Collaborative Evolution of Enterprise Architecture Models

Sascha Roth, Matheus Hauder, and Florian Matthes

Technische Universität München
Boltzmannstr. 3
85748 Garching, Germany
{roth,matheus.hauder,matthes}@tum.de

Abstract. Enterprise Architecture (EA) management seeks to align business and IT while realizing cost saving potentials, improving availability and fault tolerance, and increasing flexibility of an organization. Regarding these objectives, decision makers need to be supported with solid and relevant models about the organization's architecture to guide the future development of the EA. In practice, many EA initiatives struggle with inflexible models not meeting the information demand of stakeholders. In this paper, we propose a solution that empowers stakeholders to reveal their information demand collaboratively to facilitate EA models that evolve with changing information demands at runtime. We present core concepts of our approach and insights of an implementation thereof as foundation to achieve our long-term goal of evolving EA models. In our implementation we extend a collaboration platform with capabilities to monitor the actual information demand and to maintain the EA model referring to this demand at runtime.

Keywords: Enterprise Architecture, modeling, model evolution, collaboration

1 Motivation

Enterprise Architecture (EA) management is a discipline addressing the immanent need for mutual alignment of business and IT to react upon frequently changing market conditions [19]. The discipline seeks to capture and manage a holistic view of the enterprise to strategically plan enterprise transformations with respect to both, business and IT. Current research efforts increasingly address the situated nature of EA management (EAM) with respect to the organizational culture and the environment [1, 6]. Main motivation of these approaches is the configuration of an EAM function that is tailored to the context of an organization and the goals as well as concerns of the EAM function and respective stakeholders. Developing an organization-specific EAM function requires an EA information model that covers the changing information demands of stakeholders [6]. This information demand depends on the maturity of the EA initiative and the specific context of an organization. Current EA tools do not support this development appropriately due to inflexible information models and missing integration of stakeholders in the modeling process [15]. We argue that the development of EA models can highly benefit from the involvement of stakeholders at an early stage (cf. e.g. [11]). Increased stakeholder involvement in combination with flexible information models are promising means to facilitate evolving EA models [16].

In practice, organizations struggle with over-sized EA information models that often do not meet the information demand [13]. Based on a literature review Lucke et al. [14] reveal scoping of EA information models as a key challenge in EAM; they describe an over-scoping or over-modeling of an EA. While over-scoping describes the missing focus on the necessary concepts in the model [2], over-modeling leads to an overuse of details not knowing which information is really relevant [2]. Based on a large empirical basis this challenge has also been recently validated by Hauder et al. [13]. Due to EA models that do not focus on the actual demand of stakeholders, benefits of EAM are not clearly visible, particularly in the initial phase of an initiative. Simultaneously, enterprises struggle with a huge effort of data collection and a bad quality of EA model data [20]. While recent approaches tackle these challenges by automating the EA documentation [10, 12, 9], leaner EA information models that better fulfill the information demand of stakeholders would be beneficial to reduce the actual amount of documentation.

Although current literature identified these challenges, existing solutions neglect the collaborative effort that is required to develop and maintain the EA model. Armour et al. [2] diagnose that the team's morale suffers when results are not shown early on and further recommend to define plans that deliverables can be shown within weeks, not months. Since information demand and knowledge about the EA is distributed over a potentially large number of stakeholders [5] and systems [4], we aim at providing a solution to capture and merge contributions of these stakeholders. While first approaches towards automated EA documentation [9] did not include stakeholders in this knowledge-intensive process, subsequent research introduced a process model for the collaborative resolution of conflicts in the course of the modeling process [21]. However, the evolution of EA information models at runtime by involving stakeholders appropriately remains an unresolved issue. As a reaction we present a solution that is based upon a collaboration platform with modeling capabilities. Goal of our efforts is to incorporate stakeholders' knowledge in the modeling process to facilitate evolution of EA information models.

2 Modeling challenges for Enterprise Architectures

Documenting and modeling an EA faces several challenges in practice since a multitude of EA Stakeholders and information sources are involved in these processes. The documentation of the EA is concerned with the collection of the required information through interviews with information providers and imports from operative systems respectively existing excel files as information sources in the organization [20]. Information required for the documentation is defined during the modeling of the EA information demand for Decision Makers. Figure 1 illustrates these EA Stakeholders and possible information sources that interact with each other. We conducted an extensive literature study in order to reveal these EA documenting and modeling challenges in detail. In the following we will summarize these challenges with respect to particular EA Stakeholders. Stakeholders for the modeling and documentation of the EA are concerned with a variety of different challenges. In this paper we distinguish between three major roles and assign the identified challenges accordingly.

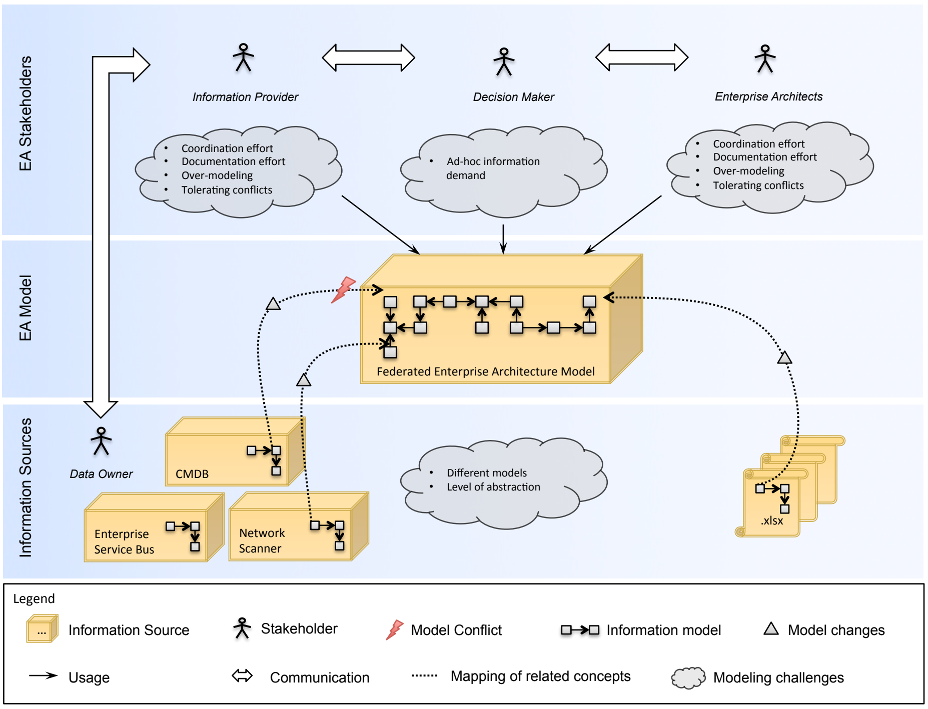

Fig. 1. Challenges for EA modeling and documentation at runtime

2.1 Information Providers

Are mainly responsible for collecting the information about the EA manually or support during the automated documentation from other operative systems [21]. Information Providers in enterprises are often faced with a huge *coordination effort* [14]. As a result, the acceptance of EAM may become a challenge [22] and the high number of involved parties can lead to insufficient Information Provider involvement or buy-in [2]. This reluctance of Information Providers also may turn into their unavailability [17], a development that takes place in particular if architectural activities have been already preceded by expensive but unsuccessful EAM endeavors [8]. The *documentation effort* with cost-intensive gathering, maintaining, and disseminating of EA information is discussed in detail by Buckl et al. [7]. It is first and foremost the initiation process of EAM that imposes considerable investments. Information sources have to be identified and assessed before data can be collected and stored by means of dedicated software. Detached from the domain of EAM, Wieland et al. [24] report on *tolerating conflicts* to foster collaboration. While in traditional software development version conflicts should be immediately resolved, conflicts in models that are used in an informal manner to develop a common language need to be tolerated and assessed collaboratively before they can be resolved eventually.

2.2 Decision Makers

Decision Makers require EA information that are typically analyzed by means of visualizations. Schmidt et al. [22] highlight it often takes years to make significant progress such that meanwhile it is often immeasurable. They consider this delay of tangible results an important reason why the EAM discipline lacks legitimation. In many cases stakeholders expect a return on investment much earlier than the discipline is eventually able to deliver [8]. Missing legitimization and late delivery often translate into little value perceived by stakeholders; in particular since they do not understand the real benefits immediately [13]. The fulfillment of *ad-hoc information demands* of Decision Makers is important to circumvent with these challenges to legitimate EAM expenses.

2.3 Enterprise Architects

Need to support Decision Makers by providing a reliable information base. Lucke et al. [14] point especially to the lack of experienced architects, missing management commitment, problems for the EAM team in *understanding requirements*, insufficient tool support as well as rapidly changing environmental conditions as main challenges for EAM. Furthermore, they call the reader's attention to problems arising with EAM scoping, stakeholder coordination and communication as well as complexity especially when it comes to modeling [14]. An issue frequently perceived in EAM is the decoupling of actual requirements on the one hand and delivered outcome on the other. As one consequence, Van der Raadt et al. speak of the *ivory tower* syndrome leading to situations where too complex EA information models possess an inappropriate level of abstraction [23]. While the phenomenon of over-modeling is observed by Armour et al. [2], the issue of *over-scoping* has been pointed out by Lucke et al. [14]. In addition, Chuang et al. [8] warn against the imminent danger of architectural work isolation. According to the authors, Enterprise Architects tend to operate and communicate in silos instead of communicating with the stakeholders continuously and closely. Another challenge pertains to the *late valuation of benefits*. Ross et al. estimate that an organization requires between two and six years to absorb the cultural and technical changes caused by the introduction of EA management entirely [19].

3 A meta-information model for runtime evolution

Dealing with the aforementioned challenges requires EA information models that evolve with respect to the maturity of the EA management function in the organization and changing information demands of stakeholders. Figure 2 illustrates the core concepts in our approach allowing the evolution of EA models at runtime.

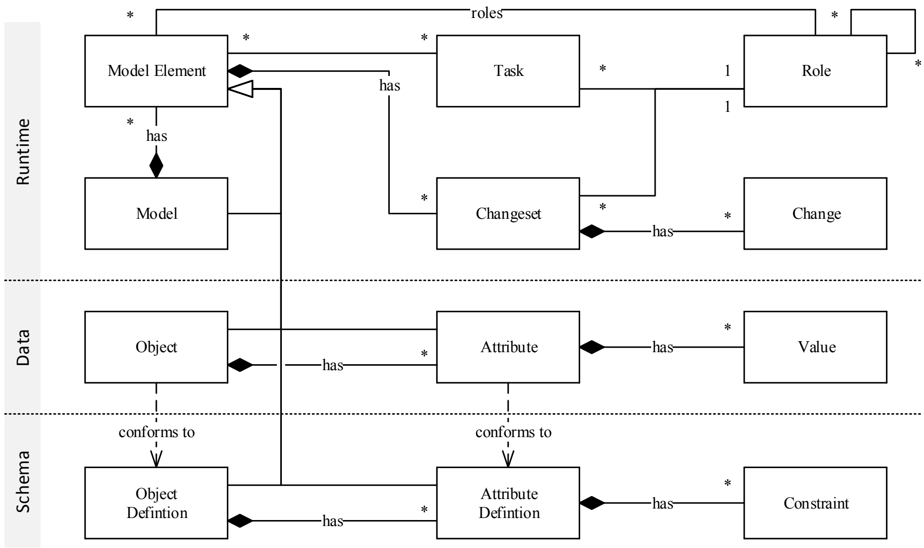

Fig. 2. A meta-information model facilitating model evolution at runtime

This meta-information model is divided into elements on the data, schema and runtime level. The data layer captures objects and attributes with the values containing the actual EA model. Data elements that conform to definitions on the schema level are referred to as mandatory elements in our model. Optional elements consist solely on the data layer and conform to no specific definition in the schema layer. The evolution of the model at runtime is facilitated through tasks that are assigned to these model elements and responsible roles conducting them. These tasks are used to turn 1) model extensions, i.e. the creation of objects or attribute definitions, and 2) potential model conflicts into collaboration by involving the roles introduced in Section 2. While the former is triggered by its creation, the latter is detected immediately as conflicts occur during concurrent operations. Table 1 describes all task types that facilitate runtime evolution of the model. These task types are triggered by model modification events in the system during write operations. Table 2 illustrates automatically generated tasks as these events occur.

Table 1. Tasks for the evolution of Enterprise Architecture models at runtime

Task	Description
Assign Role	is concerned with the assignment of the responsible role, readers, and writers to a particular object definition or object instance. If defined, it is sent to the responsible role of the object definition or object; otherwise, the EA repository owner is notified. The EA repository owner then defines a responsible role such that in any way readers and writers are assigned by the responsible role. Tasks in brackets mean either the writer is asked whether it is necessary to trigger this task, or the system checks, e.g. if roles are already set, and decides automatically.
Document	asks the writers to maintain a certain object or attribute values thereof. This task is automatically sent to writers as soon as an attribute is set as strict.
Validate	refers to the validation of particular attributes or entire objects. This can be done by any writers such that these are informed by default. Due to their write access, writers are immediately able to correct flaws in the data. As soon as a certain amount of writers defined as threshold have validated a concept, the responsible role is informed.
Approve	is required to approve certain model changes by the responsible role. For instance, deletions of entire objects must be approved, or changes of certain attributes/values that the responsible role is accountable for, e.g. changes of the service level.
Resolve Conflict	is perhaps the most complex tasks since multiple parties must be involved in a synchronous manner in order to decide on pending model changes.
Merge	seeks to merge multiple changes into one coherent model state. Since details of merging strategies are beyond the scope of this paper, we adopt the general strategy of Wieland et al. [24] to store any concurrent model changes and subsequently show these changes with the original version to the end-users.
Propagate	in the vein of a federated EA model, after merging or approving, this tasks asks to change (delete or update) a value in the information source, i.e. propagate changes to an information source. This can be done either automatically via technical interfaces, or manually by the assigned role.

For instance it is a difference to rename the entire attribute (schema manipulations have global impact) or just correct a minor typo within a value (instance manipulations have local impact). Also, the assignment of roles could be sometimes necessary, e.g. when introducing a new concept. Default values can be derived by the system sometimes, e.g. for attribute definitions the default behavior is to inherit the access rights of the respective object definition whereas objects and attributes inherit access rights of the respective definition. End-users are free to refine these derived default access rights. Thereby, we distinguish between the maintenance of objects, attributes, values, attribute definitions, and object definitions.

Table 2. Automatically generated tasks during the modification of the model at runtime

	Assign Role	Document	Validate	Approve
Create Value				
Create Attribute Definition	(x)	x	x	
Create Attribute				
Create Object Definition	(x)		x	
Create Object				
Update Value				
Update Attribute Definition		(x)	(x)	
Update Attribute				
Update Object Definition			(x)	
Update Object				
Delete Value				(x)
Delete Attribute Definition				(x)
Delete Attribute				(x)
Delete Object Definition				
Delete Object				x
Move Object				
Use Object				

During the creation of an object definition and attribute definition an *assign role* task is generated to determine which role is responsible for these elements. *Document* tasks are automatically generated when an attribute definition is created or updated and assigned to the responsible roles in the system. They are attached to all objects having this attribute in the associated object definition. Validation tasks are generated and forwarded to the responsible role in case many constraint violations appear for an attribute. Deleting attributes, objects, or definitions can lead to the generation of an approval task that is assigned to the responsible role of the concerned element.

4 Tool support for collaborative model evolution

Figure 3 illustrates a subset of an instantiated EA model based on the meta-information model presented in Section 3 to exemplify the evolution of an EA model at runtime. In our scenario, the initial information model requires an adaption due to new information demands from stakeholders at runtime. The required adaption is highlighted with a dashed box containing a new attribute *business function* for the given object. The presented EA information model only consists of *application components* which are hosted on *infrastructure* systems. In our example an accounting system is deployed on a cloud service including the required roles for the management (responsibility) of this information.

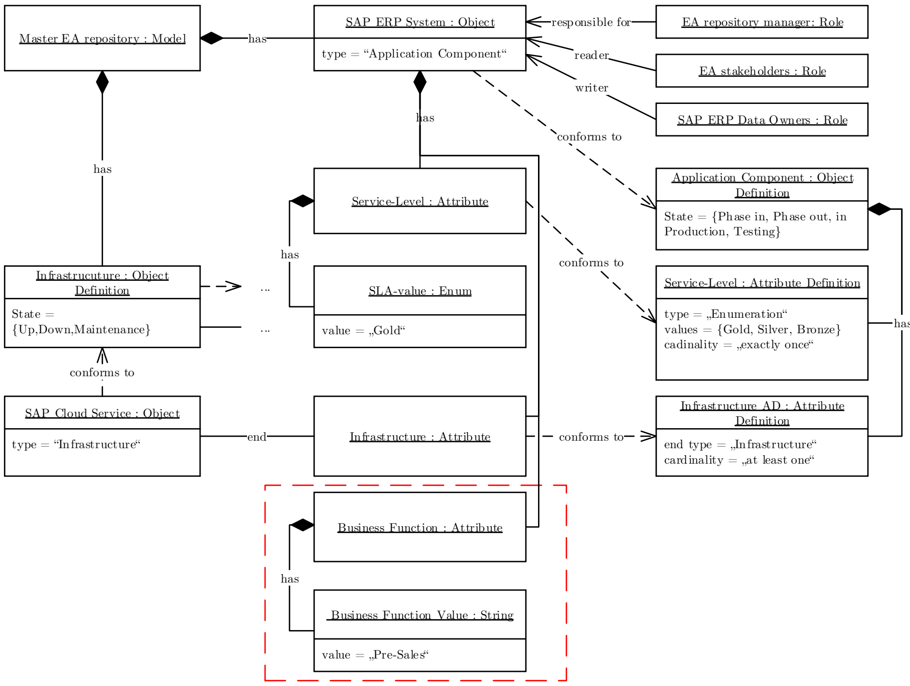

Fig. 3. Adapted EA information model containing a new business function

These basic concepts are described using an application component definition and an infrastructure definition. An application component has a defined state it currently operates in. Similarly, an infrastructure can be in a predefined state, e.g. up, down, or maintenance. In addition, an application component has an attribute definition for the service-level. Another attribute definition is used to describe the relationship between the accounting system and the infrastructure cloud service. The *repository manager* is responsible for this particular element and has to ensure the quality. Some *EA stakeholders* and the *data owners* can read this information respectively conduct changes to the related elements. Due to new external requirements, stakeholders want to know which application components support which business functions. In the initial version of the information model business functions were not covered. Therefore, the EA stakeholders create a new attribute *business function* with respective values. In the example the accounting system supports a business function for *pre-sales*. Note that at this point the stakeholders did not change the schema since no attribute definition was created, but they are able to maintain values for the business function on their own. Since the information model in Figure 3 tends to be rather complex and against the background that domain models often tend to show unnecessary complexity [3], we provide a simplified view on the adapted EA information model that is shown in Figure 4.

Its purpose is to provide a graphical representation of the model that eases the communication of changes to the model to end-users in a comprehensible manner; hence the reduced complexity. In analogy to UML class diagrams, it contains an overview of concepts used in the EA information model based on objects, attributes, respective definitions and values. This view incorporates not only defined and derived concepts but distinguishes between *undefined concepts* that do not have a definition, *defined concepts*, i.e. attributes and objects with a respective definition, and *derived concepts*, e.g. types or cardinalities that can be guessed by the system based on the instances. As illustrated, objects are shown with information about their actual usage (number of instances) including attributes and number of instances, respectively. To foster model extensions, Figure 4 shows end-users the actual frequency an attribute is used with respect to the number of total object instances. According to their frequency and, thus, end-user adoption, attributes then can be set as strict by an EA repository manager. In line with Renger et al. [18] we advocate that these model extensions must be performed by a modeling expert.

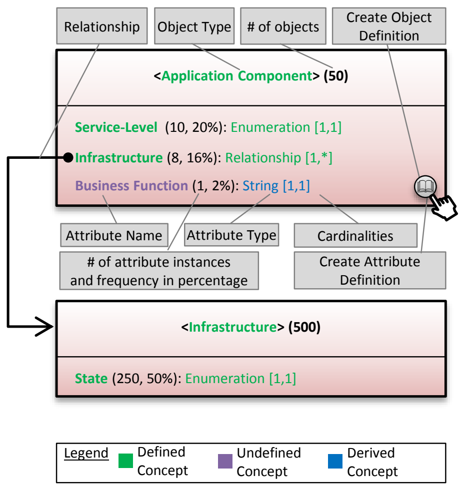

Fig. 4. Graphical representation of the EA information model

As soon as the newly created attribute business function is set as *strict*, i.e. a corresponding attribute definition is created by an expert, respective values must conform to their attribute definition; type as well as cardinality constraints are checked for validity. For any invalid value, a validate task is sent to respective writers in an automated manner. The writer is notified about the conflicting situation and performs corrections. In turn this means invalid values are not discarded for strict attributes but shown to the users to facilitate the conflict resolution. The EA repository manager sets the responsible role such that two *assign role* tasks are created for the responsible role in order to set readers and writers for the newly created concept. Also, *document* tasks are sent to writers of objects for which so far no value for the attribute *business function* has been maintained.

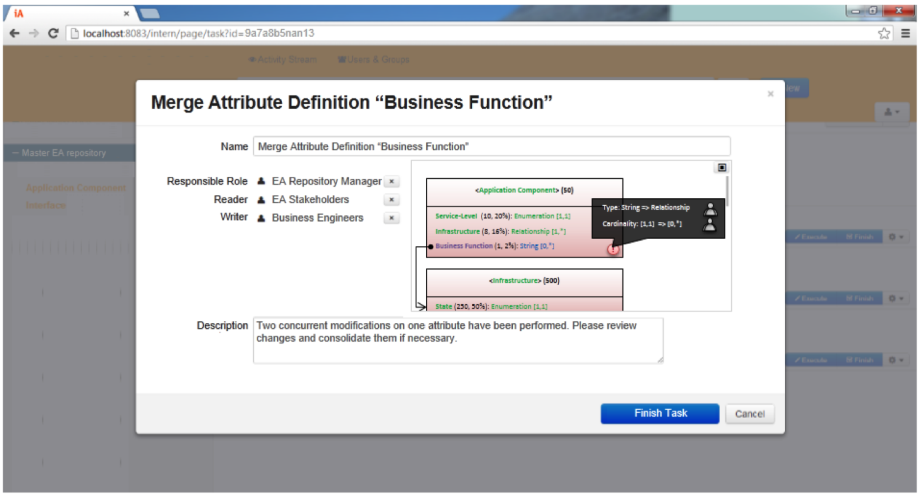

Fig. 5. Automatically generated merge task after the update/update model conflict

A conflicting situation might appear if an administrator deletes the SAP ERP System object from the model and, at the same time, another writer responds to the maintenance task by creating a value for the attribute *business function*. This attribute is attached to the SAP ERP System object. This might lead to a conflict situation as information not known to the administrator is now created and appended to the object. Thus, an approve task is automatically sent to the responsible role in order to resolve this potential conflict. In our example, the role EA repository manager is actually owned by two different persons both maintaining the EA model. Both decide to alter the newly created attribute definition for the business function attribute. While the first repository manager decides to set the cardinality constraint to (1..n) the other repository manager alters the attribute to a relationship. As a result an update/update model conflict occurs on a schema level and a resolution task is sent out immediately that is shown in Figure 5. In line with Renger et al. [18], we believe that a model expert is required to resolve such issues such that sophisticated, perhaps graph-based, strategies may be employed to ease the merging task but not to resolve it entirely without a model expert. An exclamation mark shows model conflicts; by clicking this icon details of the respective task, e.g. a merge task, are shown and the affected changesets and respective changes are given allowing the expert to consolidate concurrently performed changes.

5 Conclusion

Organizations struggle with EA models that are often over-sized and do not meet the information demand of stakeholders. In this paper we presented an approach that empowers stakeholders to collaboratively reveal their information demand. With the presented approach the EA model can evolve with changing requirements of stakeholders. Main advantages of our approach are early benefits and a reduced documentation effort in the early stages of an EA initiative. We detailed the notion of tasks with respect to maintenance and validation tasks to dynamically extend an EA information model and foster consistency in an EA information model. Future steps may address issues arising when approaching a federated EA modeling. Especially concurrent model and metamodel changes pose new challenges to an evolving EA modeling approach.

Acknowledgment

This research has been sponsored in part by the German Federal Ministry of Education and Research (BMBF) with grant number TUM: 01IS12057.

References

1. Ralf Abraham, Stephan Aier, and Robert Winter. Two speeds of eama dynamic capabilities perspective. In *Trends in Enterprise Architecture Research and Practice-Driven Research on Enterprise Transformation*, pages 111–128. Springer, 2012.
2. Frank J Armour, Stephen H Kaisler, and Simon Y Liu. Building an enterprise architecture step by step. *IT professional*, 1(4):31–39, 1999.
3. Colin Atkinson and Thomas Kühne. Reducing accidental complexity in domain models. *Software and System Modeling*, 7(3):345–359, 2008.
4. Ruth Breu. Ten principles for living models - a manifesto of change-driven software engineering. *2010 International Conference on Complex, Intelligent and Software Intensive Systems*, 0:1–8, 2010.
5. Sabine Buckl, Florian Matthes, Sascha Roth, Christopher Schulz, and Christian M. Schweda. A conceptual framework for enterprise architecture design. In *Workshop Trends in Enterprise Architecture Research*, 2010.
6. Sabine Buckl, Florian Matthes, and Christian M Schweda. A method to develop ea modeling languages using practice-proven solutions. *Advances in Enterprise Engineering V*, pages 91–105, 2011.
7. Sabine Buckl and Christian M. Schweda. On the state-of-the-art in enterprise architecture management literature. Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Germany, Munich, Germany, 2011.
8. Cheng-Hui Chuang and Johan van Loggerenberg. Challenges facing enterprise architects: A south african perspective. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10. IEEE, 2010.
9. Matthias Farwick, Berthold Agreiter, Ruth Breu, Steffen Ryll, Karsten Voges, and Inge Hanschke. Automation processes for enterprise architecture management. In *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2011 15th IEEE International*, pages 340–349. IEEE, 2011.

10. Matthias Farwick, Ruth Breu, Matheus Hauder, Sascha Roth, and Florian Matthes. Enterprise architecture documentation: Empirical analysis of information sources for automation. In *46th Hawaii International Conference on System Sciences (HICSS), Maui, Hawaii*, 2013.
11. Max Fiedler, Matheus Hauder, Alexander Schneider, and Florian Matthes. Foundations for the integration of enterprise wikis and specialized tools for enterprise architecture management. In *11th International Conference on Wirtschaftsinformatik (WI), Leipzig, Germany*, 2013.
12. Matheus Hauder, Florian Matthes, and Sascha Roth. Challenges for automated enterprise architecture documentation. In *Trends in Enterprise Architecture Research and Practice-Driven Research on Enterprise Transformation*, pages 21–39. Springer, 2012.
13. Matheus Hauder, Christopher Schulz, Sascha Roth, and Florian Matthes. Organizational factors influencing enterprise architecture management challenges. In *21st European Conference on Information Systems (ECIS), Utrecht, Netherland*, 2013.
14. Carsten Lucke, Sascha Krell, and Ulrike Lechner. Critical issues in enterprise architecting a literature review. In *AMCIS 2010 Proceedings*, pages 1–11. Association for Information Systems, 2010.
15. Florian Matthes, Sabine Buckl, Jana Leitel, and Christian M Schweda. Enterprise Architecture Management Tool Survey 2008. Technical report, Technical Universtity Munich, 2008.
16. Florian Matthes, Christian Neubert, and Alexander Steinhoff. Hybrid wikis: Empowering users to collaboratively structure information. In *Proceedings of the 6th International Conference on Software and Data Technologies*, pages 250–259, 2011.
17. A Nakakawa, P van Bommel, and HA Proper. Challenges of involving stakeholders when creating enterprise architecture. In *5th SIKS/BENAIS Conference on Enterprise Information Systems*, pages 43–55, 2010.
18. Michiel Renger, Gwendolyn L. Kolfschoten, and Gert-Jan de Vreede. Challenges in collaborative modeling: A literature review. In Jan L. G. Dietz, Antonia Albani, and Joseph Barjis, editors, *CIAO! and EOMAS, held at CAiSE 2008*, volume 10 of *Lecture Notes in Business Information Processing*. Springer, 2008.
19. Jeanne W Ross, Peter Weill, and David Robertson. *Enterprise architecture as strategy: Creating a foundation for business execution.* Harvard Business Press, 2006.
20. Sascha Roth, Matheus Hauder, Matthias Farwick, Ruth Breu, and Florian Matthes. Enterprise architecture documentation: Current practices and future directions. In *11th International Conference on Wirtschaftsinformatik (WI), Leipzig, Germany*, 2013.
21. Sascha Roth, Matheus Hauder, and Florian Matthes. Facilitating conflict resolution of models for automated enterprise architecture documentation. In *Americas Conference on Information Systems (AMCIS)*, 2013.
22. Christian Schmidt and Peter Buxmann. Outcomes and success factors of enterprise it architecture management: empirical insight from the international financial services industry. *European Journal of Information Systems*, 20(2):168–185, 2010.
23. Bas van der Raadt, Sander Schouten, and Hans van Vliet. Stakeholder perception of enterprise architecture. *Software Architecture*, pages 19–34, 2008.
24. Konrad Wieland, Philip Langer, Martina Seidl, Manuel Wimmer, and Gerti Kappel. Turning conflicts into collaboration - concurrent modeling in the early phases of software development. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, tba:1–52, 2012.