Alexander Rossmann, Alfred Zimmermann (eds.): Digital Enterprise Computing 2017,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 13

Applying the Minimal Cost of Change Approach to inductive Reference Enterprise Architecture Development

Felix Timm[1], Valentina Sauer[2]

Abstract: Enterprise architectures (EA) help organizations to analyze interrelations among their strategy, business processes, responsibilities, application landscape and information structures. Such ambitious endeavors can be supported by using reference models for EAM. Although research thoroughly addresses the development of reference models, the characteristics of EA models are not investigated in this context. Our work therefore applies one approach for inductively constructing reference process models to the EA domain. We thus contribute to the reference modeling research field in general and its application to development of reference enterprise architectures in detail.

Keywords: Reference Enterprise Architecture, Reference Modeling, Inductive Development, Enterprise Architecture, Enterprise Modeling

1 Introduction

Enterprises need to be aware of the relations among their strategy, business processes, applications, information infrastructures and roles to be able to rapidly react on changing demands in the market and within their organization. Enterprise Architecture Management (EAM) contributes to this purpose by providing methods and tools to establish a more holistic perspective on enterprises [ASM12, Lan17], which includes to systematically capture and develop the different architectural layers of an enterprise (e.g. business, application and technology architecture). Since EAM projects are highly time- and resource-consuming organization would benefit from reference models for EAM that are related to a problem of a certain group of organizations, e.g. to a certain industry. Reference models are information models that are reusable, of exemplary practice and universal applicability [Fet07]. In the context of EAM, van der Beek et al. define a reference enterprise architecture as *"... a generic EA for a class of enterprises, that is a coherent whole of EA design principles, methods and models which are used as foundation in the design and realization of the concrete EA that consists of three coherent partial architectures..."* [Bee12]. Although there exist many methods for the development of reference models [FL04], these approaches lack an investigation regarding their applicability towards EAM [TSF17] and mainly focus on business process model structures [RFL13].

[1] Chair of Business Information Systems, University of Rostock, Albert-Einstein-Str. 22, 18059 Rostock, felix.timm@uni-rostock.de
[2] Chair of Business Information Systems, University of Rostock, Albert-Einstein-Str. 22, 18059 Rostock, valentina.sauer@uni-rostock.de

In order to contribute to the gap of reference enterprise architecture developement, our work applies the "Minimal Cost of Change" (MCC) approach for inductive reference model development [Ard13] to EAM. Therefore, we adjust the MCC, which was develop for reference process model construction, and apply it to sixteen EA models of different financial institutions in order to derive a first reference enterprise architecture using the ArchiMate language for EA modeling. In chapter 2 we therefore set the theoretical fundament for reference modeling and EAM in general and have a close look on the MCC approach as it is defined in [Ard13]. Afterwards, chapter 3 applies the approach to our EAM endeavor. Finally, chapter 4 concludes our findings and addresses further research work in this field.

2 Theoretical Background

2.1 Reference Modeling

From a user-oriented perspective, Thomas understands a reference model as a model used to support the construction of another model [Tho06b]. From the perspective of reusability, other authors such as vom Brocke argue that reference models are characterized by the concepts of universality and recommendation [Bro03]. In general, reference models are information models developed for a certain problem in a certain application domain. The purpose of their development is to be reused in a concrete application case in this domain. The reuse of a reference model is intended to increase both efficiency and effectivity of an enterprise's information systems and their change management [Bro03].

Regarding the overall approach of reference modeling, the life cycle of reference models can be distinguished between the phase of construction and the phase of application [FL04, Tho06a]. By presenting insights in reference enterprise architecture development we contribute to the first phase, i.e. the construction of reference models. Research discusses two generic strategies for reference model construction. While the deductive reference modeling derives reference models from generally accepted knowledge, the inductive approach abstracts from individual models to agree on a common understanding within the reference model [BS97]. Regarding Ardalani et al. most of the established reference models have been developed based on deductive approaches since only a few inductive approaches exist [Ard13]. Still, inductive reference modeling provides potential because more and more relevant data in terms of logs and concrete information models of organizations are available. Further, Rehse et al. point out that inductively developed reference models tend to have a higher degree of detail, are more mature and seem to be more accepted when it comes to reference model application [Reh16].

One available method for inductive reference model construction is proposed by Fettke [Fet14]. In seven steps the method (i) defines the reference model's requirements, (ii) collects the individual models and (iii) pre-processes them before (iv) deriving an initial reference model. After (v) post-processing the resulting reference model it has to be (vi) evaluated from different perspective and (vii) maintained since the integration of new individual models may enhance the reference model. In the fourth step a reference model is abstracted from the set of presorted individual models. In the literature no general approaches exist how to perform this abstraction. Rehse et al. suggest to apply abstraction techniques from business process mining since the majority of reference models are based on business process model structures [RFL13]. Especially developed for inductive reference process model development, Ardalani et al. propose the *minimal cost of change (MCC)* approach, which is based on the idea of minimized graph edit distance [Ard13].

Based on the former findings, it can be derived that most contributions in inductive reference modeling research are focused on reference models that follow the structure of business process models. Our work tries to enhance this circumstance by applying existing abstraction techniques to other reference model structures, i.e. reference models based on the structure of enterprise architectures. In concrete, we apply the MCC approach to individual gathered enterprise architectures in a certain domain. We chose the MCC approach since it is the best documented approach. Hence, the following sections explain the MCC approach in more detail and reveal the characteristics of enterprise architectures before documenting the MCC's application in the main part of our work.

2.2 The Minimal Costs of Change Approach

In their work Ardalani et al. present an approach for the inductive development of business process reference models [Ard13]. The approach is called "minimal cost of change" (MCC) and is based on a minimized graph edit distance in order to derive a reference model from a given set of individual process models that address the same process. These individual process models represent real-world models from, e.g., a certain enterprises of an industry domain. The MCC approach can be used to develop a reference process model from these models. According to Fettke and Loos, reference models can provide common, good, or even best practice [Fet07]. Ardalani et al. say that reference models, which were developed inductively, mainly serve for developing reference models of common practice nature, since their basis lies in existing individual models, whose quality cannot always be assured but their practical usage. Hence, they claim that the MCC approach serves for inductively developing common practice reference process models [Ard13]. The approach is thoroughly presented by explicit definitions, calculation formulas, algorithms, an abstract example and a software prototype for its realization. All mentioned aspects relate to event-driven process chains (EPC) as a representation language for process models [KNS92].

In this work we apply the MCC approach to the enterprise architecture domain. In order to increase comprehensibility for this endeavor, this section presents the MCC approach by means of used concepts and its procedure. Still, we encourage the reader to get acquainted with the work in [Ard13] for a deeper understanding.

An EPC model consists of events and functions, which are related to each other by means of connectors. Connectors can be of different type, like “and”, “or” or “xor”. EPCs are digraphs and represent control flows. Thus, each event or function has at least one predecessor and successor (except start and end points). By definition, a function only can follow an event and vice versa [KNS92]. The MCC approach defines the *EPCsSet* as the pool of all individual process models used for the reference model development. The general idea behind MCC is to decide whether the integration costs of a certain model element from the EPCsSet into the reference model justifies its inclusion. This decision is made by comparing this *cost value* with a pre-defined *threshold value*. Since a model element can occur in several individual models having different predecessors, successors or connection types, the cost values are not calculated for a single model element, but for a relation consisting of a model element and its predecessor. Summarizing the approach, the cost value is calculated by a formula which depends on four factors:

i. the element frequency *f(element)*, i.e. how often a function or event occurs in the pool of individual models. E.g. gets 0,4 when occurring in 40% of the models

ii. the relation frequency *f(relation)*, i.e. how often a function or event has the same predecessor in the pool of individual models.

iii. the cost functions *cost(ins)*, *cost(del)*, *cost(mov)*. Ardalani et al. assume costs occur when inserting, deleting or moving a function or event in the reference process model. They define cost(ins)=10, cost(del)=1 and cost(mov)=5.

iv. the existence of elements’ *preelement* in the reference model. If a function’s or event’s predecessor is already integrated in the reference model, its integration costs will decrease. In this case the function *exist(preelement)* returns 1, otherwise 0 is returned.

Ardalani et al. define the formula, which is used in order to calculate the integration costs of a relation from the individual models, as follows [Ard13, p. 5]:

$$costValue(relation) = f(element)*cost(ins) - (\, f(element) - f(relation)\,)*cost(mov) \\ - (\, 1\text{-}f(relation)\,)*cost(del) - (\, 1\text{-}exist(preelement)\,)*f(relation)*cost(mov) \quad (1)$$

At this point we want to stress, that according to Ardalani et al. the value of *costValue* represents the costs, that can be saved when adding an element to the reference model [Ard13, p.5]. The higher this value, the more relevant is the element for the reference model. Thus, an element is integrated into the reference model when it exceeds the threshold value. Further, the MCC approach makes several assumptions for its usage. First, they define the same costs for inserting, deleting and moving for different elements. Second, the costs of inserting, deleting and moving can be predefined. Third, the approach requires a stable namespace, i.e. the same EPC event has the exact same label in each individual model.

Based on these definition and formula (1) the MCC approach provides an algorithm how the reference model is developed. The algorithm comprises three main steps, in which the individual models are processed and assigned to different kind of sets (e.g. the EPCsSet) during the algorithm's iteration. In *Step 1* the *ElementsSet* and *RelationsSet* are initialized. While the former parses the individual models for events and functions together with their frequency of occurrence in the EPCsSet, the latter stores the element (i.e. event or function), its preelement and the type of connection. Since each EPC starts with one event, this start event's preelement is defined as "*null"*. Afterwards the costValue is calculated for each entrance of the RelationsSet using formula (1). At this point, no element exists in the reference model, thus, exist(preelement) returns 0 for each calculation.

In *Step 2* the MCC approach iteratively picks a relation from the RelationsSet and integrates it into the reference model. Each iteration works as follows: First, the relation with the highest costValue is selected. Second, it will be decided, whether the relation is inserted into the *ReferenceModelRelationsSet* (i.e. the set defining the reference model) or into the *ReservedRelationsSet*. The latter stores all relevant relations, where the preelement is not part of the reference model yet. If the preelement of the relation at hand is in the ReferenceModelRelationSet, the relation will be added there as well. Otherwise, the relation is added to the ReservedRelationsSet. Although [Ard13] do not specify it explicitly we can assume, that a relevant with a "null" preelement is added to the ReferenceModelRelationsSet. Third, after the relation is added to the reference model the ReservedRelationsSet is checked for successors of the reference model elements, which will then be integrated. Fourth, the RelationsSet is updated, since changes in the ReferenceModelRelationsSet cause changes of the costValues. This loop is traversed until the threshold value is reached or all relations of the RelationsSet are integrated.

In *Step 3* the reference model is created by the entries of the ReferenceModelRelationsSet. Some problems may occur when deriving the reference model from this set. For example, the right connectors have to be used when multiple elements have the predecessor. Therefore, the MCC approach defines several rules that intend to avoid such conflicts [Ard13].

2.3 Enterprise Architecture Management

Architecture is defined as a fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principle guiding the organizations design and evolution. An Enterprise Architecture (EA) is the formal declaration of the basic structures of an organization, its components and relations, as well as the processes used for development [Lan17]. Through EA models the complex interrelations between an enterprise's organizational and operational structure with used information systems, processed data and realizing technologies are made explicit. Such models consist of layers and elements, which define different perspectives on the enterprise. Therefore, [WF08] define essential layers, of which an enterprise architecture consist: business architecture, process architecture, integration architecture, software architecture and infrastructure architecture.

In this context, Enterprise Architecture Management (EAM) can be seen as a management discipline, which is integrated in IT management, business management or even can be seen as a separate discipline within an enterprise [ASM12]. It aims to provide a powerful approach for a systematic development of the enterprise in accordance with its strategic visions, yet its value depends on the organizational ability to perform EAM effectively. As a management philosophy, it is a holistic way to understand, plan, develop control and adjust organizations architecture. As an organizational function it enables and improves existing strategic planning and implementation processes. As a methodology and culture it represents an open approach among the managers and proposes a set of management practices in order to reach a global optimum for the firm, free of egoism and opportunism [ASM12].

In order to develop an EAM endeavor, enterprise can draw on a variety of EA frameworks, which are thoroughly analyzed in [Mat11]. In our work we use the TOGAF framework [The10] as it is widely accepted among practitioners and comes with an detailed modeling language specification ArchiMate [The15] as well as an open source modeling tool Archi[3]. ArchiMate is a well-documented modeling notation and currently accessible in version 3.0 [The16]. For this work we utilized the former version 2.1, which was the latest version when we started our endeavor. For modeling the ArchiMate Core Meta-Model was used, which constitutes of the business layer, application layer and technology layer. For each layer, ArchiMate provides meta models for each of these layers and further specifies relations among them. Each meta model consists of architecture elements, e.g. the business process and business role element on the business layer or the application component element on the application layer. Further, the language introduces the concept of architecture viewpoints, which are projections of the EA model, each considering a certain purpose by addressing different stakeholders' interests. For example, the business function viewpoint reveals defined responsibilities between business roles and business functions for process architects [The15].

3 Application in Reference Enterprise Architecture Development

This chapter gives an introduction into the application of the MCC approach described above to EA models. Instead of calculating formula (1) on individual EPC models, we applied them to a pool of EA models, which came from enterprises of the same domain and focus on the same phenomenon. As depicted earlier, EA models can be represented by different viewpoints, which are projections of the EA model with a certain intention

[3] See http://www.archimatetool.com.

[Lan17]. The Open Group defines a set of standard viewpoints that summarize common utilization of the viewpoint concept [The15]. Before demonstrating our results, we shortly explain the individual EA models at hand. They were acquainted during a project, which is located in the financial domain. The project aims to develop a reference model that guides financial institutes how to effectively and efficiently implement a compliance organization into their organizational structures. Therefore, the reference model is based on the structure of enterprise architectures. The individual models we used for applying the MCC approach consider how financial institutes identify new industrial customers in order to comply with the German Money Laundering Act [Ger08]. The models were gathered by dint of 16 structured interviews with responsible persons from the particular institutes. Afterwards, all interviews were transferred to EA models using the same modeling structure and guidelines, which was done according to Lankhorst [Lan17]. Each individual EA model was structured by six different ArchiMate viewpoints, that displayed different aspects of the EA models. In the following we concentrate on one specific viewpoint of these resulting 16 individual EA models – the "Oboarding Procedure". All activities described in section 3.1 were performed for each viewpoint, which resulted in the final reference model for industrial customer identification.

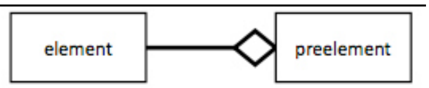
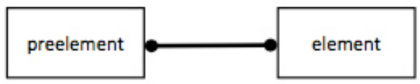
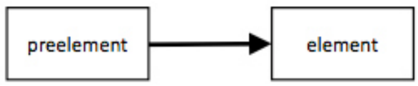
3.1 Applying MCC for inductive R-EA development

Before describe the MCC application we first want to explain the viewpoint of the individual models we use for it. Therefore, we use the ArchiMate standard viewpoint *business process viewpoints*, which intends to reveal main activities, its flow and related responsibilities in a financial institute when new industrial customers are identified. We call this viewpoint "*Onboarding Procedure"*. We therefore adjusted the standard viewpoint to our purposes and used the following model elements: *business function* (grouping behavior), *business event* (something that happens and influences behavior) and *business role* (the responsibility for performing a behavior). These ArchiMate elements can be related to each other by dint of different relationship types: *assignment relation* (a role is assigned to a function), *triggering relation* (an event or function triggers another function) and *aggregation relation* (a function groups other functions).

In order to enable the applicability of the MCC to the EA model viewpoint, we made the following adjustments:

- In contrast to EPCs, EA models have a variety of relation types, from which not all clearly define that one element is a preelement of another. Since [Ard13] require this from a relation, we define a preelement for each ArchiMate relation type. Tab. 1. provides an extraction of these consideration due to space limitations. We further differentiated between the elements that take part in the relationship, e.g. a business role is the preelement when being assigned to a business process or function.

- EA models do not follow such a strict control flow like EPC models. Thus, we changed *cost(mov)* from 5 to 2. While EPC models require a strict order of elements, EA models only do partially.
- In line with [Ard13] we assume, that identical model elements of the individual EA models have the identical label. This was ensured, since we firstly compiled a model element library when identifying new elements and developed each EA model from this library.

Connector Type	*Visualization*
Aggregation Relation (preelement aggregates element)	
Assignment Relation (preelement is assigned to element)	
Triggering Relation (preelement triggers element)	

Tab. 1: Assigning preelements to used ArchiMate relations

After these pre-adjustments the three steps of the MCC's algorithm were performed. This happened completely manually, since no tool was accessible or implemented for this endeavor. In Step (1) of the MCC the *RelationsSet* was initialized by traversing all 16 individual viewpoints and identifying all existing relationships in the EA models. For each relationship identified its frequency of occurrence (*f(relation)*) and its elements' frequency of occurrence (*f(element)*) was calculated. On this basis, the *costValue* for each relation was initially calculated. This was done using Excel sheets. In Step (2) the MCC loop for integrating relevant model elements into the *ReferenceModelRelationsSet* was traversed. Initially, we defined the threshold value with 50%, i.e. 0,5. We decided so in order to only included model elements, that were mentioned by at least every second financial institute. After interpreting the final *RelationsSet* we then decided to drop the threshold value to 30%, since no business role would have been included into the reference model. This is due to the high diversification of organizational structure in the different institutes. The resulting *RelationsSet* is shown in Tab. 2 . The entries above the bold line indicate all relations, that were integrated into the *ReferenceModelRelationsSet*, which resulted in the reference model. The table shows the round of MCC iteration, the element name, its preelement and the respective relation type as well as *f(element), f(relation)* and *costValue*.

Round	element	preelement	relation	f(element)	f(relation)	costValue
1	New Industrial Customer	(null)	---	0,93	0,93	9,27
2	Customer Identification	New Industrial Customer	triggering	1	0,93	9,8
3	Due Diligence	Customer Identification	triggering	1	0,93	9,8
4	Assessment of Customer Relation	Due Diligence	triggering	1	1	10
5	Customer Due Diligence (CDD)	Due Diligence	aggregation	1	1	10
6	Rejection of Customer	Assessment of Customer Relation	triggering	1	0,93	9,8
7	Acceptance of Customer	Assessment of Customer Relation	triggering	1	0,93	9,8
8	Conduction of Identification (Legal Person)	Customer Identification	aggregation	0,93	0,93	9,27
9	PEP-Screening	Assessment of Customer Relation	aggregation	0,93	0,93	9,27
10	Sanctions Screening	Assessment of Customer Relation	aggregation	0,93	0,93	9,27
11	Conduction of Identification (Natural Person)	Customer Identification	aggregation	0,87	0,87	8,53
12	Verification of Customer Data	Customer Identification	aggregation	0,8	0,8	7,8
13	Enhanced Due Diligence (EDD)	Due Diligence	aggregation	0,8	0,8	7,8
14	Simplified Due Diligence (SDD)	Due Diligence	aggregation	0,8	0,8	7,8
15	Risk Assessment of Customer	Assessment of Customer Relation	aggregation	0,8	0,73	7,6
16	Due Diligence	Risk Assessment of Customer	triggering	1	0,07	7,2
17	New Deputy of Existing Customer	(null)	---	0,6	0,6	5,6
18	Customer Identification	New Deputy of Existing Customer	triggering	1	0,53	8,6
19	Interest of Customer in Product	(null)	---	0,6	0,6	5,6
20	Customer Identifiaction	Interest of Customer in Product	triggering	1	0,53	8,6
21	Risk Assessment of Customer	Customer Identification	triggering	0,73	0,07	5,07
22	Regular Customer Monitoring	(null)	---	0,53	0,53	4,87
23	Customer Identification	Regular Customer Monitoring	triggering	1	0,4	8,2
24	Due Diligence	Regular Customer Monitoring	triggering	1	0,07	7,2
25	Product Specific Identification	Customer Identification	aggregation	0,47	0,47	4,13
26	Employee Compliance	Assessment of Customer Relation	assignment	0,4	0,33	3,2
27	Customer Risk assessed	Assessment of Customer Relation	aggregation	0,4	0,33	3,2
28	Employee Compliance	Due Diligence	assignment	0,4	0,27	3
29	Anti Money Laundering Officer	Due Diligence	assignment	0,33	0,2	2,27
30	Anti Money Laundering Officer	Assessment of Customer Relation	assignment	0,33	0,2	2,27
31	Customer Account Manager	Due Diligence	assignment	0,27	0,27	1,93
32	Anti Money Laundering Officer	PEP-Screening	assignment	0,33	0,07	1,87
33	Anti Money Laundering Officer	Sanctions Screening	assignment	0,33	0,07	1,87
34	Customer Account Manager	Customer Identification	assignment	0,27	0,13	1,53
35	Customer Account Manager	Assessment of Customer Relation	assignment	0,27	0,07	1,33
36	Customer Account Manager	Risk Assessment of Customer	assignment	0,27	0,07	1,33
37	Corporate Account Officer	Customer Identification	assignment	0,2	0,2	1,2
38	Employee Back Office	Customer Identification	assignment	0,2	0,2	1,2
39	Chief Compliance Officer	Due Diligence	assignment	0,2	0,13	1
40	Chief Compliance Officer	Assessment of Customer Relation	assignment	0,2	0,13	1
41	Decision over Customer Relation	Assessment of Customer Relation	aggregation	0,2	0,13	1
42	Rejection of Customer	Decision over Customer Relation	triggering	1	0,13	7,4
43	Rejection of Customer	Decision over Customer Relation	triggering	1	0,13	7,4
44	Customer Risk assessed	Decision over Customer Relation	triggering	0,4	0,07	2,4

Tab. 2: The final RelationsSet for the viewpoint "Onboarding Procedure"

In the application of Step (3) we modeled the resulting EA model viewpoint based on the resulting *ReferenceModelRelationsSet* from the previous step. The resulting ArchiMate model was then further analyzes, since the MCC approach may exclude relevant model elements. This was the case for several business roles like the anti-money laundering officer. These model elements were found by further investigating the interviews' transcripts as well as other literature like the German law from [Ger08]. The resulting reference model is shown in Fig. 1. To shortly explain the resulting reference model, one can see several functions the onboarding procedure consists of, namely the customer identification, due diligence and the assessment of the customer relation. Each function aggregates sub-functions, which are not explained here in more detail. Customer Account Manager, Anti-Money Laundering Officer and Compliance Employee as well as the Chief Compliance Officer are assigned to the main functions. The customer identification is triggered by different events and the assessment of the customer relation triggers three different events.

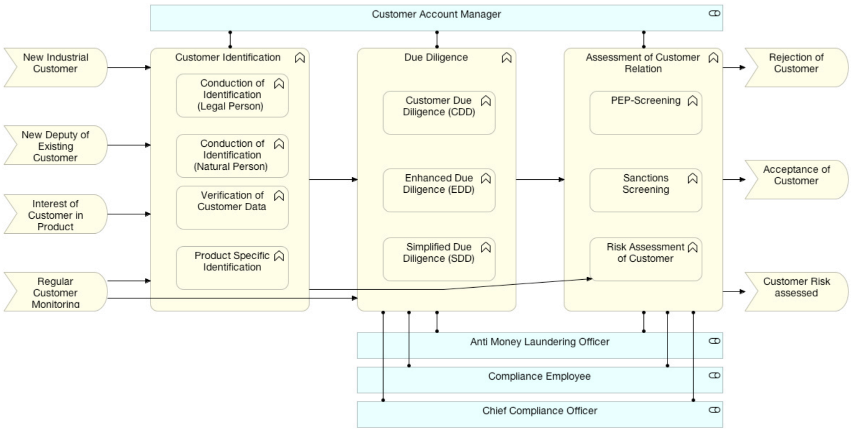


Fig. 1: Resulting Reference Model for Viewpoint "Onboarding Procedure"

4 Conclusion

In order to facilitate the inductive development of reference enterprise architectures, our work applies the "Minimal Cost of Change" approach by Ardalani et al., which abstracts from individual process models to a reference process model based on the idea of a minimized graph edit distance [Ard13]. Although initially developed for the business process domain, our work shows, that it can be applied to EA models by dint of several adjustments of the approach itself and the definition of EA models (without change EA model structure itself). The main findings are as follows: EA models do not share the strict control flow order like process models and define much more types of interrelations among the EA model elements. Thus, each type has to be investigated how to be processed during the MCC application. Further, EA models are represented by a set of viewpoints. Consequently, the development of a reference enterprise architecture requires several MCC iterations, each for one viewpoint. Another aspect is, that the costs for inserting, moving and deleting have a different influence when developing reference models in the EA domain. All in all, we assess the MMC approach applicable to the inductive development of a reference enterprise architecture. Still, there are many aspects to be considered in future research. EA model relations should thoroughly be investigated - the role of preelements as well as whether the costs of change differ among the relation types. Further, the definition of thresholds currently is very vague and it needs to be discussed whether its value should change between different EA viewpoints. Like for business process models, a tool that implements the MCC approach for EAM would support the development process. Likewise to the limitations mentioned in [Ard13] the integration of semantic measurement methods when calculating the frequency of occurrence of a model element in the pool would enhance the method. From a more general perspective on this topic other abstraction techniques, e.g. the ones mentioned in [RFL13], should be investigated regarding their applicability to EAM, too.

References

[ASM12] Ahlemann, F.; Stettiner, E.; Messerschmidt, M.: Strategic Enterprise Architecture Management. Challenges, Best Practices, and Future Developments. Springer Berlin Heidelberg, Berlin Heidelberg, 2012.

[Ard13] Ardalani, P. et al.: Towards a minimal cost of change approach for inductive reference model development. In ECIS 2013 - Proceedings of the 21st European Conference on Information Systems, 2013.

[BS97] Becker, J.; Schütte, R.: Referenz-Informationsmodelle für den Handel: Begriff, Nutzen und Empfehlungen für die Gestaltung und unternehmensspezifische Adaption von Referenzmodellen. In (Krallmann, H. Hrsg.): Wirtschaftsinformatik '97. Internationale Geschäftstätigkeit auf der Basis flexibler Organisationsstrukturen und leistungsfähiger Informationssysteme. Physica-Verlag HD, Heidelberg, 1997; S. 427–448.

[FL04] Fettke, P.; Loos, P.: Referenzmodellierungsforschung. In WIRTSCHAFTSINFORMATIK, 2004, 46; S. 331–340.

[Fet07] Fettke, P. Hrsg.: Reference modeling for business systems analysis. Idea Group Publ, Hershey Pa. u.a., 2007.

[Fet14] Fettke, P.: Eine Methode zur induktiven Entwicklung von Referenzmodellen: Tagungsband Multikonferenz Wirtschaftsinformatik 2014, MKWI 2014, 2014; S. 1034–1047.

[Ger08] German Goverment: Gesetz über das Aufspüren von Gewinnen aus schweren Straftaten (Geldwäschegesetz - GwG). https://www.gesetze-im-internet.de/bundesrecht/gwg_2008/gesamt.pdf.

[KNS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)". Saarbrücken, 1992.

[Lan17] Lankhorst, M.: Enterprise Architecture at Work. Modelling, Communication and Analysis. Springer Berlin; Springer, Berlin, 2017.

[Mat11] Matthes, D.: Enterprise Architecture Frameworks Kompendium. Über 50 Rahmenwerke für das IT-Management. Springer-Verlag Berlin Heidelberg, Berlin Heidelberg, 2011.

[RFL13] Rehse, J.-R.; Fettke, P.; Loos, P.: Eine Untersuchung der Potentiale automatisierter

Abstraktionsansätze für Geschäftsprozessmodelle im Hinblick auf die induktive Entwicklung von Referenzprozessmodellen. In (Rainer Alt; Bogdan Franczyk Hrsg.): Proceedings of the 11th International Conference on Wirtschaftsinformatik. Internationale Tagung Wirtschaftsinformatik. Internationale Tagung Wirtschaftsinformatik (WI-2013), February 27 - March 1, Leipzig, Germany. tba, 2013.

[Reh16] Rehse, J.-R. et al.: Inductive Reference Model Development. Recent Results and Current Challenges. In (Mayr, H. C.; Pinzger, M. Hrsg.): INFORMATIK 2016. Jahrestagung der Gesellschaft für Informatik (INFORMATIK-2016), September 26-30, Klagenfurt, Austria. GI, Bonn, 2016.

[The10] TOGAF Version 9. Van Haren Publishing, Zaltbommel, 2010.

[The15] The Open Group: ArchiMate® 2.1 specification. Open Group Standard. Van Haren Publishing, Zaltbommel, 2015.

[The16] ArchiMate 3.0 specification. Open Group standard, 2016.

[Tho06a] Thomas, O.: Management von Referenzmodellen. Entwurf und Realisierung eines Informationssystems zur Entwicklung und Anwendung von Referenzmodellen. Zugl.: Saarbrücken, Univ., Diss., 2006. Logos-Verl., Berlin, 2006.

[Tho06b] Thomas, O.: Understanding the Term Reference Model in Information Systems Research: History, Literature Analysis and Explanation. In (Bussler, C. Hrsg.): Business process management workshops. BPM 2005 international workshops, BPI, BPD, ENEI, BPRM, WSCOBPM, BPS, Nancy, France, September 5, 2005 ; revised selected papers. Springer, Berlin, 2006; S. 484–496.

[TSF17] Timm, F.; Sandkuhl, K.; Fellmann, M.: Towards A Method for Developing Reference Enterprise Architecture. In (Leimeister, J. M.; Brenner, W. Hrsg.): Proceedings der 13. Internationalen Tagung Wirtschaftsinformatik (WI2017), St.Gallen, 2017; S. 331–345.

[Bee12] van der Beek, Wijke ten Harmsen; Trienekens, J.; Grefen, P.: The Application of Enterprise Reference Architecture in the Financial Industry. In (Aier, S. et al. Hrsg.): Trends in Enterprise Architecture Research and Practice-Driven Research on Enterprise Transformation. 7th Workshop, TEAR 2012, and 5th Working Conference, PRET 2012, Held at The Open Group Conference 2012, Barcelona, Spain, October 23-24, 2012. Proceedings. Springer, Berlin, Heidelberg, 2012; S. 93–110.

[Bro03] Vom Brocke, J.: Referenzmodellierung. Gestaltung und Verteilung von Konstruktionsprozessen. Univ., Diss.--Zugl.: Münster, 2002. Logos, Berlin, 2003.

[WF08] Winter, R.; Fischer, R.: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture: Proceedings / 10th IEEE International Enterprise Distributed Object Computing Conference. [workshops] ; Hong Kong, China, October 16 - 20, 2006. IEEE, Piscataway, NJ, 2008; S. 30.