Rules for Validation of Models of Enterprise Architecture
Rules of Checking and Correction of Temporal Inconsistencies among Elements of the Enterprise Architecture

Amiraldes Xavier, André Vasconcelos and Pedro Sousa
INESC-ID, IST, Universidade de Lisboa, Portugal

Keywords: Enterprise Transformation, Enterprise Architecture, Rules, Inconsistencies, As-Is, To-Be.

Abstract: The organizational structure of the elements in an enterprise architecture model is key for decision-making and business transformation. Over time, it is possible that the relationships among the elements of the EA (Enterprise Architecture) become inconsistent in the EA model. To address this problem in this research we specify a set of temporal rules divided into two categories: rules for verification and rules for inconsistencies correction. The specification of these rules is based on the states of the elements of the EA and the concepts presented by the ArchiMate metamodel. The Rules are translated into logical expressions in order to make them easier to implement. This research was developed on the basis of a concrete study of an enterprise architecture management tool, but the solution proposed can be adapted to any EA management tool.

1 INTRODUCTION

The enterprise architecture (EA) provides a holistic view of organizations, including several viewpoints, such as business, information, systems and technology (Ylimäki et al, 2007). In general, these viewpoints are represented through modelling. The process of modelling consists in the representation, through an appropriate language, of the elements and relationships among EA elements.

EA is an approach to control and manage the complex and constant transformations in the organization and its business environment, assisting organizations in conducting a multitude of positive impacts at various organizational levels. Such impacts can be verified at administrative level, in decision-making, as well as the economic level because it allows better management and control of resources (Ylimäki. et al., 2007).

In many cases, these transformations are influenced by temporal factors that are associated with the elements that constitute the model of the EA.

These temporal factors are responsible for defining the states of the elements within the model as "alive" for the elements that are functional and "dead" when they stop working. As time passes, the state of the EA elements can be changed, influencing right way the operation of other elements linked by certain relationships, influencing the operation (performance) of the architecture as a whole.

It is essential that, during the transformation process to check the completeness of the models, to assess the compliance to a set of (recommended) rules, in order to avoid problems in managing and controlling the EA.

1.1 Problem

It is important for organizations to follow the EA evolution over time. For this purpose, organizations use EA management tools that allows to have a perspective of temporal evolution of the organization through the ability they have to define states for the elements that make up an organization and the moment that they join these states. Time is an important attribute to assess defines when an element enters to a state, in order to ensure the validity of relationships among the elements taking account to the sort of functional dependence among these elements.

For example, if an application is abandoned at one time, and this occurs before it is possible, the service that may depended on that Application will stop working.

These problem occur because the EA

Xavier, A., Vasconcelos, A. and Sousa, P.
Rules for Validation of Models of Enterprise Architecture - Rules of Checking and Correction of Temporal Inconsistencies among Elements of the Enterprise Architecture.
DOI: 10.5220/0006187503370344
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 3*, pages 337-344
ISBN: 978-989-758-249-3

management tool don't have the temporal evaluation mechanism of the models. And these problems affect the quality of the decision-making related to the enterprise transformation process.

In order to solve this problem, the following issue is raised: how to ensure the consistency of the relationship among elements in the EA modes serving its evolution?

1.2 Goals

Look upon to the problem introduced in the previous section, the main goal of this research is to define a set of rules that allows the verification and correction of inconsistencies in the relationship among EA elements to the extent that they will evolve over time (change of State). To have this purpose achieved it is fundamental the achievement of the following specific goals: (1) specify temporal rules considering the relationship among elements, (2) formalize the rules in an appropriate language to facilitate the implementation in EA management tools.

In section 2 we present the theoretical background that focusses on three main concepts, which are

ArchiMate, Enterprise Transformation and Enterprise Cartography. In the following section (3) we present the solution proposal including the general definition of the temporal rules, the definition and formalization of the inconsistencies verification and correction rules. In the section 4 we present this research evaluation. Finally, in section 5, we synthesize the conclusions and future work to be developed.

2 RELATED WORK

2.1 ArchiMate

ArchiMate is an open and independent modelling language of EA which is supported by different software vendor's tools and consulting firms. ArchiMate provides a notation that allows enterprise architects to describe, analyse and visualize the relationships among domains (Braun and Winter, 2005). It is currently a framework of EA that is part of the technical standards of the Open Group.

ArchiMate offers a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems, and technical infrastructure. This insight helps stakeholders to design, evaluate and communicate the consequences of decisions and changes within and among these business domains.

ArchiMate uses the concept of layered views to present the service-oriented model, where the layers above make use of services that are provided by the lower layers (Lankhorst, 2004).

The concepts (types of elements and relationships) defined in the specification of the ArchiMate (Iacob et al, 2012), serve as a basis for the development of the proposed solution presented in this article.

2.2 Enterprise Transformation

The transformation is seen as a set of initiatives that change the domain of the organization from its current state (As-Is) for a desired state (To-Be). These states consist of a representation of organizational elements in different periods of time. The As-Is corresponds to the state whose elements have changed due to past events. The To-Be corresponds to the expected state of the organizational elements. And among these two states, the company reacts to other events that are triggered by the transformation processes (Tribolet et al, 2014).

So to control and coordinate this process of enterprise transformation several authors propose the Enterprise Architecture Management (EAM) (Ahlemann et al, 2012; Aier and Gleichauf, 2010; Harmsen et al, 2009). The main purpose of EAM is to provide a high-level overview of a company, involving aspects such as business and information technology (IT) and especially the dependence among them. EAM provides solid bases of models and methods to support the analysis, planning and design of organizations from a "business-to-IT" perspective (Abraham and Aier, 2012).

Three modelling techniques of enterprise transformation are covered: activity modeling, modeling and life cycle modeling (that is the focus of this paper) (Buckl et al., 2011).

A life cycle indicates that a single element of the EA evolves over the time going through different stages (states) where we can highlight the "conception", "alive" and "retired/dead".

According to Buckl et al (Buckl et al., 2011) the most basic way to modelling temporal aspects of EA elements is to assign validity periods for each element individually. In projects carried out through the EAMS tool this approach is also applied to the metamodel of the ArchiMate by assignment of attributes such as "Begin Date" (BD) and "End

Date" (ED) in its elements. Figure 1 presents an approach to modelling the temporal aspects of temporal elements defined by Buckl.

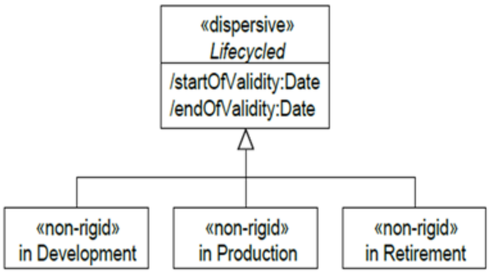

Figure 1: temporal aspects of modelling elements of EA (Buckl et al., 2011).

With the attributes defined above it is possible to determine a set of restrictions on the terms of validity of any phase/state is valid only in a limited period of time by the corresponding start and end activities. These restrictions can influence right on the relationship among the EA elements.

For Buckl et al (Buckl et al., 2011), these restrictions are not necessary for all types of relationships. Therefore, the same authors make the distinction into two types of relationships for their temporal qualities:

- Synchronic Relationship: consists of the relationships that are valid only among elements whose validity periods intersect.
- Diachronic Relationship: are relationships that are valid regardless of the validity periods of the elements, here the issue of intersection of the periods is not taken into account.

2.3 Enterprise Cartography

Cartography is the practice of design and creation of maps. Having regard to this assumption, Tribolet et al., define enterprise cartography as the design, production and dissemination of business maps to support their analysis and collective compression (Tribolet et al., 2014).

Of the eight principles of enterprise cartography presented in (Tribolet et al., 2014) the sixth sets that all organizational articles can be categorized as being in one of four States invariants:

- Gestating: the state describes an artefact after designed, that is, once you start being planned or produced;
- Alive: is the state in which an artefact enters after the birth. This means that the designed product is now able to produce a behavior as part of transactions and organizational processes;
- Dead: it’s when an artefact in gestation or alive is disabled in the sense that it is no longer able to play a role in transactions and organizational processes;
- Retired: is the after-death state, where the artefact is unable to interact with other artefacts.

3 SOLUTION PROPOSAL

3.1 Solution Overview

Considering the related work presented in the previous sections, we argue that the elements that form the basis for the specification of rules are: states of elements, temporal attributes and the level of dependency among the EA elements that is defined by the relationship type among these elements.

The proposal to solve the problem presented in the section 2 of this paper can be summarized on the conceptual map presented in Figure 2.

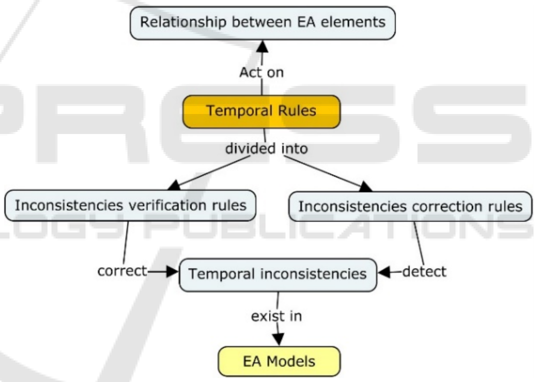

Figure 2 Proposed solution overview.

In the figure 2, we can see that the solution is implemented by creating rules that act on EA models (properly the relationship among the elements).

The temporal rules are divided into two categories: (i) Inconsistencies Verification Rules (IVR), which serves to verify temporal inconsistencies, and (ii) Inconsistency Correction Rules (ICR), that serve to correct temporal inconsistencies in models through the transformation of those models (temporal adjustment process). The temporal rules are created from the combination of the states of the elements and EA relationships. Temporal rules for its specifications can only be applied to elements.

3.2 General Definition of Temporal Rules

We call temporal rules the rules that define, in terms of the EA elements, states through its temporal attributes and can be used to verify or correct inconsistencies in the relationship among EA elements. To define these rules, we consider the characteristics of each relationship. Table 1 present the general definitions of the temporal rules.

The discontinuation rule is derived from the insertion in the EA model of “discontinuation” state. This state appears as proposed in the scope of this article and serves to identify the moment (Discontinuation Begin Date – DBD) in which an EA element enters the state of discontinuation.

3.3 Inconsistencies Verification Rules

The IVR are rules that, as presented before, have the role of verifying the existence or not of inconsistencies in the relationship among elements on the basis of the time evaluation.

The inconsistencies verification is accomplished by applying in the model the mathematical logic expressions that make the combination of the elements used to assess inconsistencies; the result "true" implies the non-existence of inconsistency which means relations among the elements analysed are correct. The result "false" implies the existence of inconsistency in the relationship.

Depending on the relationship reading mode, two versions of formalization and definition for each rule are presented. To distinguish them, we use A → B

Table 1: General definition of the temporal rules.

Rule Name	Rule Definition
Composition Rule	The composition relationship is valid only if the elements in them are involved ("whole" and the "parts") are in the "alive" state in the same time period.
Aggregation Rule	In aggregation relationship the “whole” element enters to the state "alive" when all the elements "parts" are ins "alive" state.
Create Rule	The create relationship, the created element should only go to the “alive” state as creative element is in “alive” state.
Synchronic rule for Relationship	In the relationship amongA and B where the A acts on B and B depends on A so B should only go to the "alive" state and keep in this state while A is "alive"
Discontinuation Rule	The new elements added to EA cannot relate to elements in a state of discontinuation.

Table 2: Definition and formalization of Composition IVR.

C-IVR1 – Definition	If A is composed of B, for this relationship to be valid the A "Begin Date" must be equal to B "Begin Date” and the A "End Date” must be equal to the B "End Date".
Formalization	$A \rightarrow B = \{BD_A = BD_B \wedge ED_A = ED_B\}$
C-IVR2 - Definition	If B composes A, for this relationship to be valid the B "Begin Date" must be equal to A "Begin Date" and B "End Date" must be equal to the A "End Date".
Formalization	$B \rightarrow A = \{BD_B = BD_A \wedge ED_B = ED_A\}$

Table 3: Definition and formalization of the Aggregation IVR.

A-IVR1 - Definition	If A Aggregates B elements, this relationship is only valid if the A "Begin Date" is greater or equal than to the greater "Begin Date" among B elements and the A "End Date" must be less or equal than the less "End Date" among the B elements.
Formalization	$A \rightarrow B = \{BD^A >= \max (BD_{(B1, B2 \ldots Bn)}) \wedge ED_A <= \min (ED_{(B1, B2 \ldots Bn)})\}$
A-IVR2 - Definition	If B is Aggregated by A, this relationship is valid if the B "Begin Date" is less or equal than A "Begin Date" and the B "End Date" is greater or equal than to the A "End Date".
Formalization	$B \rightarrow A = BD_B <= BD_A \wedge ED_B >= ED_A$

Table 4: Definition and formalization of the Create IVR.

Cr-IVR1- Definition	If A creates B, this relationship is valid only if the A "End Date" is greater or equal than B "Begin Date".
Formalization	$A \rightarrow B = ED_A >= BD_B$
Cr-IVR2- Definition	If B is created by B, this relationship is valid only if the B "Begin Date" is less or equal than A "End Date".
Formalization	$B \rightarrow A = BD_B <= ED_A$

Table 5: Definition and formalization of Synchronic Relationship IVR.

SR-IVR1- Definition	If A acts on B, this relationship is valid if A "Begin Date" is less or equal than B "Begin Date" and A "End Date" must be equal or greater than B "End Date".
Formalization	$A \rightarrow B = BD_A <= BD_B \wedge ED_A >= BD_B$
SR-IVR2- Definition	If B is actuated by A, this relationship is valid if B "Begin Date" greater or equal than A "Begin Date" and B "End Date" is less or equal than B "End Date".
Formalization	$B \rightarrow A = BD_B >= BD_A \wedge ED_B <= BD_A$

Table 6: Definition and formalization of Discontinuation IVR.

D-IVR1 - Definition	On the right relationship among new element (B) and an existing one (A), is only valid if A " Discontinuation Begin Date" is less than the B "Begin Date".
Formalization	$A \rightarrow B = DBD_A < BD_B$
D-IVR1 - Definition	On the inverse relationship among new element (B) and an existing one (A), is only valid if B "Begin Date" is greater than A "Discontinuation Begin Date"
Formalization	$B \rightarrow A = BD_B > DBD_A$

Table 7: Definition and formalization of Composition ICR.

First case	$A \rightarrow B$	$B \rightarrow A$
Inconsistencies:	$C - INC1 = BD_A \neq BD_B$	$C - INC2 = BD_B \neq BD_A$
C-ICR1- Definition	To correct these inconsistencies, the B "Begin Date" should receive the value of the B "Begin Date".	
Formalization	$C\text{- }ICR1 = BD_B \leftarrow BD_A$	
Second Case:	$A \rightarrow B$	$B \rightarrow A$
Inconsistency:	$C\text{-}INC3 = ED_A \neq ED_B$	$C\text{-}INC4 = ED_A \neq ED_B$
Translate C-ICR2	To correct these inconsistencies, the B "End Date" must receive the value of B "End Date".	
Formalization	$C\text{- }ICR2 = BD_A \leftarrow BD_B$	

Table 8: Definition and formalization of the Aggregation ICR.

First case:	$A \rightarrow B$	$B \rightarrow A$
Inconsistency:	$A\text{-}INC1 = BD_A < \max (BD_{(B1, B2\ldots Bn)})$	$A\text{-}INC2 = \max(BD_{(B1, B2\ldots Bn)}) > BD_A$
A-ICR1 - Definition	To correct these inconsistencies, the A "Begin Date" should receive the value of the largest "Begin Date" amongthe B elements	
Formalization	$The\text{-}RIC1 = BDA \leftarrow \max (BD_{(B1, B2 \ldots Bn)})$	
Second case:	$A \rightarrow B$	$B \rightarrow A$
Inconsistency:	$A\text{-}INC3 = ED_A > \min (ED_{(B1, B2\ldots Bn)})$	$A\text{-}INC4 = \min (ED_{(B1, B2\ldots Bn)}) < ED_A$
A-ICR2 - Definition	To correct these inconsistencies, the A "End Date" must receive the lowest "End Date" amongthe B elements	
Formalization	$A\text{-}ICR2 = ED_A \leftarrow \min (ED_{(B1, B2\ldots Bn)})$	

Table 9: Definition and formalization of Synchronic Relationship IRC.

First case	$A \rightarrow B$	$B \rightarrow A$
Inconsistency:	$Sr\text{-}INC1 = BD_A > BD_B$	$Sr\text{-}INC2 = BD_B < BD_A$
SR-ICR1 - Definition	To correct these inconsistencies, the B "Begin Date" must receive the A "Begin Date".	
Formalization	$SR\text{- }ICR\ 2 = BD_B \leftarrow BD_A$	
Second case:	$A \rightarrow B$	$B \rightarrow A$
Inconsistency:	$SR\text{-}INC3 = ED_A < ED_B$	$SR\text{-}INC4 = ED_B > ED_A$
SR- Definition	To correct these inconsistencies, the B "End Date" must receive the A "End Date"	
Formalization	$SR\text{- }ICR\ 2 = ED_B \leftarrow ED_A$	

representation to designate the right relations and $B \rightarrow A$ to designate inverse relations. Tables 2, 3, 4, 5 and 6 present the definition and formalization of the IVR.

3.4 Inconsistency Correction Rules

If there is an inconsistency there must be a way to correct it. Therefore, we propose next a set of Inconsistency Correction Rules.

In this paper we consider inconsistency when the relationship among two elements does not comply with the specifications outlined in IVR. Otherwise we can say that there is inconsistency when applying the IVR results in false. The ICR consist of temporal adjustment procedures among the elements involved in an inconsistent relationship. According to the establishment of formal expression, each rule can result in one or two inconsistencies (INC) and correction rules respectively. The 7 to 11 tables present the definition and formalization of ICR.

The temporal adjustment is always done in function of dependency exists among related elements is defined by the relationship. In general, the dependent element will always receive the value (temporal attribute value that is inconsistent) of the element from which it depends.

3.5 Mapping among the Relationships and the Rules

Thus, it is possible to make a match among the types of relationships, IVR, INC and ICR as well as the types of relationships defined in ArchiMate metamodel, as shown in table 12. For each relationship are presented their respective rights relations (top) and the inverse relations (down).

3.6 Temporal Analysis of EA Elements based on ArchiMate Metamodel

Considering the characteristics of the EA elements, it is possible to regroup them into two categories. (1) ***Temporal Elements*:** are those which it is possible to determine through temporal attributes when they come into a certain State. (2) ***Timeless Elements*:** we consider the element that due to the specifications it is difficult to determine when they will enter to a specific State. Due to their nature, temporal rules are used only for the temporal elements.

4 EVALUATION

To test the rules, we develop a prototype software: EAMS-RulesTime. The development process of this

Table 10: Definition and formalization of Create ICR.

	A → B	B → A
Inconsistency:	Cr-INC1 = $ED_A < BD_B$	Cr-INC2 = $BD_B > ED_A$
Cr-ICR	To correct these inconsistencies, the A "End Date" of the should receive the value of B "Begin Date".	
Formalization	Cr-ICR= $ED_A \leftarrow BD_B$	

Table 11: Formalization of Discontinuation Inconsistency.

	A → B	B → A
Inconsistency:	D-INC1 = $EDB_A > ED_B$	D-INC2 = $EB_B < EDB_A$
Correction rule	Not applied. Only a warning should be issued.	

Table 12: Mapping among the rules, relationship, IVR, INC and ICR.

Rules	Relationships	IVR	INC	ICR
Composition Rule	Composed of	C-IVR1	C-INC1, C-INC2	C- ICR1
	Composes	C- IVR2	C-INC3, C-INC4	C- ICR2
Aggregation Rule	Aggregated by	A- IVR1	A-INC1, A-INC2	A- ICR1
	aggregates	A- IVR2	A-INC3, A-INC4	A- ICR2
Create Rule	creates	Cr- IVR1	Cr-INC1	Cr- ICR
	Created by	Cr- IVR2	Cr-INC2	Cr- ICR
Synchronic Relationship rule for	Used by, Assigned from, Realizes, Read by, Deletes, Updates, Influences, Specializes, Accessed by, Flow to, Associated to, triggers, Owns,	SR-IVR1	SR-INC1, SR-INC1	SR- ICR1
	use, assigned to, realized by, reads, deleted by, updated by, influenced by, specialized by, accesses, flow from, associated with, triggered by, Owned by	SR-IVR2	SR-INC3, SR-INC4	SR-ICR2
Discontinuation Rule	All, except (create)	D- IVR1	D-INC1, D-INC2	-

Table 13: Classification of elements of and when the time.

Temporal Elements	Timeless Elements
Product, Representation, Contract, Business Object, Business Service, Business Process, Application Service, Infrastructure Service, Application Component, Node, Communication Path, System Software, Location, Goal, Representation, Work Package	Value, Date Object, Artifact, Business Function, Business Event, Function, Function, Business Infrastructure Player, Business Role, Business Collaboration, Business Interface, Device, Network, Assessment, Deliverable, Driver, Principle, Requirement, Stakeholder, Meaning, application Collaboration, Business Interaction, application Interaction, application interface, infrastructure, interface Constraint, Gap, Plateau

Table 14: Result of the evaluation of the solution (C = Composition, A= Aggregation, SR= Synchronic Relationship, Cr=Create).

Rule type	C-Rule	A-Rule	SR-Rule	Cr-Rule	Total
Nº of inconsistencies founded	18	20	48	40	126
Nº of inconsistencies corrected	12	15	31	36	92

prototype complied with all requirement defined in the mapping presented in table 12 and also the classification of the elements shown in table 13, to check the operation of EAMS-RulesTime we did two experiences.

First we created an xml file with some projects where we simulate some business architecture scenarios based on the ArchiMate metamodel and checked the inconsistencies and later corrected them through EAMS-RulesTime.

To test the rules in a real environment, we use an xml file containing several projects from various organizations in Portugal including AMA (Agência para a Modernização Administrativa), extracted from the EAMS tool developed and marketed by Link Consulting. The file had a total of 3742 elements of which 1434 timeless elements and 2308 temporal elements. The result of this test is presented in table 14.

The result obtained in the tests reveal the importance of using this tool (heance the rules) within organizations. The set of modifications allowed the model EA were updated which improves the quality of decision-making on the enterprise transformation.

5 CONCLUSIONS

In this research we propose a set of rules to validate the temporal relationship of the EA elements. These rules are divided into two categories: The Inconsistencies Verification Rules and Inconsistencies Correction Rules.

For specification of the rules we consider the following aspects: the type of dependency that each relationship represents, the States of elements and element types, fulfilling this way the goal: set logging rules specifications on the basis of the ratio among elements.

We proceeded with the classification and subsequently and the validation criteria formalized through mathematical logic expressions.

The logical expressions proposed are a simple way to represent the verification rules and inconsistencies correction.

The rules, even though they are based on operation of the EAMS, they were specified in a generic way; therefore, it is possible that other similar systems to EAMS can use them in the implementation of features based in the application of the rules.

We verify that the impact of using the rules through the development of a prototype called EAMS-RulesTime. In this prototype we applied to all proposed specifications in section 5. Despite the limitations, the results of the tests are satisfactory which allows us to conclude that the proposed rules are functional so we recommend the implementation of the same management tools and in order to minimize the amount of problems of existing inconsistencies.

The contributions of this work open doors to define new rules for the validation of models and can be applied on EA management tools used by companies.

5.1 Limitations

During the research process some work-related limitations were observed.

The first limitation is that the proposed rules be specified taking into account only three of the possible States that can be assigned to an element of and.

The proposed solution is focused only on temporal elements and which limits the validation of models to this area. But we believe that other types of inconsistencies such as a passive element be responsible for creating an element of active structure.

The last limitation is the fact that the rules were developed according to the specifications of the ArchimMate metamodel, thus limiting its use to other EA metamodels.

5.2 Future Work

We think that, despite the satisfactory results, the proposed solution opens important perspectives for the future work, derived from the limitations presented in section 5.1.

We think that it is crucial to develop other temporal rules in order to cover other States and temporal elements that have not been treated at work, since the elements of and may be in States such as pregnancy, test, or removed in a way that may affect the operation or even the transformation of EA.

Given other types of existing inconsistencies and models, we found important to create new types of rules (which are temporary only) in order to resolve these inconsistencies.

Improve the rules in order to make possible its use by other types of EA metamodels that not only the ArchiMate metamodel.

ACKNOWLEDGEMENTS

This work has been partially supported by the Fundação da Ciência e Tecnologia through funding of INESC-ID, ref. UID/CEC/50021/2013.

REFERENCES

Abraham, R., & Aier, S. (2012). Extending Enterprise Architecture Management into an Enterprise-Wide Coordination Service. *74. Wissenschaftliche Jahrestagung Des Verbandes Der Hochschullehrer Für Betriebswirtschaft E. V.*, (1). Retrieved from http://www.alexandria.unisg.ch/export/DL/219228.pdf.

Ahlemann, F., Steiner, E., & Messerschmidt, M. (2012). *Strategic Enterprise Architecture Management*. http://doi.org/10.1007/978-3-642-24223-6.

Aier, S., & Gleichauf, B. (2010). Application of Enterprise Models for Engineering Enterprise Transformation. *Enterprise Modelling And Information Systems Architectures*, *1*(5), 58–75.

Braun, C., & Winter, R. (2005). A comprehensive enterprise architecture metamodel and its implementation using a metamodeling platform. *Desel, Jörg; Frank, Ulrich*, 1215. http://doi.org/10.1145/1244002.1244267.

Buckl, S., Matthes, F., Monahov, I., & Schweda, C. (2011). Modeling Enterprise Architecture Transformations. *Weblogs.in.Tum.De*. Retrieved from https://weblogs.in.tum.de/file/attachments/Tmp/1f9x8r3yxcbm8/Bu11c.pdf.

Harmsen, F., Proper, H. A. E., & Kok, N. (2009). Informed governance of enterprise transformations. In *Lecture Notes in Business Information Processing* (Vol. 28 LNBIP, pp. 155–180). http://doi.org/10.1007/978-3-642-01859-6_9.

Iacob, M.-E., Jonkers, H., Lankhorst, M. M., Proper, H. A., & Quartel, D. A. C. (2012). *ArchiMate 2.0 Specification*. *ArchiMate 2.0 Specification*.

Lankhorst, M. (2004). ArchiMate Language Primer. *Work*. Retrieved from http://www.uio.no/studier/emner/matnat/ifi/INF5120/v07/undervisningsmateriale/Archimate.pdf.

Tribolet, J., Sousa, P., & Caetano, A. (2014). The Role of Enterprise Governance and Cartography in Enterprise Engineering. *Emisa*, *0*(0), 1–13.

Ylimäki., T., Niemi., E., & Hämäläinen., N. (2007). Enterprise architecture compliance: The viewpoint of evaluation. *ECIME 2007: European Conference on Information Management and Evaluation*, (Ecime), 409–418. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-84896534690&partnerID=40&md5=4b8dd9934fc902d4a1c7f2a2428e81ec.