Socio-technic Dependency and Rationale Models for the Enterprise Architecture Management Function

Sabine Buckl, Florian Matthes, Christian M. Schweda

Technische Universität München (TUM)
Chair for Informatics 19 (sebis)
Boltzmannstr. 3, 85748 Garching bei München, Germany
{sabine.buckl|matthes|christian.m.schweda}@mytum.de

Abstract. Enterprises are complex socio-technical system, whose management can be considered a challenging task. Especially against the background of intricate relationships, dependencies, and contributions that link social actors and technical components, the design of an effective enterprise architecture (EA) management function is not easy to accomplish. While for modeling social dependencies on the one hand and for modeling technical dependencies in the context of an EA on the other hand, well established approaches exist, yet no attempt has been undertaken to bring together the disciplines of modeling. Regarding such an embracing modeling approach as a valuable contribution for designing organization-specific EA management functions, this paper closes the aforementioned gap by combining facilities for intensional and technical dependency modeling. The integrated model is thereby devised along an anonymized practice case.

Key words: enterprise architecture, i* models, EA management governance

1 Introduction

Enterprises present themselves as complex socio-technical systems, in which information technology (IT) plays a crucial role in delivering business services to both internal and external customers. In the light of an ever changing economic as well as regulatory environment, the mutual alignment between business and IT has become an increasingly important goal for modern enterprises. First discussions on this topic date back to the early works of Henderson and Venkatraman in 1993 [14]. Not least the enduring need for guidance on how to align business and IT has recently promoted the development of a new management discipline, namely the discipline of enterprise architecture (EA) management. Central to this discipline is a comprehensive understanding of the enterprise as a system, whose architecture deserves mindful management. Architecture thereby is defines as the "fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, as well as the principles guiding its design and evolution" [16]. The enterprise architecture (EA) covers manifold different facets of the enterprise ranging from business-related aspects to more IT-related ones, see Figure 1. Complementing this static understanding of the enterprise as system of interrelated social and technical constituents, different cross-cutting aspects exert influence or realize the design of the architecture, namely *standards*, *goals*, and *projects*. Especially the latter two cross-cutting aspects play a crucial role in evolving the architecture: goals describe the desired result of the evolution, whereas projects are the implementors‘' of architectural change. Between these two concepts a delicate relationship exists in a way that projects can be analyzed in respect to their goal contribution. This analysis pertains to the underlying *means-end*-relationship connecting projects and goals, respectively. While this relationship may be regarded a cornerstone of governing the evolution of the EA, it has yet not been subjected to in-depth research. This is especially surprising against the background of goals as well as projects being objects of research in EA management and EA modeling for several years now (see the work of Aier et al. [1], Buckl et al. [3, 9], or Frank et al. [12]).

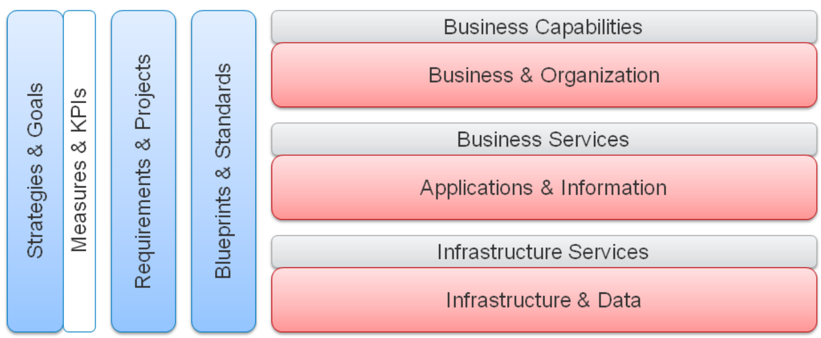

Fig. 1. Conceptual structure of an EA

This paper contributes to the field of EA management, especially in the area of EA evolution governance, by making the relationships between EA-related goals on the one hand and EA-relevant projects on the other hand explicit. By doing so the paper promotes a systemic view on the activity of EA management centered around a conceptual model covering the inner nature of this activity. This model can build on different relevant prefabricates from distinct discipline, among other the *i*-model* of Yu [19], the *goal-question-metric*-approach of Basili et al. [2] as well as related approaches, and the conceptualization of EA change projects presented by Buckl et al. in [3]. Each of these foundational theories and works is shortly introduced in Section 2, preparing the exposition of the interlinking EA governance model in Section 3. Complementing this exposition, the section further delivers some insights into the application of the method at a financial service provider. This application example is anonymized due to confidentiality reasons. Section 4 shows how the presented approach compares to related approaches in the field of EA management and the development of ultra-large scale systems. Final Section 5 critically reflects the contribution of the paper and gives an outlook on future research topics in the field.

2 Foundations and prefabricates

A model for governing the evolution of the EA, i.e. a model targeting the function of EA management, has to bring together three largely different perspectives. Firstly, it has to account for the *intensional* perspective in which *dependers* and *dependees*, i.e. different actors, make explicit the *dependa*, as goals or resources, via which they influence and relate to each other. Secondly, a model for the EA management function must incorporate a more mechanistic and computational perspective on EA-related goals, which are operationalized via *questions* to concrete architectural *metrics*. Thirdly and finally, such EA management model must account for the transformational nature of EA-relevant projects by providing mechanisms to reflect the changes made by a project in the according model of the EA. In the following three Sections 2.1 to 2.3 prominent models and theories covering one or another of the aforementioned aspects are introduced. Together these models form the basis for the comprehensive EA management model introduced in Section 3.

2.1 i* model

The i* model, initially developed by Yu (cf. [19]) and ever since applied and furthered in manifold publications not at least from an own workshop series, is a model that aims at making explicit the intensional relationships between different actors of a complex system. In its current form the i* model is comprised of two submodels, namely the *strategic dependency* and the *strategic rationale* model. The former model describes the dependencies between two actors in terms of *depender*, *dependee*, and *dependum* which can be a (soft) goal, a task, or a resource, respectively. These dependencies are explained in the strategic rationale model, especially detailing on the intentionally desired elements for the corresponding actors. Put in other words, the dependency model provides an abstract black-box perspective on actors and their relationships, whereas the rationale model explains dependencies via a white-box perspective on actors and their intentions. Figure 2 exemplifies a strategic dependency model, also providing an example for the notion of the one-side dependency. In such dependencies, only depender and dependum or dependee and dependum are known, leaving the opposite actor unknown.

A final remark on the different types of dependa in i* strategic dependency models should be added. Two actors can depend via

goals where the depender needs the dependee to get the (measurable) goal fulfilled (as shown in Figure 2),

tasks where the depender relies on the dependee for getting a task executed,

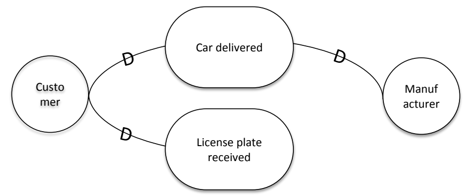

Fig. 2. Modelling dependers, dependa, and dependees with i*

resources where the depender needs the dependee to provide a certain artifact, or

softgoals where the dependee can satisfy a not-measurable goal of the depender.

For the context of EA management especially goal-mediated dependencies may be of interest. Using the mechanisms of the strategic rationale model such goals may further be operationalized via tasks necessary to accomplish these goals and resources necessary to perform the task. We shall revisit these decomposition or means-end-links in Section 3 in more detail.

2.2 Goal-question-metric approach

In [2], Basili et al. present the goal-question-metric (GQM) approach as a way to define measurements in a top-down fashion, focused on goals and based on models. With this way they seek to address a challenge from the field of software development, where many observable characteristics of the design product exist, but a concise interpretation of according metrics is not easily at hand. Central to the approach is a trifecta of steps starting with "goal specification", proceeding over "goal anchoring" to "interpreting measurements". These steps are reflected in the measurement model of the GQM approach, which employs three distinct levels:

Conceptual level defining goals that target specific objects of analysis from perspectives reflecting certain characteristics or models of quality.

Operational level defining a set of questions characterizing the way of assessing the achievement of the according goal. The questions thereby characterize the object of measurement.

Quantitative level defining concrete metrics used to answer the according questions in a quantitative way.

The levels of GQM allow a differentiation between three aspects, which are colloquially subsumed under the term "goal". Thereby, GQM especially facilitates discussions on the abstract nature of a goal and its anchoring in a concrete model of the artifact, on which the goal should be specified. In [9], Buckl et al. make use of this understanding and differentiation to discuss on the role of goals in EA management and modeling. They specifically delineate that goals are *cross-cutting aspects* of EA modeling (cf. Section 1). Further, they show how *dispersive* types, so called *mixins*, may be used to incorporate goals – more precisely their corresponding questions – into structure-oriented models of an EA. Figure 3 shows the basic idea of modeling the relationship between goals, questions (as mixin types), and metrics (as attributes of questions) and gives an application example from the context of EA modeling.

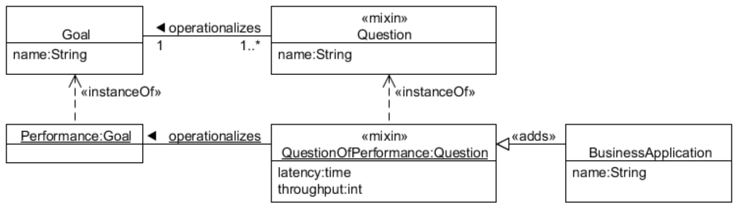

Fig. 3. Modelling goals and questions in the EA context

2.3 Project dependency model

Projects, i.e. the facilitators of organizational change, make up an important cross-cutting aspect of EA models. This has manifold reasons, of which two prominent ones as discussed by Buckl et al. in [4, 3] as well as Ernst and Schneider in [11] should be shortly stated here. Firstly, projects are used during EA planning to describe different possible ways to evolve the EA or parts thereof. Where this understanding of projects focuses on the change perspective associated to a project, secondly they are also modeled to control the EA evolution process itself. In the latter sense, the relationships between projects and EA elements are explored to identify architecture-mediated project dependencies that may aggravate the implementation of specific project portfolios. Both aforementioned scenarios critically rely on an understanding on how projects and elements in the EA are interlinked, although the first scenario presents a more intricate set of relationships. Before delving into the very details thereof, we briefly revisit the term project to highlight a distinction between two types of projects: *run projects* and *change projects*. Projects of both types actually change the enterprise in some way, but only projects of the latter type perform adaptations on a level that is "visible" in the corresponding EA model. An appropriate modeling – using mixins to designate the architecture concepts subject to change – is shown in Figure 4. This model introduces a general relationship "change" to reflect run projects. Specialized relationships, as "introduces" or "retires", may be used designate transformations performed by a change project. These relationships may further be used to supply a *period of validity* to an architecture element in a way that no such element may be regarded valid prior to the completion of its introducing project. Complementing, an architecture element ceases to be valid at that point in time, when its retiring project is completed.

Concluding the exposition of the prefabricates of this paper's approach a minor remark has to be added concerning the relationship between "projects"

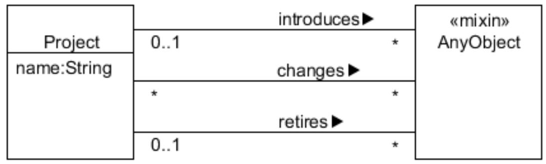

Fig. 4. Modeling run and change projects in the EA context

and "activities" as covered by the i* approach. Whereas Buckl et al. (cf. [3]) do not explicitly account for ongoing maintenance activities in an enterprise, their understanding of "run projects" may also apply on these activities, although statements on "startDate" and "endDate" should not be used in this context. Conversely, every project in the sense of Buckl et al. may be understood as activity in the i* approach. With neither the temporal planning for projects nor the distinction between run and change projects being of central importance for our subsequent considerations, we will identify "projects" and "activities" in the following, hence allowing a uniform treatment of both types.

3 Socio-technic dependency and rationale models for the EA management function

In this section, we devise a model bringing together intensional dependency modeling from the i* method with goal and project dependency modeling from typical EA models. In order to facilitate a good understanding of this model, we explain the steps of model creation along a practice case.

EA management is performed by the bank SBM in order to achieve distinct EA-related goals, as increased disaster tolerance or reduced operating cost. These EA-related goals are raised by different *stakeholders* in SBM, i.e. by different individuals or groups that have certain interests in respect to the system enterprise [16]. As part of EA management or more precisely the setup of EA management, SBM concretizes the abstract goal statements by anchoring them in the architecture elements, which they are to be realized at (cf. Buckl et al. in [8]). This leads to the following concrete goal: "increased disaster tolerance in business process execution", stating that the first abstract statement is anchored in the concept of the business process. The corresponding stakeholder of this goal is a business unit of the organization, which depends on the execution of the corresponding business processes. Understanding this dependency in terms of an i* model, we may identify it with a one-sided dependency, where the dependee is yet not made explicit.

Further detailing "disaster tolerance of business processes" the GQM approach is applied to derive a question suitable for operationalizing the according goal. In the case of SBM the notion of disaster tolerance is reflected in a question of availability on business processes. The goal building on the notion can in this sense be put in more operational terms by stating that availability and disaster tolerance are positively related. Put in other words, this means that an increased availability implies an increased disaster tolerance. With the operationalization via a question on the one hand and an explicit understanding of the relationship quality on the other hand, the dependum of the business unit's one-sided dependency is directly linked to a corresponding concept from the architecture, namely the business process. Using a notation combining notational elements of i* on the one hand and of the UML on the other hand as language to describe EA information models, the afore described situation can be visualized as shown in Figure 5.

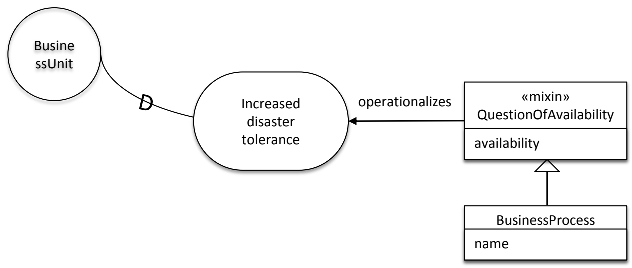

Fig. 5. Partial strategic dependency model enhanced with EA management information

At this point SBM explores the remainder of its EA information model, especially traversing the conceptual relationships starting and ending in the concept of the business process. Domain experts thereby diagnose that the attribute of availability in the business process may be affected by the availability of the supporting business applications. This conversely means that the question of availability also applies to the business application, whereas a feature dependency relationship mechanisms as described by Buckl et al. in [5] may be used to relate business process and business application availability. With this modeling the goal dependency of the business unit is traced down to an operational quality characteristic of an architecture element, namely the business applications supporting the business process under consideration.

Utilizing the mechanisms of EA modeling, SBM is able to describe that maintenance projects executed by the IT department exert influence on the business applications and may hence also influence the relevant property of availability. Identifying a project with a task of the strategic dependency model, SBM can establish a link back to the IT department as dependee, i.e. as responsible actor for this task. Figure 6 illustrates the complete set of dependencies as devised by the EA management team of SBM.

The EA-enhanced strategic dependency model allows SBM to discover the relationship between availability-relevant maintenance activities of the IT department and the goal "increased disaster tolerance in business process execution" on which the business unit critically depends. Having uncovered this relationship, the EA management team of SBM decides to adapt the EA management

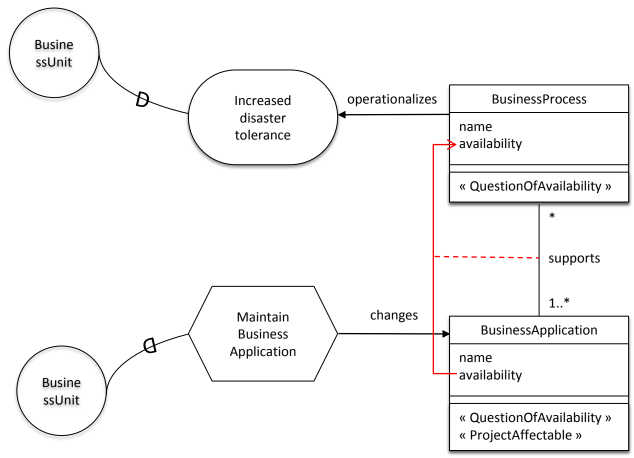

Fig. 6. Enhanced strategic dependency model in the EA management context

function to better pursue the business relevant goal. In particular, an artificial goal "increased business application availability" is introduced and linked to the question of availability as already incorporated in the corresponding architectural concept. Further assuming that the IT department pursues a softgoal of "increased reputation", the EA management team decides to establish a contribution from the availability goal on business applications to the unit's reputation. For doing so the EA management team creates a dashboard on application availability values and makes this dashboard available in the corporate intranet. Thereby, the control loop is closed as a high availability makes a positive contribution to the subjective reputation of the corresponding IT unit.

Above we presented how i* models can be combined with project dependency modeling and the GQM approach in order to comprehensively cover both the intensional and technical dependencies behind an EA management function. What has been modeled in an intuitive manner above, is subsequently grounded in a formal conceptual model based on a revisited understanding of the foundational approaches. Figure 7 presents the conceptual meta-model backing the strategic dependency modeling as utilized by our approach. The terms and conception of this model is rooted in the basics of the i* model as described by Yu in [19]. In this meta-model, we apply the notion of the mixin (cf. Section 2) to form a *dispersive* type "dependum" to reflect the different kinds of concepts that may mediate dependencies between different actors. The thereby supplied modeling facility largely corresponds to the facility provided in Yu's original work.

Further detailing on the nature of the dependencies and their underlying rationales, we present a conceptual meta-model for strategic rational modeling as shown in Figure 8. Central to this model is the linkage between projects and tasks, more precisely a specialization relationship identifying any project with a task. On the contrary, goals are concretized via questions which are

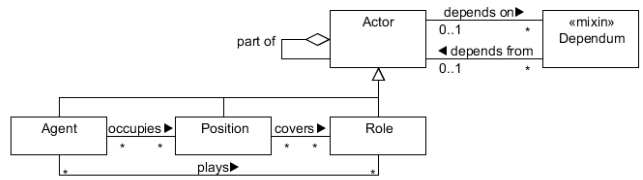

Fig. 7. Conceptual meta-model for strategic dependency modeling

complementary operationalized via metrics. These metrics – attributes of the corresponding type in the sense of an object-oriented model – may finally be linked via feature dependencies describing in which way one property exerts influence on another.

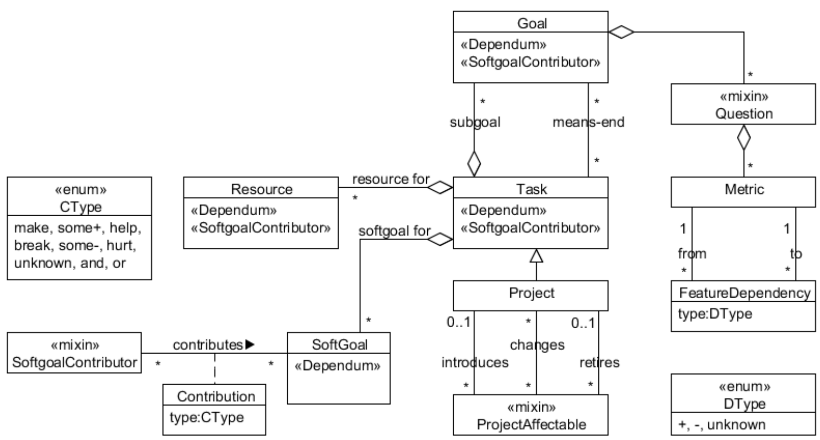

Fig. 8. Conceptual meta-model for strategic rationale modeling

In line with the understanding of Buckl et al., who in [7, 9] discuss different cross-cutting aspects of the EA, the meta-model presented above may be regarded as EA information model building block covering the cross-cutting aspects of projects and goals (cf. Figure 1) in conjunction. By doing so, this building block delineates relationships that back the EA management activity. These relationships can be subjected to further investigations, when the overall performance of the EA management function should be assessed. Further, the dependencies are subject to configuration and adaptation in case the EA management function does not perform as expected. Especially the introduction of additional contribution relationship, i.e. the organizational implementation of feedback mechanisms, may help to respond to unexpected management behaviors.

4 Related work

In [18], Pulkkinen discusses the difficulties of rationalizing architectural decisions in EA management against the background of a layered understanding of the EA. This understanding, while deemed advisable for controlling the complexity of the overall architecture, hampers an integrated perspective on the causal and intensional relationships that interlink the architectural elements and actors. Pulkkinen illustrates this with a typical decision making process, where requirements decided on one layer, e.g. the business architecture layer, are handed over e.g. to the application architecture layer for implementation. This reflects a separation of concerns between the different facets of the enterprise, but may result in a kind of "waterfallian" way of management, that Pulkkinen seeks to counter with a bottom-up information supply. In this sense the hierarchical management system is complemented with cycles that feed relevant architectural information to higher layers. This well aligns with the approach presented in Section 3, although the ideas presented by Pulkkinen in [18] are more general in their claim, while being less detailed in the realization guidance. Put in other words, the model of Pulkkinen does not concentrate on specific bottom-up relationships, but embraces feedback in a highly general manner. Nevertheless, exact statements on how to model and implement such feedback mechanisms in an enterprise are not provided.

The social and managerial dimension of adaptation is also recognized and picked up as a relevant theme in the field of *ultra-large-scale (ULS) systems*. In [17], the author team emphasize on the importance of the "human element" in ULS system. By this they mean not only that quality attributes apply to technical and social components, but also that correlations between human behavior and system quality attributes will become more prominent. Further detailing this conception, the work describes that interactions between people have to be modeled and accounted for in order to understand and control the behavior of the socio-technical ULS systems. Complementing this, the author team of [17] delineates that human and system quality attributes have to be "blended" illustrating this with typical questions that target ULS systems as socio-technical holons. An exemplary question taken from the work asks, what the human part of a system was to do, when the technical part fell below certain thresholds. The model and approach presented in Section 3 of our paper may in this sense be regarded as a contribution to a riposte on the challenges presented in [17]. Moreover, the model allowing to conjointly reason on technical and intensional dependencies linking the actors in an enterprise may provide a valuable ground on which game-theoretic methods prospected in the ULS systems environment can be based on.

Strong emphasis on the social dimension of enterprises is made by the *enterprise ontology* approach presented by Dietz [10]. Central to the ontological approach is an understanding of the enterprise as a system of interlinked commitments. These commitments relating different social actors in the EA are modeled from a *language-action*-perspective, which allows an enterprise ontology to abstract from, what Dietz calls "implementation details". The perspective's underlying Ψ-theory defines four axioms that are used to understand the activities of an enterprise performed as part of their business activities. Against the background of this limitation on business activities not accounting for transformation activities, the approach presented in this paper may be regarded as a continuation of Dietz' Ψ-theory to the level of the management processes concerned with developing and evolving enterprises.

Hoogervost discusses in [15] aspects of governance on the levels of the IT, the corporation, and the enterprise in total. Central to these discussions is a new understanding of governance seeking to bring together what Hoogervost calls the "western and eastern ways of management", respectively. The two ways are distinguished as "mechanistic" and "holistic", of which the former falls short of accounting for the human or social perspective on the enterprise. Put in other words, Hoogervost criticizes prevalent management methods of "western" style to be not open enough for covering the social interdependencies in an appropriate way. As an example for this, the notion of *uncertainty* is presented. Uncertainty is, according to [15], not well accounted for in traditional management and governance models, leading to not-optimal governance and dependency structures. The approach presented by us in Section 3 may be helpful to address the aforementioned issue and close the gap between mechanistic and human-centric management models. More precisely, the relationship between tangible and measurable questions, and informal as well as intensional dependency structures allows analyses covering both relevant management aspects. Furthermore, the both dependency models – namely the one relating different quality characteristics as well as the one expressing dependencies and intensional relationships – may well be augmented to cover uncertainty in respect to the strength of the dependencies.

5 Critical reflection and outlook

The approach and model presented in Section 3 is helpful in linking the social and intensional dependencies in a management function to the structural and qualitative dependencies of the management subject. The simplified practice example allows to get first insights into the applicability of the approach, whereas its grounding in well-established techniques and research results from related fields as described in Section 2 may be regarded as further indication towards the soundness of the approach. Nevertheless, a singular case in addition to a consequent derivation from literature may not completely replace more in-depth investigations covering different applications in distinct practical settings. More evaluation would be needed to show both the utility of the model as well as its practical applicability in different enterprise environments.

Further streams of research originate from the findings presented in this paper. The model for relating stakeholders, goals, and projects in an interlinked web may be helpful for designing and adapting the processes and methods of an EA management function. As such function is always performed to satisfy the goals raised by its stakeholders, an understanding of the dependencies of goal satisfaction can be beneficially in revising the management structures for this field. In this vein, both model and approach provide a contribution to the field of *EA management governance* (cf. Buckl et al. in [6] and Harmsen et al. in [13]), i.e. to the field of establishing, maintaining, and evolving the appropriate structures for managing the EA. This well aligns with another perspective on this subject, namely a systemic perspective on the EA management function itself. In such perspective the social and technical relationships may be aggregated to relatively abstract models, e.g. as *causal loop diagrams*. While it would – against the background of the importance of soft-goals in this model – not be sensible to make quantitative analyses of the causal dependencies, it may nevertheless be possible to derive qualitative statements on the overall behavior of the management system. Especially statements on oscillating management behavior as well as unintended control trends may be discovered qualitatively.

Finally, the model and approach as presented in Section 3 albeit targeting the topic of EA management do not rely on specific assumptions rooted in this management field. In this sense, one may reasonably assume that the model can be applied to other management functions targeting socio-technical management subjects as well. The field of IT service management seems to be a valuable candidate in this respect, especially when incidents and their technical reasons are interlinked. Future research may target the applicability of our model and approach on this or other management functions.

References

1. S. Aier, S. Kurpjuweit, J. Saat, and R. Winter. Enterprise Architecture Design as an Engineering Discipline. *AIS Transactions on Enterprise Systems*, 1:36–43, 2009.
2. V. R. Basili, G. Caldiera, and H. D. Rombach. *The Goal Question Metric Approach*. Wiley, New York, 1994.
3. S. Buckl, A. M. Ernst, F. Matthes, and C. M. Schweda. An information model for managed application landscape evolution. *Journal of Enterprise Architecture (JEA)*, 5(1):12–26, 2009.
4. S. Buckl, A. M. Ernst, F. Matthes, and C. M. Schweda. Visual roadmaps for enterprise architecture evolution. In *The 1st International Workshop on Enterprise Architecture Challenges and Responses*, Korea, 2009.
5. S. Buckl, U. Franke, O. Holschke, F. Matthes, C. M. Schweda, T. Sommestad, and J. Ullberg. A pattern-based approach to quantitative enterprise architecture analysis. In *15th Americas Conference on Information Systems (AMCIS)*, San Francisco, CA, USA, 2009.
6. S. Buckl, F. Matthes, and C. M. Schweda. A viable system perspective on enterprise architecture management. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 11-14 October 2009*, pages 1483–1488. IEEE, 2009.
7. S. Buckl, F. Matthes, and C. M. Schweda. 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.

8. S. Buckl, F. Matthes, and C. M. Schweda. A design theory nexus for situational enterprise architecture management – approach and application example. In *14^{th} IEEE International EDOC Conference (EDOC 2010)*, Vitoria, Brazil, 2010.
9. S. Buckl, F. Matthes, and C. M. Schweda. A technique for annotating ea information models. In J. Barjis, editor, *Enterprise and Organizational Modeling and Simulation: 6^{th} International Workshop, EOMAS 2010 held at CAiSE 2010, Hammamet, Tunisia, June 2010 Selected Papers*, Lecture Notes in Business Information Systems, pages 113–127, Berlin, Heidelberg, Germany, 2010. Springer.
10. J. L. Dietz. *Enterprise Ontology.* Springer, Heidelberg, Germany, 2006.
11. A. M. Ernst and A. W. Schneider. Roadmaps for enterprise architecture evolution. In G. Engels, M. Luckey, A. Pretschner, and R. Reussner, editors, *2^{nd} European Workshop on Patterns for Enterprise Architecture Management (PEAM2010)*, pages 253–266, Paderborn, Germany, 2010.
12. U. Frank, D. Heise, H. Kattenstroth, and H. Schauer. Designing and utilising business indicator systems within enterprise models – outline of a method. In *Modellierung betrieblicher Informationssysteme (MobIS 2008) – Modellierung zwischen SOA und Compliance Management 27.-28. November 2008*, Saarbrücken, Germany, 2008.
13. A. F. Harmsen, E. H. Proper, and N. Kok. Informed governance of enterprise transformations. In H. E. Proper, F. Harmsen, and J. L. Dietz, editors, *Advances in Enterprise Engineering II*, pages 155–180, Berlin, Heidelberg, Germany, 2009. Springer.
14. J. C. Henderson and N. Venkatraman. Strategic alignment: leveraging information technology for transforming organizations. *IBM Systems Journal*, 32(1):472–484, 1993.
15. J. A. P. Hoogervorst. *Enterprise Governance and Enterprise Engineering (The Enterprise Engineering Series).* Springer, Heidelberg, Germany, 1 edition, February 2009.
16. International Organization for Standardization. ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems, 2007.
17. B. Pollak. *Ultra-Large-Scale Systems – The Software Challenge of the Future.* Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA, USA, 2006.
18. M. Pulkkinen. Systemic management of architectural decisions in enterprise architecture planning. four dimensions and three abstraction levels. In *39^{th} Hawaii International Conference on System Sciences (HICSS 2006)*, volume 8, page 179c, 2006.
19. E. S. K. Yu. *Modelling strategic relationships for process reengineering.* PhD thesis, University of Toronto, Toronto, Ont., Canada, 1996.