

# Pattern-based Design Research in Enterprise Architecture Management

Sabine Buckl<sup>1</sup>, Florian Matthes<sup>2</sup> Alexander Schneider<sup>2</sup>, and  
Christian M. Schweda<sup>1</sup>

<sup>1</sup> iteratec GmbH, Inselkammerstr. 4, 82008 München-Unterhaching, Germany,  
{sabine.buckl,christian.schweda}@iteratec.de,  
<http://www.iteratec.de>

<sup>2</sup> Technische Universität München, Chair for Software Engineering for Business  
Information Systems,  
Boltzmannstr. 3, 85748 Garching, Germany,  
{matthes,schneial}@in.tum.de,  
<http://wwwmatthes.in.tum.de>

**Abstract.** Enterprise architecture (EA) management represents an evolving discipline in the curriculum of information systems research. Research in the area of EA management is often conducted in close cooperation with industry partners, who on the one hand provide input in terms of current challenges and on the other hand are more than willing to apply and evaluate the research findings. Therefore, researchers in the area of EA management applying the design science paradigm are confronted with the challenge to make theoretical contributions which additionally can be applied and help to solve current and anticipated problems in practice.

In this paper we present the so-called pattern-based design research (PDR) an iterative design research method to overcome this problem by enabling researchers to theorize and learn from the intervention at the industry partner(s) while performing rigorous and relevant design science research. We illustrate the applicability of PDR by discussing our research projects of the last 7 years in the area of EA management. We conclude that PDR provides a suitable research method that can be applied in the area of EA management and discuss further challenges from the perspective of practitioners and researchers.

## 1 Motivation

Enterprise Architecture (EA) Management is an evolving management discipline within the curriculum of Information Systems (IS). The discipline targets at the holistic management of the enterprise, in particular its architecture formed by elements as business processes, applications, information flows, hardware, as well as their interrelations. Researchers active in the field of EA management are faced with various challenges of high practical relevance. For these challenges, the researchers seek to develop technology-based solutions, i.e., ‘do’ design science

research [1]. Such research has to be conducted in close cooperation with industry experts, in order to:

- identify a problem of practical relevance,
- develop a solution that fits the application context, and
- evaluate the purposefulness of the solution in its context.

During solution development and evaluation, design research has to rely on sound methodologies [2] to ensure rigor of the research results (cf. [3]). This puts researchers in the field of EA management research in a particularly challenging situation. On the one hand enterprises are inherently complex subjects, for which rigorous development of purposeful solutions is likely to take time. On the other hand research in close cooperation with an industry partner has to adapt to the pace of partnering organization, delivering useful results early. Therefore, researchers find themselves in a field, where – as Davenport already states in [4] – rigor and relevance compromise each other to some extent. A key challenge hence is to account for both rigor and relevance in EA management research endeavors with industry. Further, researchers are challenged to ensure that their research does not degenerate into ‘routine design’ that must be distinguished from design science [1].

EA management seeks to establish sustainable solutions to challenges that affect the structure of the enterprise as a whole. Key to this management effort is a holistic perspective on the EA. Usually, EA descriptions, containing different views on the enterprise, are used. These descriptions are again defined on a conceptual model of the enterprise. EA management uses the descriptions to increase transparency about the enterprise [5] and to control the architecture’s evolution [6]. In this context, researchers target, for example, at the identification of ‘good’ architectures (behavioral science) [7] as well as at the development of models [8], methods [9], and tools [10] (design science). Especially for the latter, a close cooperation with industry is essential. Only industry experts are able to raise relevant business problems, provide data about real enterprise architectures and participate in an artifact’s evaluation. To allow for rigor in a close cooperation with an industry partner which allows for relevance inherently, a design science research method has to account for several challenges occurring in such context [11]:

- Research has to provide outcomes, whose level of abstraction matches the expectations of the research community [12]. For research conducted in cooperation with industry this community is heterogeneous consisting of both practitioners and researchers, aiming at different and potentially inconsumerable levels of abstraction.
- Many EA management-related problems are “wicked problems” (cf. [13, 14]). They occur, if the specific contexts of companies relate to asymmetric criteria determining different solutions [15]. To increase the relevance of research outcomes in terms of importance and suitability (cf. [16]), the company’s context has to be taken into account when performing EA management research with an industry partner.

In this paper, we describe the research method of *Pattern-based Design Research* (PDR). This method uses patterns to provide early and relevant results to practitioners and design theories to rigorously abstract these results to scientific contributions. We repeatedly applied this method to different problem dimensions in EA management, like the management processes and the description techniques. Latter application is described in more detail in this paper to show by example, how PDR can facilitate EA management research. Section 2 introduces the basic conception of PDR, referencing relevant groundwork from both pattern research and design research. The course of a research as well as the results thereof are described in Section 3 starting with early results on patterns and concluding with their abstraction to building blocks. In Section 4 we discuss challenges from both academic and practitioner direction, that we had to face applying the PDR method. Further, we give an outlook on possible future applications of PDR to EA management and briefly sketch the research ecosystem that would be needed to facilitate PDR in the context of EA management.

## 2 The Pattern-based Design Research Method

The Pattern-based Design Research (PDR) method introduced in [11] and visualized in Figure 1 provides a research approach for a close cooperation of researchers and industry experts suitable for the context of EA management.

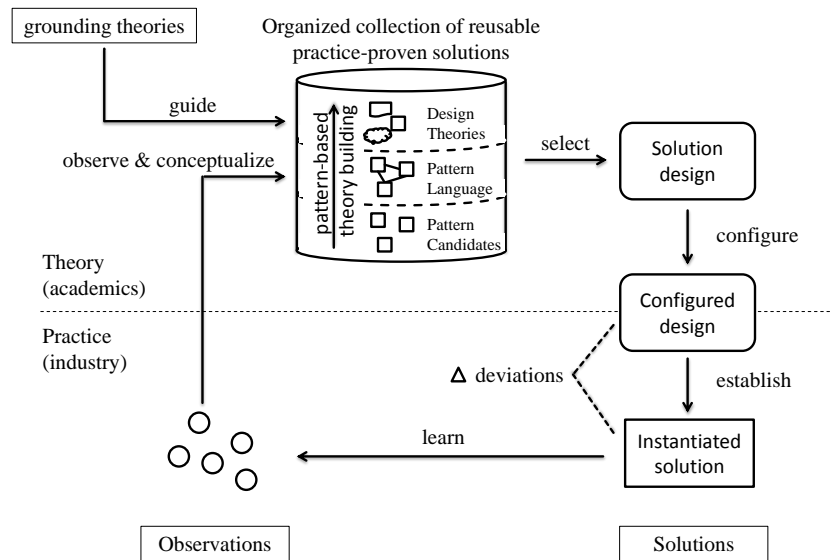


Fig. 1. The PDR method [11]

By the observation of common solutions to reoccurring EA management problems in practice and their conceptualization the basis for an EA management

Design Theory Nexus (DTN) is formed. The identified patterns (cf. [17, 18, 19]) as well as related grounding theories are then used to build design theories [20, 21]. By the conceptualization of the context in which a pattern is suitable as well as the specific goals, a DTN is built. Industry partners willing to use the practice-proven knowledge provided by the DTN can design their organization-specific EA management function by using their context and goals as input. The resulting solution design can then be applied to the organization’s context. For example, the *solution design* can be configured by adjusting the used terminology. The resulting *configured solution* serves as a blueprint to be implemented by the organization. The implementation includes the realization of processes as well as the provision of tool support if required. After the actual implementation of the configured solution we speak of an *instantiated solution*. This differentiation allows, for example, to distinguish a process’s design from its instances. Of course, the instantiated solution is subject to evolution when the organization runs it over time. If process instances differ from their original design, a better solution has been implemented by the organization itself. This solution might, for example, suite better to the organizations context than the solution design created by the use of the DTN. In general, the derivations between the configured and the instantiated solution can be used as a basis to extend the knowledge contained in the DTN. In addition to adjustments of already known solutions, even completely new solutions can be observed this way. Such an observation closes the cycle of the PDR method and a new iteration begins by the solution’s conceptualization to form a pattern candidate.

### 3 Applied PDR – Course of Action

Our research on Software Cartography, EA management patterns, and Building Blocks for EA management Solutions has followed the PDR method since 2004. Close cooperation with more than 30 industry partners over the years has allowed us to identify, design, refine, and validate many relevant solutions. Being at least partially industry-funded, the patterns as well as tooling based on the patterns has helped us to deliver useful solutions to our partners. In the following, we present the different stages of our research and outline both directly applicable results and the theoretic underpinnings developed by us. Table 1 gives an overview of the different research artifacts.

In 2004 we started a research project on visualizing application landscapes. This project was called “software cartography” [22] and could, due to the support of Prof. Dr. Denert, start with a series of interviews with practitioners. In these interviews, we identified the challenge of missing and not structured techniques for visualizing application landscapes. We further were able to collect exemplary visualizations that some of the partnering enterprises created without applying re-usable techniques. In the publications [23, 24], we identified four types of “software maps”, called “cluster map”, “process support map”, “time-interval map”, and “graph-layout map”. For all these types we not only described typical application scenarios, but also outlined best-practices applying to their practical

Year	Applicable results	Theoretic underpinnings
2004	Software Map types (1)	
2005	Software Map types (2)	IEEE 1471 for Software Cartography
2006	Software Cartography tool	Transformation perspective on Software Cartography (1)
2007		Transformation perspective on Software Cartography (2)
2008	EA management Pattern Catalog Wiki	Structure and style of EA management patterns
2009	Workshop: Patterns in EA management	
2010	Building Blocks for EA management Solutions (BEAMS)	Refining the ISO 42010 Meta-Language for EA Information Modeling
2011	BEAMS Wiki	Development of Organization-Specific EA Modeling Languages using Building Blocks

**Table 1.** PDR for EA management – course of action

application. Without having them called so explicitly, we documented the first *pattern candidates*. These pattern candidates provided the basis for continuing research in cooperation with almost 30 industry partners over the next five years.

From a more theoretic perspective, we experienced a significant lack of means in describing and formalizing how the different visualizations in an EA description interrelate. From work with the practitioners we learned that different roles in an organization are using the different visualizations for different purposes, i.e., to answer different relevant questions. The theoretical underpinnings from the field of software engineering, more precisely the “documentation of software-intensive systems” (cf. IEEE Std. 1471 [25]) provided a terminology to structure and formalize the pattern candidates. In [26] we explored software cartography as a means to document software-intensive enterprises and motivated the concepts *concern* and *stakeholder* related to the *viewpoints*, as which we understood the software map types.

Giving account to the importance of EA modeling and data collection we focused our research in the years of 2005 and 2006 on issues related to the automated creation of EA visualizations. Thereby, we identified issues in visualization handling during an extensive survey of existing tools for EA management [27]. To overcome the error-prone and time-consuming task of manual creation of EA viewpoints, we propose a model transformation approach, which links the data to be visualized and their graphical representations. The fundamental principles of software cartography, an approach for EA modeling, including a method for the automatic creation of visualizations based on EA models is presented in [28, 29].

In [30] we sketch an approach to designing organization-specific information models for EA management based on patterns. This approach is intended to

manage the complexity of such models, to which the wide spread domains (e.g. processes, technical architecture, strategic issues) that are involved in EA management contribute. Our contribution in this respect lies in the introduction of patterns into the field of EA management information models. Thereby, other approaches, as e.g. structuring the information models into layers, are complemented by the possibility to reuse pre-existing solutions to address EA management issues. Having provided an initial example, how such a pattern could look like, we embarked to describing more patterns, as we have encountered them in our industry projects.

Extending the idea of patterns to the area of EA management from a comprehensive perspective, we introduce three different types of EA management patterns: Information Model Patterns (I-Pattern), Viewpoint Patterns (V-Patterns), and Method Patterns (M-Patterns) [31]. M-Patterns document proven-practice methods to address typical problems in EA management. V-Patterns represent best-practice EA visualizations, and I-Patterns indicate information requirements for EA management. The EA management patterns thereby follow the typical structure of patterns (cf. [32]) containing e.g. a name, context, problem, solution, and consequence descriptions. The different EA management patterns are interlinked as e.g. a V-Pattern is referenced within the method description of an M-Pattern and the V-Pattern links to the I-Pattern defining the information demands underlying a respective V-Pattern. Thus, the different patterns build up a pattern language for EA management.

The EA management pattern catalog [33] documents EA management patterns for management methods, viewpoints, and information models as well as the dependencies between these patterns and the thereby addressed concerns. These patterns are presented in a consistent terminology and information organization to simplify the selection, adaptation, and integration of patterns. The catalog incorporates the findings from earlier research in the project Software Cartography and the EA management Tool Survey 2005 [27]. Subsequently the patterns were evaluated using an on-line questionnaire to identify methodologies and viewpoints that are considered relevant and useful by practitioners. The EA management Pattern Catalog tries to find a balance between a green field approach and a completely predefined approach as provided by some EA management frameworks and EA management tools. It avoids a giant integrated information model but utilizes a consistent terminology and a common organization to permit an understanding and comparison of multiple approaches from different sources. Sample viewpoints help readers to grasp essential concepts.

With the first workshop on “Patterns in Enterprise Architecture Management” (PEAM 2009) we seek to set the stage for EA management patterns also in the scientific community. The introductory words in [34] refrain the growing interest in academia and industry to “identify, collect, document, and exchange best-practices” for EA management. The PEAM workshop addresses European EA management researchers and practitioners to exchange their knowledge using patterns as means of documentation. An EA management pattern “is a general, re-usable solution to a common problem in a given context which iden-

tified driving forces, known usages, and consequences”. Six paper presented in the workshop address various topics ranging from processes for EA management, over modeling of smart enterprise networks, to viewpoint patterns for EA management in the context of mergers and acquisitions. The specialized format of the workshop ensured relevance and applicability of the presented solutions. During the review both scientists and practitioners were invited to judge relevance and contribution of the solution. Afterwards, a “pattern shepherd” was assigned to the authors of each accepted paper, providing guidance on the writing style of the paper. In the workshop, no comprehensive presentation was given by the authors, but well-prepared attendants discussed aspects of understandability and presentation. Their feedback was incorporated into the final papers to ensure that other readers of the patterns are able to understand and apply the presented solutions.

Focusing again more on the theoretic underpinnings of EA and EA management, we further investigated architectural descriptions in 2010. Architectural descriptions representing and modeling the architecture of a system or parts thereof are typically used in the engineering disciplines to plan, develop, maintain, and manage complex systems. Primarily originating from construction engineering, the means of architecting and architectural descriptions have been successfully transferred to related disciplines like software engineering. While a rich and formal theory on conceptual modeling exists as well as frameworks on how to approach architectural descriptions, e.g. the ISO standard 42010 [35], only few attempts have yet been made to integrate the prescriptions and guidelines from these sources into a formal architectural description framework. In [36], we establish such a framework against the background provided by the ISO standard 42010 [35] by formally defining the terms concern, view, viewpoint, and architectural description.

In the light that different EA descriptions are based on viewpoints that again entail different underlying information models, we analyze these information models in [37]. In particular, we focus on the meta-languages used for prevalent information models and diagnose their diversity. For the analysis we apply the work of Guizzardi [38], who established a theoretic base for ontological modeling. Based on the analysis of the used meta-languages, we elicit requirements for a “unifying meta-language”. The requirements for these meta-language in turn are based on recurring phenomena in EA information modeling, e.g. “hierarchy modeling” for element hierarchies or “multi-level modeling” for ontological instantiation of elements. We further reflect that multi-purpose modeling facilities, like the UML, fail to satisfy these requirements. The requirements further shape the path for subsequent theoretic research on how to describe EA information models.

The enduring need for applicable standards in the field of EA management and the lack of a commonly accepted understanding of scope, reach, and focus of EA management motivate our work in [39]. We reflect on the different organizational structures that raise implications on EA management and demand an enterprise-specific realization of the management function. We further

reflect that pattern, while providing re-usable solutions, lack guidance on their integration. In response to these deficiencies we describe the building blocks for EA management solutions (BEAMS). BEAMS provides practical guidance for organizations to support the design and development of an organization-specific EA management function by presenting method and language building blocks. These building blocks are selected based on the concerns, goals, and context of the using organization. They are then integrated following structured integration procedures based, e.g. on the technique of method engineering [40, 41]. Alongside we discuss how BEAMS gives hints for researchers willing to contribute to the discipline of EA management, and outline the theoretic foundations of BEAMS.

To ensure that the best practices contained in BEAMS are available to a wide audience of practitioners, the organized library of building blocks was made available using an enterprise 2.0 system<sup>1</sup>. In the BEAMS wiki each building block is documented by a single wiki page. Furthermore, the pages are equipped with hybrid templates that describe the characteristics of the corresponding building block in a structured manner [42].

The thesis [43] presents the complete method for developing organization-specific EA modeling languages to support EA management addressing the challenges of enterprise transformation. The method builds on a library of language building blocks resulting from the PDR over the years. These building blocks are organized according to a conceptual framework of the EA. We distinguish three types of building blocks: “information model building blocks” (IBB) describing the abstract syntax, “viewpoint building blocks” (VBB) describing the applicable concrete syntax, and “glossary building blocks” (GBB) providing textual semantic definitions for the concepts. The method further builds on a set of formalisms to integrated IBBs into a comprehensive and organization-specific information model. Additional formalisms help to apply the VBBs on the information model and to evolve the GBBs into a terminology aligning to the terms used in the organization. All formalisms are based on the theoretic underpinnings of two domain-specific meta-languages, one for information modeling and one for viewpoint definition. In addition, we provide guidelines how to consistently evolve and adapt existing EA description languages developed by this method.

## 4 PDR – Experiences, Challenges and Outlook

In this paper, we outlined our Pattern-based Design Research (PDR) method and their successful application to the field of EA management, in particular EA modeling. We experienced both the benefits of applying the method as well as challenges in application, which we describe subsequently.

The practice-driven research and the early delivery of results as patterns was highly appreciated by the industry partners supporting and funding our research.

<sup>1</sup> The BEAMS wiki is available at <http://wwwmatthes.in.tum.de/pages/ste22z023rd3/BEAMS> (last accessed 2013-04-03).



In particular, the viewpoint patterns described in the pattern catalog have been applied as means to determine, which ‘images’ of the architecture would be useful to have. A number of (invited) talks on the topic [44, 43] and several publications in ‘yellow press’ [45, 46] reflect the positive reaction of practitioners regarding the EA management patterns. The pattern catalog wiki further promoted the findings and more than 1000 registered users accessing the patterns in 2010 (2 years after setup of the wiki) give an indication on the practical impact of the findings [47].

Nevertheless, these registered users did not form the active community that we expected them to form. We enabled the users to comment on patterns and to document known uses in practice. Further, we invited users with a track-record in EA management research and practice to contribute their findings and results as patterns. After two years, only a small number of comments (less than 20) were added to the wiki, and in addition to the patterns received at the PEAM workshop only one pattern was documented by an invited user. This in particular ascribes to the fact that EA management practitioners in a single company have difficulties in observing the same pattern repeatedly, i.e., three times. Practitioners with the possibility to observe solutions at different organizations, e.g. consultancies, on the other hand are not that likely to share their overarching insights in the field.

In publishing the patterns that we observed, identified, and documented in our research we experienced several difficulties. In early publications in the IS environment, reviewers were susceptible to the idea of applying patterns in the field of EA management. A similar discussion on the applicability of pattern in the context of IS research can be found in [48]. When it came to the presentation of single patterns, outlets with a strong focus on rigorous research were not likely to accept, what the reviewers called “non-representative cases”. Different reviewers advised us to perform empiric analyses, but thereby did not acknowledge the fact that empiric analyses targeting novel design solutions are hard to undertake. Pattern conferences and workshops, like the PLoP conference series<sup>2</sup> conversely provided an outlet receptive for pattern-based documentation of results. The feedback received in the reviews and on-site nevertheless did not focus on the relevance and applicability of the solution, but on its documentation, i.e., the ‘patternness’ of its documentation. In our opinion nevertheless, feedback on all three facets of a pattern solution would be required, i.e., feedback on:

- the practical *relevance* of the presented context, problem, and solution,
- the *theoretic groundwork*<sup>3</sup> on which context, problem, and solution description rely, and
- the *patternness* of the documentation, i.e., the understandability to a practitioner.

Our PEAM workshops were an attempt to ensure that feedback regarding all three facets can be provided. A program committee of both practitioners and

<sup>2</sup> See <http://hillside.net/conferences> (last accessed 2014-04-03).

<sup>3</sup> Grounding theories

researchers in the particular field review relevance of the pattern and appropriateness of the theoretic foundations. Once a pattern is regarded relevant and sufficiently grounded, a shepherd supports the authors to improve the pattern-like presentation. This presentation in turn is subjected to a final discussion at the writers' workshop, at which practitioners and researchers provide final feedback. With the novel format of the workshops, we were able to overcome what we experienced to be deficiencies of 'classic' publication outlets, when it comes to patterns.

The PDR method has proven its applicability in the context of EA management, but is in its nature more universally applicable. Disciplines and research fields with high practical relevance, wicked design problems, and the need to perform research in cooperation (and with the funding) of practitioners are likely susceptible for this research approach. Possible fields may be design solutions for cloud computing, enterprise-wide knowledge management, or business intelligence. The method does further not rely on a completely open presentation of the pattern candidates and patterns: while a receptive and critic external audience promotes the formulation of relevant solutions, also consultancies may take advantage from applying the PDR method. Pattern candidates and patterns can in this context be useful to document the findings from different consulting projects, and to consolidate the relevant knowledge in a knowledge base for the organization.

## References

1. Hevner, A., March, S., Park, J., Ram, S.: Design science in information systems research. *Management Information Systems Quarterly* **28**(1) (2004) 75–105
2. Benbasat, I., Weber, R.: Rethinking 'diversity' in information systems research. *Information Systems Research* **7** (1996) 389–399
3. Gallupe, R.B.: The tyranny of methodologies in information systems research. *SIGMIS Database* **38** (2007) 20–28
4. Davenport, T.H., Markus, L.M.: Rigor and relevance revisited: Response to benbasat and zmud. *Management Information Systems Quarterly* **23**(1) (1999) 19–23
5. Hanschke, I.: *Strategic IT Management: A Toolkit for Enterprise Architecture Management*. Springer (2009)
6. Murer, S., Bonati, B., Furrer, F.J.: *Managed Evolution*. Springer (2010)
7. Niemi, E., Pekkola, S.: Enterprise architecture quality attributes: A case study. In: 46th Hawaii International Conference on System Sciences (HICSS). (2013) 3878–3887
8. Frank, U.: Domain-specific modeling languages – requirements analysis and design guidelines. In Reinhartz-Berger, I., Sturm, A., Clark, T., Wand, Yair AND; Cohen, S., Bettin, J., eds.: *Domain Engineering: Product Lines, Conceptual Models, and Languages*, Springer (2013)
9. Riege, C., Aier, S.: A contingency approach to enterprise architecture method engineering. In: *Pre-Proceedings of the 3rd Workshop on Trends in Enterprise Architecture Research (TEAR 2008)*, Sydney, Australia (2008)
10. Matthes, F., Neubert, C., Schneider, A.W.: Fostering collaborative and integrated enterprise architecture modelling. *Enterprise Modelling and Information Systems Architectures* **8**(1) (2013)

11. Buckl, S., Matthes, F., Schneider, A.W., Schweda, C.M.: Pattern-based design research – an iterative research method balancing rigor and relevance. In: Proceedings of the 8th International Conference on Design Science Research in Information Systems and Technology. (2013)
12. Vaishnavi, V., Kuechler, W.: Design research in information systems. <http://desrist.org/design-research-in-information-systems> (2004)
13. Churchman, C.W.: Guest editorial: Wicked problems. *Management Science* **14**(4) (1967) B141–B142
14. Rittel, H., Webber, M.: Dilemmas in a general theory of planning. *Policy Sciences* **4** (1973) 155–169
15. Pries-Heje, J., Baskerville, R.: The design theory nexus. *Management Information Systems Quarterly* **32**(4) (2008) 731–755
16. Rosemann, M., Vessey, I.: Toward improving the relevance of information systems research to practice: the role of applicability checks. *Management Information Systems Quarterly* **32**(1) (2008) 1–22
17. Alexander, C.: Notes on the Synthesis of Form. Harvard University Press, Cambridge, MA, USA (1972)
18. Gamma, E., Helm, R., Johnson, R., Vlissides, J.M.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional (1994)
19. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-oriented software architecture: a system of patterns. John Wiley & Sons, Inc., New York, NY, USA (1996)
20. Buckl, S., Matthes, F., Schweda, C.M.: Utilizing patterns in developing design theories. In: 2010 International Conference on Information Systems (ICIS 2010). (2010)
21. Buckl, S., Matthes, F., Schweda, C.M.: From ea management patterns towards a prescriptive theory for designing enterprise-specific ea management functions – outline of a research stream. In: Multikonferenz Wirtschaftsinformatik (MKWI2010), Gttingen, Germany (2010) 21–22
22. sebis: Jahresbericht lehrstuhl für informatik 19. <http://wwwmatthes.in.tum.de/file/1md32csjj4emf/sebis%20Public%20Website/Publications/sebis-Ganzjahresbericht-2002-2004.pdf> (2004) last accessed 2013-04-03.
23. Matthes, F., Wittenburg, A.: oftwarekartographie: Visualisierung von anwendungslandschaften und ihrer schnittstellen. In: Informatik 2004 - Informatik verbindet, 34. Jahrestagung der GI, Ulm, Germany (2004)
24. Lankes, J., Matthes, F., Wittenburg, A.: Softwarekartographie: Systematische darstellung von anwendungslandschaften. In Ferstl, O., Sinz, E., Eckert, S., Is-selhorst, T., eds.: *Wirtschaftsinformatik 2005*, Physica-Verlag (2005) 1443–1462
25. IEEE: Recommended Practice for Architectural Description for Software-Intensive Systems. IEEE Std. 1471-2000 (2000)
26. Lankes, J., Matthes, F., Wittenburg, A.: Softwarekartographie als beitrag zum architekturmanagement. In Aier, S., Schnherr, M., eds.: *Unternehmensarchitekturen und Systemintegration*. Volume 3. Gita, Berlin, Germany (2005)
27. Matthes, F., Buckl, S., Leitel, J., Schweda, C.: Enterprise Architecture Management Tool Survey 2005. TU Mnchen, Chair for Informatics 19, Prof. Matthes (sebis) (2005)
28. Ernst, A.M., Lankes, J., Schweda, C.M., Wittenburg, A.: Using model transformation for generating visualizations from repository contents – an application to software cartography. Technical report, Chair for Informatics 19 (sebis), Technische Universitt Mnchen, Munich, Germany (2006)

29. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F., Schweda, C.M., Wittenburg, A.: Generating visualizations of enterprise architecture using model transformation (extended version). *Enterprise Modelling and Information Systems Architectures – An International Journal* **2**(2) (2007) 3–13
30. Buckl, S., Ernst, A., Lankes, J., Schneider, K., Schweda, C., Wittenburg, A.: A pattern based approach for constructing enterprise architecture management information models. In: 8. Internationale Tagung Wirtschaftsinformatik, Karlsruhe, Germany (2007) 145–162
31. Ernst, A.: Enterprise architecture management patterns. In: *Pattern Languages of Programs Conference 2008 (PLoP08)*, Nashville, USA (2008)
32. Meszaros, G., Doble, J. In: *A Pattern Language for Pattern Writing*. Addison Wesley (1998) 529–574
33. Buckl, S., Ernst, A.M., Matthes, F.: Enterprise architecture management pattern catalog. Technical report, Technische Universität München, Munich, Germany (2008)
34. Matthes, F., Ernst, A.: Patterns in enterprise architecture management (peam). In Münch, J., Liggesmeyer, P., eds.: *Software Engineering 2009 – Workshopband*, Bonn, Germany, *Lecture Notes in Informatics (LNI)* (2009) 17–18
35. International Organization for Standardization: *ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems* (2007)
36. Buckl, S., Krell, S., Schweda, C.M.: A formal approach to architectural descriptions – refining the iso standard 42010. In Aalst, W., Mylopoulos, J., Sadeh, N.M., Shaw, M.J., Szyperski, C., Albani, A., Dietz, J.L.G., eds.: *Advances in Enterprise Engineering IV*. Volume 49 of *Lecture Notes in Business Information Processing*., Springer Berlin Heidelberg (2010) 77–91
37. Buckl, S., Matthes, F., Schweda, C.M.: A meta-language for ea information modeling - state-of-the-art and requirements elicitation. In: *Enterprise, Business-Process and Information System Modeling - 15th International Conference on Exploring Modelling Methods in Systems Analysis and Design (EMMSAD 2010)*, Hammamet, Springer (2010) 169–181
38. Guizzardi, G.: *Ontological Foundations for Structural Conceptual Models*. Telematica Instituut, Enschede, The Netherlands (2005)
39. Buckl, S., Dierl, T., Matthes, F., Schweda, C.: Building blocks for enterprise architecture management solutions. In: *Practice-Driven Research on Enterprise Transformation, second working conference, PRET 2010*, Delft, Springer (2010) 17–46
40. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. *Information and Software Technology* **38**(4) (1996) 275–280
41. Harmsen, A.F.: *Situational Method Engineering*. PhD thesis, University of Twente, Twente, The Netherlands (1997)
42. Matthes, F., Neubert, C.: Wiki4eam: Using hybrid wikis for enterprise architecture management. In: 7<sup>th</sup> International Symposium on Wikis and Open Collaboration (WikiSym), Mountain View, California (2011)
43. Schweda, C.M.: *Development of Organization-Specific Enterprise Architecture Modeling Languages Using Building Blocks*. PhD thesis, Technische Universität München (2011)
44. Buckl, S.M.: *Developing Organization-Specific Enterprise Architecture Management Functions Using a Method Base*. PhD thesis, Technische Universität München, München, Germany (2011)

45. Buckl, S., Ernst, A.M., Schweda, C.M.: Eam patterns: Best practices for enterprise architecture management. *Architecture & Governance Magazine* **5** (2009)
46. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F.: Eam pattern catalog liefert best practices. *isReport* **5** (2008)
47. Ernst, A.M.: A Pattern-Based Approach to Enterprise Architecture Management. PhD thesis, Technische Universität München, München, Germany (2010)
48. Winter, R., vom Brocke, J., Fettke, P., Loos, P., Junginger, S., Moser, C., Keller, W., Matthes, F., Ernst, A.M.: Patterns in business and information systems engineering. *Business & Information Systems Engineering* (2009) 468–474