transformation process through applying a series of five pipelined stages: 1) Import: At first the EA model is transformed to an expanded graph G_E to allow the insertion of new relation properties. 2) Analyze: The transformation to an analysis graph G_A contains additional so called selectors pointing to existing vertices of G_E. Selectors can be classified as analysis functions, which updates the graph, or as decorator functions, which only add additional attributes to existing vertices. 3) Map: After additional information has been added, so-called visual rules are used to translate data into visual information. Therefore a visual rule is a pair of a visual attribute and a mapping function. Furthermore a visual decorator can be used to apply a set of visual rules to a vertices marked by a selector in order to create a visual graph G_V, which is an isomorphism of G_E enriched with visual properties. 4) Visualize: G_V can be transformed into the desired graphical format to support View Operators, which do not alter the visualization, and Data Operators, which are able to modify all internal pipelines of the stages applied before to alter the visualization. 5) Communicate: Finally, filters can be applied sequentially to remove vertices. Thereby, stakeholders can highlight relevant parts of the model in a resulting view. Naranjo et al. present a holistic and modular approach of combining the two important EA functions analysis and visualization. For EA analysis purposes the pipeline concept carries a major advantage: analysis methods can be applied sequentially to support highly sophisticated analysis structures.

Antunes et al. present in [An13] an ontology-based approach to document knowledge about the EA and support EA analysis through model reasoning. Thereto, the authors use a core meta-model called domain-independent ontology (DIO) that can be extended by domain-specific languages. The concepts presented in [An13] are at an early stage. Further explorations of the analysis possibilities considering reasoning and querying methods are still open tasks.

3 EA Analysis Technique

The analysis patterns presented by Hanschke in the appendix A of [Ha13] (c.f. Section 2.1) provide an operationalization of EA analysis and planning. Hanschke describes the patterns in a generic way to enhance reusability and flexibility concerning the underlying meta-model. However the analysis patterns of Hanschke cannot be automated without additional work. We especially identify two further aspects needed: 1) a meta-model defining the structure needed to perform the analysis and 2) the need of a concise algorithm describing the application of the analysis in steps that can be implemented. Although we strive for automated EA analysis, it is not the scope of our approach to establish automated decision-making. Modification of the existing EA can have extensive impact on many other elements of this intrinsically complex system, which may be not documented in the analyzed EA model. This would not be a constructive approach for optimizing the EA. Notwithstanding the theoretical concept to establish automated EA decision-making exists; we assume the practicability nowadays as not mature enough. Therefore we agree with Hanschke [Ha13] that the EA analysis patterns can only recommend possible needs for actions and optimization potentials, which have to be verified by stakeholders.

In the following we extend an exemplary analysis named “Redundancy concerning the business support”. We develop the underlying meta-model needed to perform the analysis. The underlying meta-model satisfies the requirement proposed by Ramos et al. [Ra14] to explicitly document the information requirements of every analysis method. The analysis patterns comprise a context section that textually describes the concepts and relations of the necessary meta-model. Furthermore, some necessary properties and relations of the concepts have to be extracted from the description section that describes the analysis’ proceeding. The meta-model (cf. Fig. 1) is influenced by the meta-model, which is used by Hanschke [Ha13] and also implemented in the EAM tool iteraplan[3]. In Fig. 1 we introduced transitive relations to designate semantically important relations more easily. Transitive relations are highlighted with as dotted line.

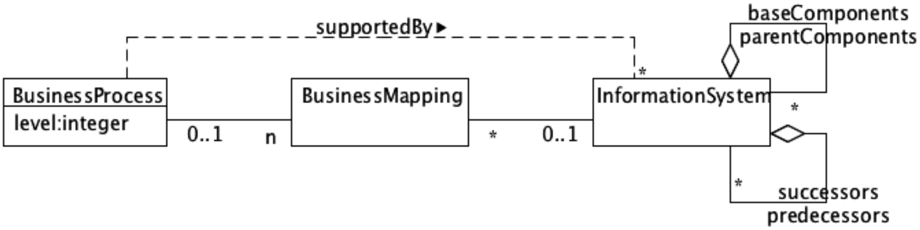

Fig. 1: Meta-model for the exemplary analysis pattern

Hanschke describes a redundant business support as the existence of more than one information system that supports the same business process. In addition she states the following additional conditions: only business processes above the Event-driven Process Chain (EPC) layer are considered; different releases of information systems and information systems that are not isochronal active are not considered; information systems being in a direct or indirect “is part of” relationship are not considered. The following pseudo code operationalizes above definition.

[3] www.iteraplan.de/en

```
//Parameters
NL = <<level of EPC layer>>
RESULT = []
FOREACH BP ∈ BusinessProcess:
   IF BP.supportedBy.size < 2 OR BP.level <= NL:
      CONTINUE
   END IF
   FOREACH IS ∈ BP.supportedBy:
      M = BP.supportedBy \ IS.baseComponents \
          IS.parentComponents
      FOREACH PD ∈ IS.predecessors:
         IF PD ∈ BP.supportedBy:
            IF !(PD.ISR$TypeOfStatus == CURRENT
             AND IS.ISR$TypeOfStatus == CURRENT):
               M.remove(PD)
            END IF
         END IF
      END FOREACH
      FOREACH SC ∈ IS.successors:
         IF SC ∈ BP.supportedBy:
            IF !(SC.ISR$TypeOfStatus == CURRENT
             AND IS.ISR$TypeOfStatus == CURRENT):
               M.remove(SC)
            END IF
         END IF
      END FOREACH
      IF M.size > 1:
         RESULT.add(BP, M)
      END IF
   END FOREACH
END FOREACH
```

The pseudo code is a mix of high-level programming language syntax and uses mathematical concepts of set theory extensively. To implement the analysis "patterns" additional assumptions sometimes have to be made. In case of our pattern example we interpret the considered information about the business process layer to be an integer value. To consider all business processes above the EPC layer they need to have a value for level higher than the defined level of the EPC layer. We consider relationships between model elements to be bidirectional traversable. Therefore we use the inverse function called on the contrary model element. Whereas `IS.supports` for `IS ∈ InformationSystem` delivers a set of business processes which are supported by the information system `IS`, `BP.supportedBy` for `BP ∈ BusinessProcess` returns a set of information systems supporting the business process BP. As result the algorithm provides a set of business processes, which uses more than one information system like described in the pseudo code.

4 Analyses with interactive visualizations

In this section we describe how to integrate automated EA analyses into the approach presented in [JS14]. The authors present interactive functions of a cockpit to visually support EA planning [JS14]. The cockpit is characterized as a room with multiple (interactive) screens to display multiple coherent views in parallel. Several interactive functions like "graphical highlighting & filtering" or an "impact analysis" are named. However detailed analyses are not described. The integration of this approach with automated analyses is a first step to realize visual analytics. Our intention is supporting stakeholders in collaborative EA analysis and decision-making processes. Thereby we apply the cockpit approach supporting the consideration of multiple viewpoints in parallel to the domain of EAM. The cockpit is further extended with interactive functions that facilitate the collaboration and discussion between the stakeholders. In particular, analyses are considered as an interactive function that generates additional information about the EA. This additional information is added via so-called "annotations", which – in contrast to Florez et al. [FSV14] – are not part of the original EA model. The annotations are stored alongside the EA model, representing the – potentially quickly changing – state-of-thought in the discussion of the EA model. In [JSZ15] we refine the concept of annotations by distinguishing different types of annotations to document knowledge arisen during the decision-making process. In this case the type "Information" is especially of interest, because this concept can be used to add findings and discussion results. The application of a particular EA analysis and the result thereof is a valuable additional knowledge that has to be collected. Such information helps stakeholders in decision-making, but it can be also used to understand decisions of the past especially why a decision is taken and what is the reason thereof. In addition quality assurance processes can be established, which define analyses as prerequisites for particular decisions. By using the collected information during the decision-making process, it can be ensured that a particular analysis is applied. Fig. 2 illustrates the integration of analysis patterns into the approach described in [JS14].

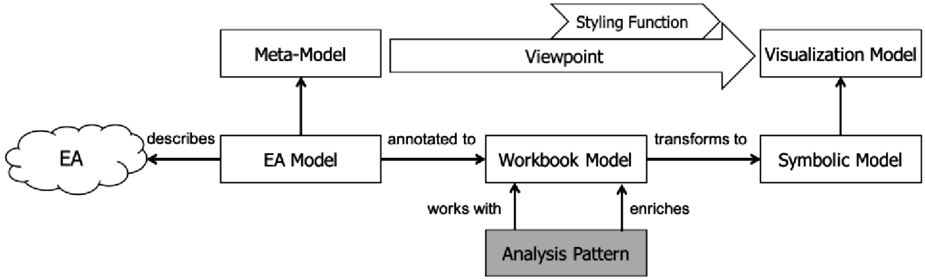

Fig. 2: Integration of Analysis Patterns into Workbook Model

On the left side the *Meta-Model* describes the concepts used to model an EA. For instance there is a concept named "Information System" with several properties and relations to other concepts. The *EA Model* is the instance of the *Meta-Model* and includes the model of the “real world“. This model contains concrete architecture elements like the Information System “CRM 2.3“. Thus we do not want to add additional knowledge to the *EA Model* we introduced the *Workbook Model*, which is a combination of the *EA Model* and related annotations. The starting point for performing the analysis patterns described in Section 3 is the *Workbook Model*. The prerequisite for performing a pattern is that the defined meta-model of the pattern is part of the *Workbook Model*. Thereby the calculation algorithm is based on the structure described by the *Meta-Model* and on annotations as well. For instance there is information needed by the algorithm that is not part of the *Meta-Model*, but the information is generated during the analysis and decision-making process by performing other analyses. After performing an analysis pattern, annotations with the result are generated automatically.

In analogous the *Visualization Model* on the right side describes concepts needed to visualize viewpoints. For instance there is a concept named ”View“ that consists of several symbols, which can be e.g. rectangles with a particular fill color and a distinct position within the view. The *Symbolic Model* is the instance of the *Visualization Model* and contains concrete views with concrete rectangles, e.g. an Information System Diagram consisting of several rectangles representing the information systems. *Viewpoints* are the link between the left side and the right side. This aligns with the definition of viewpoint in the ISO Std. 42010 [Iso11] as a prescription for the construction of views and their interpretation and usage. A viewpoint defines a model transformation from the *Workbook Model* to a *Symbolic Model*. Moreover the Styling Function facilitates adding styles to annotations that can be used to highlight or filter architectural elements within views.

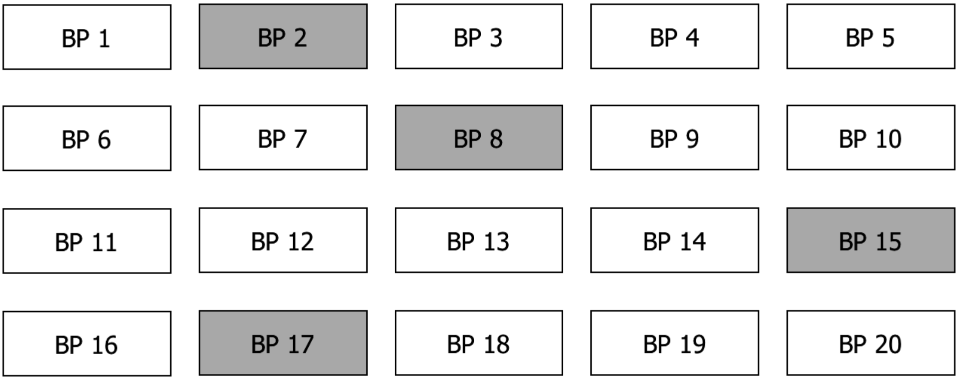

Fig. 3: Business Process Overview Viewpoint

Concluding we want to illustrate the approach by considering the analysis pattern named “Redundancy concerning the business support” described in Section 3. Imagine there is an enterprise architect, who has to investigate these redundancies. To get an overview about the situation he uses the “Business Process Overview Viewpoint” that is represented by a list containing all business processes. The viewpoint provides different interactive functions like described above. One of these functions suggests available analysis patterns concerning business processes. The enterprise architect chooses the pattern

“Redundancy concerning the business support”. The pattern algorithm is performed in the background to detect business processes that are using more than one information system. An annotation of the type “Detailed Information” is created that relates these business processes. In addition a styling function is applied to assign the elements of the annotation with a particular fill color. Annotations represent additional knowledge. To visualize them the styling function mechanism is needed. The result is illustrated in Fig. 3. Now the enterprise architect can decide what he wants to do next. For instance, he decides to perform another analysis to get information about the information system that is affected by one of these identified business processes.

5 Conclusion

In this article we have introduced a novel approach for integrating automated EA analyses with interactive visualizations as our contribution for an integral Visual EA analytics. The analysis patterns of Hanschke [Ha13] are very useful to identify optimization potentials within EAs. We extended these EA patterns by developing a meta-model and a calculation algorithm to provide automatic support for these EA analysis patterns. Next, we have integrated the analyses in the approach described in [JS14]. The automated analyses work with the *Workbook Model* that provides the EA model annotated with additional knowledge. The annotation mechanism provides a generic vehicle to model analysis results during decision-making processes. In future work we want to investigate different kinds of analyses in more detail, and extend these in a consistent and holistic way. Buckl et al. describe in [BMS09] three kinds of analysis techniques: expert-based, rule-based and indicator-based. The analysis patterns of Hanschke, which are rule-based analyses, are a first step to support stakeholders in decision-making. Now we investigate how other kinds of analyses contribute and how these analyses can be integrated in the approach. Furthermore we want to develop a type-safe language for performing analyses. At the moment the developed algorithms to perform the patterns are described procedurally. Another way to automate the analyses is to describe them declaratively by using a query language. The calculations of KPIs described in [Ma11] use such a declarative mechanism.

References

[An13] Antunes, G. et al.: Using Ontologies for Enterprise Architecture Analysis. In: 17th IEEE International Enterprise Distributed Object Computing Conference Workshops, pp. 361-368, 2013.

[BMS09] Buckl, S.; Matthes, F.; Schweda, C.M.: Classifying Enterprise Architecture Analysis Approaches. In: The 2nd IFIP WG5.8 Workshop on Enterprise Interoperability (I-WEI'2009), Valencia, Spain, pp. 66-79, 2009.

[BJS13] Buschle, M.; Johnson, P.; Shahzad, K.: The Enterprise Architecture Analysis Tool – Support for the Predictive, Probabilistic Architecture Modeling Framework. In: Proceedings of the Nineteenth Americas Conference on Information Systems (AMCIS), Chicaco, Illinois, pp. 3350-3364, 2013.

[FSV14] Florez, H.; Sánchez, M.; Villalobos, J.: Extensible Model-based Approach for Supporting Automatic Enterprise Analysis. In: 18th International Enterprise Distributed Object Computing Conference (EDOC), Ulm, Germany, pp. 32-41, 2014.

[Ha12] Hanschke, I.: Enterprise Architecture Management – einfach und effektiv: ein praktischer Leitfaden für die Einführung von EAM, Hanser Verlag, 2012.

[Ha13] Hanschke, I.: Strategisches Management der IT-Landschaft Ein praktischer Leitfaden für das Enterprise Architecture Management (3rd edition), Hanser Verlag, 2013.

[Iso11] ISO/IEC/IEEE: Systems and software engineering – Architecture description. Technical standard, 2011.

[JE07] Johnson, P.; Ekstedt, M.: Enterprise Architecture – Models and Analyses for Information Systems Decision Making, Studenttliteratur, 2007.

[Jo07] Johnson, P. et al.: A Tool for Enterprise Architecture Analysis. In: 11th International Enterprise Distributed Object Computing Conference (EDOC), Annapolis, Maryland, pp. 142-153, 2007.

[JS14] Jugel, D.; Schweda, C.M.: Interactive functions of a Cockpit for Enterprise Architecture Planning. In: International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), Ulm, Germany, pp. 33-40, 2014.

[JSZ15] Jugel, D.; Schweda, C.M.; Zimmermann, A.: Modeling Decisions for Collaborative Enterprise Architecture Engineering. In: 10th Workshop Trends in Enterprise Architecture Research (TEAR), held on CAISE 2015, Stockholm, Sweden, 2015.

[Ke08] Keim, D.A. et al.: Visual Analytics: Definition, Process, and Challenges. In: Information Visualization, Lecture Notes in Computer Science, vol. 4950, Springer Verlag, Heidelberg, pp. 154- 175, 2008.

[KW07] Kurpjuweit, S.; Winter, R.: Viewpoint-based meta model engineering, Enterprise Modelling and Information Systems Architectures-Concepts and Applications. In: Proceedings of the 2nd Int'l Workshop EMISA, pp. 143–161, 2013.

[La13] Lankhorst, M.: Enterprise architecture at work: Modelling, communication and analysis. Springer, 2013.

[LKL10] Lucke, C.; Krell, S.; Lechner, U.: Critical Issues in Enterprise Architecting - A Literature Review. In: Proceedings of Sustainable IT Collaboration Around the Globe, 16th Americas Conference on Information Systems (AMCIS), Lima, Peru, pp. 305-315, 2010.

[Ma08] Matthes, F.; Buckl, S.; Leitel, J.; Schweda, C.M.: Enterprise Architecture Management Tool Survey 2008. Technical Report, Chair for Informatics 19, Technical University of Munich, 2008.

[Ma11] Matthes, F. et al.: EAM KPI Catalog v 1.0, Technical Report, 2011.

[NSV12] Naranjo, D.; Sanchez, M.; Villalobos, J.: Visual Analysis of Enterprise Models. In: 16th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), Bejing, China, pp. 19-28, 2012.

[NSV14] Naranjo, D.; Sanchez, M.; Villalobos, J.: Towards a Unified and Modular Approach for Visual Analysis of Enterprise Models. In: International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), Ulm, Germany, pp. 77-86, 2014.

[Ra14] Ramos, A. et al.: Automated Enterprise-Level Analysis of ArchiMate Models. In: Proceedings of the 15th International Conference, BPMDS 2014, 19th International Conference, EMMSAD 2014, held at CAISE 2014, Thessaloniki, Greece, pp. 439-453, 2014.

[RZM14] Roth, S.; Zec, M.; Matthes, F.: Enterprise Architecture Visualization Tool Survey 2014. Technical Report, Chair for Informatics 19, Technical University of Munich, 2014.

[TOG09] The Open Group: The Open Group Architecture Framework (TOGAF), Version 9, 2009.