Modeling Enterprise Architecture Transformations

Sabine Buckl, Florian Matthes, Ivan Monahov, and Christian M. Schweda

Chair for Software Engineering for Business Information Systems
Technische Universität München, 85748 Garching, Germany
{sabine.buckl,matthes,ivan.monahov,schweda}@in.tum.de

Abstract. A rapidly changing business environment, regulatory development, and technological advancements constantly impose changes on the modern enterprise. Thus, the ability to perform EA transformations in a timely manner has attracted growing attention recently within the field of EA planning. Based on the different EA planning perspectives, numerous models and modeling techniques have been developed and examined over the last few years. Yet, the question "How do these perspectives of EA planning interrelate?" remains unanswered. This paper focuses on the ways in which different perspectives of EA transformation – activities, replacements, and lifecycles – can be modeled as add-ons for EA models and the ways they interrelate.

Keywords: Enterprise architecture, EA transformation, EA transformation modeling

1 Motivation

Challenging economic, regulatory, and technical environments force modern enterprises to continuously change and adapt themselves in order to reshape the foundation of their business execution (cf. Ross et al. in [16]). This need for constant change especially pertains to what is called the *enterprise architecture* (EA), i.e. pertains to the "fundamental organization of the enterprise embodied in its components, their relationships to each other and to the environment as well as the principles guiding its design and evolution". This definition of the EA, based on the ISO Std. 42010 [13], gives an indication on the complexity of the subject of change. According to Wagter et al. [19] this subject has to be made *dynamic* in order to react to environmental changes or to proactively realize potential for optimization.

Above requirements lead to the discipline of EA management in general and the activity of *EA planning*, in particular. EA planning provides an embracing and holistic perspective on organizational change and facilitates enterprise transformation, thereby seeking to maintain and foster the alignment between the different parts of the enterprise. Especially, ensuring and strengthening the mutual alignment of business and IT is, according to Luftman et al. [15], a key objective of enterprise transformations and hence for EA planning.

Reflecting the ongoing interest in the field of EA management, the area of EA planning has received particular attention in the last years. Different methods and modeling techniques aim to cover the intricate topic of guiding and describing EA change in a way that is applicable to enterprise architects in transforming organizations. The works of Buckl et al. presented in [5, 4] or of Aier and Gleichauf, see [1, 2], make excellent examples of recent contributions to the field. These and other approaches, as we show in Section 2, provide punctiform solutions to distinct problems in EA planning. In particular, each approach puts emphasis on a dedicated relevant aspect of transformation planning, with each aspect reflecting a specific perspective on changing EAs. These perspectives complementary lead to different ways of modeling EA transformations. This motivates the first research question of this article, which reads as follows:

> Which different perspectives on EA planning exist in literature and how do they interrelate?

Foreclosing the answer to the first question, we state that four perspectives exist, of which one – namely the one of *temporality* – is fundamental to all other perspectives. These three perspectives in turn do not have much in common and are described in approaches largely unrelated to each other. With the missing linkage between the approaches, a user willing to establish EA planning in a practice-relevant way on a rigorous basis, is faced with a situation that he or she has to decide which of the aspects is most important, being able only to select one very aspect for implementation. Rephrasing these considerations to the impact of EA transformation planning on EA modeling, which this article is dedicated to, this means the following: an organization that wants to plan its EA's transformation has to select one of four different ways of modeling transformation. Having selected not only the basic solution of temporality, there is indeed no predefined way to add other perspectives to the selected one. This situation yields the second research questions of this article, which can be described as follows:

> How can different perspectives on EA transformation be modeled as add-ons for EA models? How do the add-ons interrelate?

With the first research question being answered at the end of Section 2, the subsequent Section 3 is dedicated to a concise representation of EA transformation perspectives. Preparing this, we discuss some liabilities of object-oriented modeling and present a slightly enhanced form in order to capture the 'true' ontological nature of the different perspectives. As our presented solution is well-grounded in existing literature on the field, we abstain from giving a virtual case for evaluating the findings. Instead, we head for a critical reflection in Section 4 summarizing questions that arise from the article's findings and give an outlook on how to incorporate our solution into a comprehensive method for EA transformation planning.

2 Literature review

Schönherr showed in [17] that a plurality of literature on EA management currently exists, thus mirroring the continuing interest in the subject. This situation nevertheless particularly aggravates a comprehensive coverage of the topic. Against that background, we opt to build the subsequent review of literature on the analysis of Buckl and Schweda conducted in [8]. This general analysis gives an overview on the prescriptions made be individual EA management-related approaches. Based on the analysis, we select the The Open Group Architecture Framework (TOGAF) [18], the approach presented by Buckl et al. in [5, 4], and the one of Aier and Gleichauf [1, 2], as these approaches are mentioned to provide good coverage of EA planning related aspects. The review is complemented by the approach of Brückmann et al. which present a practitioners perspective on modeling lifecycles of EA elements. During the review of these approaches, we specifically focus on modeling techniques provided and in particular on the underlying *information models*, if present.

The most basic way of modeling temporal aspects of EA elements is to assign a period of validity to each individual element. TOGAF [18] applies this way of modeling onto different concepts in its "content metamodel", as the information model is called in the framework. Key architectural concepts as "process", "service", "information system", and "application component" are equipped with an attribute "retire date". According to the metamodel's documentation, this attribute is used to delineate the end of life for an element of that type. For the "physical application component" in particular TOGAF does not only provide a retirement date, but complementary introduces an "initial live date", used to denote the day of that very component going into production. Regarding the utilization of the corresponding attributes, TOGAF does not make specific prescriptions, abstaining from giving consistency constraints. Lifecycle modeling is also alluded to in the content metamodel, but receives limited attention with regard to the fact that only the current lifecycle status for physical application components may be modeled, ranging from "proposed" over "live" to "retired". With this kind of lifecycle modeling, we can again exemplify the omission of consistency constraints, such that a component may be in state "proposed" even at point in time, where its "retire date" has long passed.

In [5, 4] Buckl et al. take a different perspective on EA planning, emphasizing on the role of projects as drivers of enterprise transformation. Committing to the principle that any change on enterprise-level is the consequence of execution of a project or a similar activity. Exemplifying this with the subject of business support provided by the enterprise's business applications, Buckl et al. devise a way of modeling that introduces an intermediary concept "business support" that links business applications with the supported business processes at the using organizational units. This intermediary concept is therein regarded as reification of the ternary relationship between the three concepts and is equipped with attributes to denote the relationship's period of validity. The period of validity is nevertheless not defined for each particular business support element, but is derived from the activities (named "work packages" there) that may be linked to elements committing to this concept. Being more precisely, Buckl et al. define three different types of relationships between work package and business support, namely "introduces", "changes", and "retires". The first and last relationship in particular supply the periods of validity for the associated element, such that a business support is valid once the introducing work package has finished and the retiring work package has not yet been begun. Different work packages are in turn composed to the projects changing the make-up of the overall EA. With the way of modeling introduced in [5, 4], it is not only possible to relate the business support to the drivers of its change. Buckl et al. discuss this fact, introducing a specific stereotype `<<projectDependency>>` that may be used to indicate that a specific concept in the EA has a period of validity derived from the corresponding projects. This thinking on the one hand lays the groundwork for the mechanism to be discussed in Section 3, but remains limited when it comes to the interplay between different project-dependent concepts. Specifically, Buckl et al. do not discuss consistency constraints between such concepts and their linking relationships.

Aier and Gleichauf discuss in [1, 2] how the transformation nature of EA planning may be reflected in the corresponding management methods. Central aspect of their discussions is a so-called "transformation model" that describes which EA element from a previous state is replaced by which element in a subsequent state of the EA. As part of the considerations, Aier and Gleichauf introduce different types of EA states, namely the "as-is" and the "to-be" states that are considered during transformation planning. A transformation model in terms of [2] describes the replacement relationships linking models of two different EA states. This may in particular be an as-is and a to-be state, or two to-be states targeting different points of time in the future. Complementary, Aier and Gleichauf further introduce the notion of the "will-be" state, reflecting that the actual transformation may proceed different from the planned one. Such will-be state is used as basis for subsequent planning steps instead of the to-be state actually intended to hold for the specific point in time. Building on the transformation models, Aier and Gleichauf describe a set of analysis techniques that may be used to understand the EA transformation plan in more detail. One analysis technique leverages the 'classical' critical path analyses on an EA level, seeking to find which series of transformations is critical for the evolution of the overall EA. The analysis techniques nevertheless do not account for consistency constraints in the transformation model or the models of the different states. With no dedicated information model given, it is further not easy to judge which concepts in an EA may be considered as subjects of transformation. The general terms in which the topic of EA planning is discussed in [1, 2] nevertheless support the assumption that the transformation of EA elements committing to arbitrary EA concepts may be modeled.

Brückmann, Schöne, Junginger and Boudinova present in [3] a lifecycle model of EA elements, and define the following five lifecycle states: "proposed", "test", "productive", "standard", and "retired". The "proposed" state suggests that an existing EA element either does not fully meet the requirements or that a new EA element promises advantages to existing ones. Following the proposals' approval, the EA element enters the "test" state. Depending on the test results, the EA element can change its state to either "productive" or "retired". Once an approved EA element is in use, it automatically enters the "productive" state. If an EA "productive" element fits into the long-term IT strategy, it obtains the status "standard"; otherwise-"retired". The benefits of a particular EA element in the "standard" state and its alignment with the IT strategy are proven in production by the EA management. "Retired" EA elements are not to be used in future scenarios. New versions of the "retired" EA elements should be created and proposed instead.

3 Modeling EA transformation

In the existing literature of the field as revisited above, we can discover four different principles for approaching the modeling of EA transformations. The most simplistic, but fundamental approach assigns a period of validity to any object that participates in a transformation. Speaking more precisely, any EA concept, whose transformation should be modeled, is reflected in a model type that supplies properties for specifying an associated instance's period of validity. By making distinct instances of that type valid in different periods of time, a change in the EA may be described.

Aforementioned principle is nevertheless only the most basic way to represent EA transformations. In particular, the three following aspects of EA transformation are not accounted for by this modeling principle:

EA elements do not change accidentally An element of the EA is change by some sort of activity, e.g. a project, and does not change on its own. For managing the EA transformation it may be necessary to know which activity is changing which EA element.

EA elements may replace each other An element may supersede a previously existing one, taking over some or all of this element's responsibilities. On the contrary, an element may also be retired without having a replacement or may be newly introduced.

EA elements have a lifecycle Quite many EA elements undergo a change of lifecycle phases as well, e.g. being "proposed" at a certain point in time, whereas being "retired" at the end of their life.

Any of above aspects adds a specific perspective on EA transformation that refines the basic principle of the period of validity, with a driving activity (**A**), a replacement relationship (**R**), and lifecycle information (**L**), respectively. As the literature review in Section 2 showed, each of the aspects may occur independently, but especially the work of Buckl et al. in [6] provided an indication that different perspectives may be applied simultaneously, if needed. Exploring this idea in more detail, we subsequently elaborate on the interplay of the different principles and show how this is reflected in corresponding EA information models. Figure 1 provides a conceptual framework for these elaborations introducing the different types of modeling that we would expect.

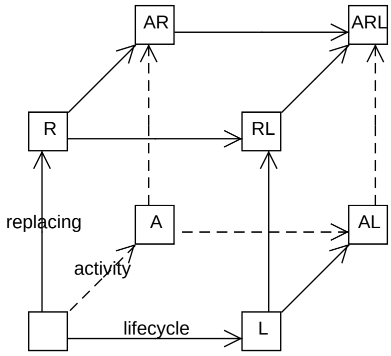

Fig. 1. Aspects of EA transformation modeling

Before nevertheless starting with the elaborations on the modeling techniques specific for the perspectives, we have to add a side-note on two ontological principles in conceptual modeling that are employed in the subsequent considerations. For doing so, we follow the exposition of the principle as provided by Guarino and Welti in [10], further calling on the terminology of Guizzardi in [12].

The first principle is the one of *identity*. A concept can carry the identity for the associated elements or may *dispersive*, i.e. not relevant with respect to the elements' identities. Put in a slightly different way, we may say that an identity-carrying concept supplies a sort of equivalence relation which the model users employ to differentiate between elements covered by the concept. An example for this would be the concept "website" which carries the identity for corresponding elements in turns of the property of the "url" or the concept "book" identifying the elements with the property "isbn". A dispersive concept does not provide that kind of equivalence relation to identify corresponding elements and can hence be thought of as some kind of categorizing concept. Continuing the above example, possible categorizations may be "English", "Swedish" or "German". These categorizations may apply both to books and to websites, thus not being sub-concepts thereof. These concepts on the contrary would name "English book" or "Swedish website". Being it that both books and websites may be categorized by concepts as "English"[1] shows a particular difference between dispersive and non-dispersive concepts. As Guizzardi delineates in [12] this distinction with respect to the ontological nature of a concept is not well reflected in today's conceptual modeling languages. In these languages each concept, either dispersive or not, is mapped to a type or class, neglecting a potentially important difference. Seeking to explore the true ontological nature of EA transformation modeling, we do not prematurely commit ourselves to the simplified point of view taken in most modeling languages. On the contrary, we employ a technique

[1] More precisely the categorizing concept should be called "English content".

inspired by Guizzardi in [12] denoting dispersive types with a stereotype in order to distinguish them from their non-dispersive counterparts.

The second principle of interest is the principle of *rigidity*. Put in short, it asks whether an element commits over its entire lifetime to the same concept or, if different concepts cover different phases of the element's life. While elements of rigid concepts stay committed to the concept of their entire existence, non-rigid concepts are applicable only for a limited period of time. Quickly calling on an example we think of the concept "human" which applies to the corresponding elements over their entire lifetime, thus being a rigid concept. In contrast the concept "teenager" is non-rigid, as a teen has previously been a "kid" and (hopefully) matures to an "adult". Thinking again of how concepts are represented as types and classes in today's prevalent languages for conceptual modeling, one comes to a similar finding, as Guizzardi does in [12], being that non-rigid typing is not well-supported in these languages. Repeating the statement on dispersive types, we call on a technique inspired by Guizzardi and will use stereotypes in the subsequent models to denote that a type actually is non-rigid. Equipped with this ontological groundwork, we are ready to proceed to modeling the first perspective as follows.

3.1 Activity modeling

Refraining the perspective taken by Buckl et al. in [5, 4], the first perspective introduces the notion of the "transformation activity". Such activities may be identified with transformation projects or, more precisely, a part thereof, e.g. a work-package. In particular, an activity may introduce or retire an element of the EA, as represented in an instance of an EA type. Conversely, any such element may be introduced or retired by at maximum one of such activities. Reflecting the ontological nature of this kind of activity modeling, the model fragment presented in Figure 2 introduces the dispersive type of the ActivityAffectable. An organization may decide to subtype this type to indicate that a specific EA type is subject to transformation.

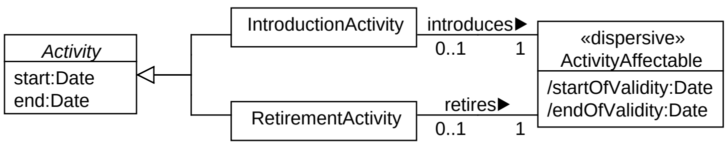

Fig. 2. Activity modeling

In order to ensure conceptual consistency of models committing to the activity modeling fragment, the following constraints apply:

```
ctx Activity: start ≤ end

ctx ActivityAffectable:
  startOfValidity = (introduces == null)?null:introduces.end
  endOfValidity   = (retires == null)?null:retires.start
```

The former constraint ensures that any activity ends after it has started, where the latter activity is used to derive the period of validity of an EA element from the corresponding introducing and retiring activities. If none such activities exist, then no actual limits for the period of validity of the corresponding element are assumed.

3.2 Replacement modeling

Aier and Gleichauf introduce in [1, 2] the concept of the transformation model linking models of different EA states. The transformation model describes which EA element from an older state is replaced by which EA element from a newer state. While Aier and Gleichauf do not provide an explicit information model, they give a graphical indication on the true nature of a transformation model. This indication mirrors the conception as shown in Figure 3, where a general dispersive type Changing is introduced. Each EA element instantiating a subtype of Changing participates in an acyclic predecessor-successor relationship. This relationship denotes which preceding EA elements are superseeded by the EA element in consideration and conversely denotes the EA elements that replace the particular element.

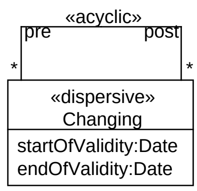

Fig. 3. Replacement modeling

The predecessor-succesor relationship complements the mechanisms of the period of validity in the sense that a superseeding EA element must have a 'later' period of validity than its superseeded pendant. The following constraints operationalize the aforementioned fact:

```
ctx Changing:
∀ p in pre: p.endOfValidity ≤ endOfValidity
∀ p in pre: p.startOfValidity ≤ startOfValidity
```

```
∀ p in post: p.endOfValidity ≥ endOfValidity
∀ p in post: p.startOfValidity ≥ startOfValidity
```

3.3 Lifecycle modeling

The topic of lifecycle modeling is already sketched in TOGAF [18] as discussed in Section 2. A lifecycle thereby denotes that a single element of the EA evolves over different phases, of which most notably "development", "production", and "retirement" are used. Also the EA management pattern catalog (see [9]) supports lifecycle modeling for different EA concepts, such as business applications or the services provided thereby. The corresponding information model nevertheless is simplistic and introduces dedicated properties for delimiting the periods of validity for the individual phases. With the ontological primitive of non-rigid typing in mind, we can express the true nature of lifecycled EA elements in a more concise manner. The information model fragment shown in Figure 4 provides such ontologically well-founded method of describing element lifecycle, by subtyping the abstract base type into distinct phased, i.e. non-rigid, subtypes. An actually lifecycled concept may in turn apply the fragment by adding the fragment's base type to the type representing the concept, thus inheriting the different phases.

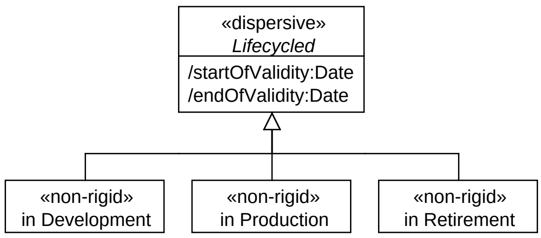

Fig. 4. Lifecycle modeling

When it comes to the question of corresponding constraints, the rich semantics of Guizzardi's phased types (cf. [12]) allows omitting any additional constraints. In particular, Guizzardi does not only consider phases as non-rigid types, but also demands them to be disjoint with respect to a given state of affairs. Thereby, he demands that exactly one of the phases applies at a particular point in time.

3.4 Modeling activities, replacement, and lifecycle

Superimposing the three types of EA transformation modeling as described above, we result in a modeling fragment as shown in Figure 5. In this fragment the simplistic replacement relationship is mediated via the relating concept of an activity. An activity in turn does not introduce or retire an EA element completely, but refrains the notion of the non-rigid type. This means that an activity describes that an EA element is put into production or into retirement, whereas the identity of that particular element remains unchanged.

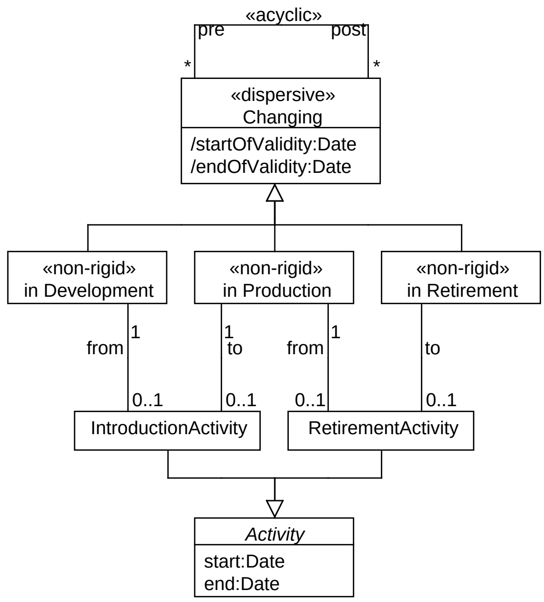

Fig. 5. Modeling activities, replacement, and lifecycle

With the embracing way of modeling activities, replacement, and EA element lifecycle described above, also different constraints apply. With the help of these constraints, we ensure that an EA information model using the corresponding modeling fragment is bound to describe ontologically consistent states of the universe of discourse. In particular, constraints on the periods of validity apply such that any phase is only valid for a period of time bounded by the corresponding starting or ending activities. An additional constraint applies with respect to the predecessor-successor relationship in a way that a superseded version must have at least one lifecycle phase starting earlier than the superseding version. A converse constraint holds for the end of one lifecycle phase. In particular former rule calls for the phases "development" and "retirement", respectively.

Revisiting the different ways for modeling EA transformation, we conclude with a critical perspective on the role of constraints. In above model fragments only the constraints that apply within the specific fragment, are alluded to. Contrariwise also constraints in the overall EA model arise from the need to model EA transformation, especially, when different types in the EA are subject to transformation. Exemplifying this, we may investigate an EA model, where transformation planning should be applied to both business applications and infrastructure components. It is sensible to assume that in such model the business applications are further linked to the infrastructure components on which they rely. With the two related types being subject to transformation, an additional constraint has to be applied to ensure temporal consistency of the model: each business application can only rely on infrastructure components with an appropriate period of validity. From a more general perspective, each relationship between two transformable types may be subject to an additional constraint demanding that the related instances are valid at the same time. Nevertheless, such constraints does not necessary hold for any kind of relationship, as exemplified above with the "replacement" relationship. This yields a critical distinction between two kinds of relationship with respect to their temporal qualities (cf. discussions of Guarino and Welti in [11]):

synchronic relationships that are only valid relating EA elements whose periods of validity intersect. In particular, relationships of that kind only hold for the intersecting period.

diachronic relationships that may validly relate EA elements with non- intersecting periods of validity and hold independently of the elements' periods of validity.

Within typical EA information models as the one presented by TOGAF [18], Lankhorst et al. [14] or in the EAM pattern catalog [9], the majority of relationships is synchronic of nature. Against this background, it is sensible to assume that any relationship is synchronic by default, whereas an additional stereotype `<<diachronic>>` may be used to denote this nature. We further shall assume that all instances of an EA type that do not participate in transformation modeling have an unlimited period of validity such that the constraint of synchronicity may natively be fulfilled.

Another interesting topic applies in the former information models, especially when they shall be used as "model fragments" for building an organization-specific and transformation-aware information model. The dispersive types and their subtypes used throughout the information models then gain an additional semantics, in which they act as "type templates". This in particular becomes obvious with the example of the different lifecycle phases. The three phased types themselves represent an abstract understanding of lifecycle, which may generically be applied ('added') to a particular type in the EA information model. In fact the phases actually become semantically meaningful with this application, while they themselves may only be regarded placeholders for a real world type. This type is indeed not actually modeled, but results from the augmentation of the EA type under consideration. We nevertheless abstain from delving into the details of templating types, which may be regarded a technical issue of formalistic model composition of fragments. With the article's focus on different perspectives on modeling EA transformations, we call for an intuitive understanding of the presented fragments rather than a formal one.

4 Outlook

This article is devoted to the topic of modeling EA transformation, whose importance reverberates through a multitude of different EA management approaches as shown in Section 1. The state of the art in this field, as revisited in Section 2, presents four different perspectives on transformation modeling, of which one – validity modeling – builds the common basis. In Section 3, we showed how the different perspectives relate to each other and presented re-usable fragments for adding transformation modeling to arbitrary EA information models. Complementing these discussions, we briefly revisited how transformation modeling influences the understanding of relationships in information models, hence devising the notions of synchronicity and diachronicity, respectively.

A key finding of this article is the framework that displays the orthogonality of the different types of EA transformation modeling. This framework may not only be helpful for practitioners to understand the different ways of modeling transformations, which they may select from. The framework also allows positioning and analyzing specific EA information models with respect to their coverage of the topic. Having complemented the different perspectives with fragments that implement the specific understanding of transformation in the particular perspective, the article further provides easy to use building-blocks for EA information model design. The distinction between synchronic and diachronic relationships finally provides a contribution to the ongoing question on the appropriate meta-language for EA information modeling (cf. Buckl et al. in [7]).

Latter point nevertheless deserves more research in the future. Synchronic relationships may not exist for the full period of validity of their corresponding relationship participants, but may have a more limited period of validity on their own. Refraining the example from Section 3, we may assume that over its life-time one business application relies on different infrastructure components without either the components or the application being retired in that time. In this context the relationship itself becomes a subject of EA transformation being valid for a shorter period than its participants. At this point, one may enter a discussion, if its is really the *same* business application or if the application has significantly changed with the infrastructural relationship being revised. Whereas from a philosophic point of view, there may be good arguments for the latter, i.e. for the business application not being 'the same', we see no benefit in such understanding. Instead, we would expect that future research in this field re-assesses the understanding of identity that underlies typical models and information models of the EA. The work of Guarino and Welti [10, 11] seem to us to be a valuable starting point for further investigations. Complementary, Guizzardi's notion of the "relator universal" (cf. [12]) may be beneficially applied to resolve the issue of a relationship's period of validity. Nevertheless, the ontological aspects conveyed in the works of Guarino and Welti as well as of Guizzardi must always be mirrored against the pragmatic quality of the EA models, as those models are to be used by enterprise architects and to be implemented in EA management tools. Future solutions in this area should hence do their best in covering the true ontologic nature of the subject, but should also present re-usable building-blocks for EA information modeling, accessible and comprehensible for their future application.

References

1. Aier, S., Gleichauf, B.: Application of enterprise models for engineering enterprise transformation. Enterprise Modelling and Information System Architectures 5, 56–72 (2010)
2. Aier, S., Gleichauf, B.: Applying design research artifacts for building design research artifacts: A process model for enterprise architecture planning. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST. Lecture Notes in Computer Science, vol. 6105, pp. 333–348. Springer (2010)
3. Brückmann, M., Schöne, K.M., Junginger, S., Boudinova, D.: Evaluating enterprise architecture management initiatives – how to measure and control the degree of standardization of an it landscape. In: Mendling Jan, Rinderle-Ma, S.E.W. (ed.) 3rd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2009). pp. 155–168. Ulm, Germany (2009)
4. Buckl, S., Ernst, A.M., Matthes, F., Schweda, C.M.: An information model for managed application landscape evolution. Journal of Enterprise Architecture (JEA) 5(1), 12–26 (2009)
5. Buckl, S., Ernst, A.M., Matthes, F., Schweda, C.M.: Visual roadmaps for enterprise architecture evolution. In: The 1st International Workshop on Enterprise Architecture Challenges and Responses. Korea (2009)
6. Buckl, S., Matthes, F., Schweda, C.M.: Conceputal models for cross-cutting aspects in enterprise architecture modeling. In: 5th International Workshop on Vocabularies, Ontologies, and Rules for the Enterprise (VORTE 2010) (2010)
7. Buckl, S., Matthes, F., Schweda, C.M.: A meta-language for ea information modeling – state-of-the-art and requirements elicitation. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) Enterprise, Business-Process and Information Systems Modeling. pp. 169–181. Lecture Notes in Business Information Systems, Springer (2010)
8. Buckl, S., Schweda, C.M.: On the state-of-the-art in ea management literature. Tech. rep., Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany (2011)
9. Chair for Informatics 19 (sebis),Technische Universität München: Eam pattern catalog wiki. `http://eampc-wiki.systemcartography.info` (cited 2010-07-01) (2010)
10. Guarino, N., Welty, C.A.: A formal ontology of properties. In: 12th International Conference on Knowledge Acquisition, Modeling and Management. pp. 97–112. Springer, Juan-les-Pins, France (2000)
11. Guarino, N., Welty, C.A.: Identity, unity, and individuality: Towards a formal toolkit for ontological analysis. In: Horn, W. (ed.) Proceedings of the 14th European Conference on Artificial Intelligence. pp. 219–223. IOS Press, Berlin, Germany (2000)

12. Guizzardi, G.: Ontological foundations for structural conceptual models. Ph.D. thesis, CTIT, Centre for Telematics and Information Technology, Enschede, The Netherlands (2005)
13. International Organization for Standardization: ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems (2007)
14. Lau, A., Fischer, T., Buckl, S., Ernst, A.M., Matthes, F., Schweda, C.M.: Ea management patterns for smart networks. In: SE 2009 – Workshopband. pp. 79–90 (2009)
15. Luftman, J.N., Lewis, P.R., Oldach, S.H.: Transforming the enterprise: The alignment of business and information technology strategies. IBM Systems Journal 32(1), 198–221 (1993)
16. Ross, J.W., Weill, P., Robertson, D.C.: Enterprise Architecture as Strategy. Harvard Business School Press, Boston, MA, USA (2006)
17. Schönherr, M.: Towards a common terminology in the discipline of enterprise architecture. In: Aier, S., Johnson, P., Schelp, J. (eds.) Pre-Proceedings of the 3rd Workshop on Trends in Enterprise Architecture Research. pp. 107–123. Sydney, Australia (2008)
18. The Open Group: TOGAF "Enterprise Edition" Version 9. `http://www.togaf.org` (cited 2010-02-25) (2009)
19. Wagter, R., van den Berg, M., Luijpers, J., van Steenbergen, M.: Dynamic Enterprise Architecture: How to Make IT Work. John Wiley (2005)