Inf Syst E-Bus Manage (2014) 12:101–137
DOI 10.1007/s10257-012-0211-y

ORIGINAL ARTICLE

Overcoming implementation challenges in enterprise architecture management: a design theory for architecture-driven IT Management (ADRIMA)

Jan Löhe · Christine Legner

Received: 21 April 2012 / Revised: 3 December 2012 / Accepted: 7 December 2012 /
Published online: 3 February 2013
© Springer-Verlag Berlin Heidelberg 2013

Abstract Enterprise architecture management (EAM) is acknowledged as a discipline to drive organizational change, to improve IT landscapes' transparency, and to align business and IT. Despite its increasing popularity in practice, many EAM initiatives are confronted with substantial challenges, as demonstrated by the low usage level of enterprise architecture (EA) documentation and enterprise architects' lack of authority, and often fail. This motivates our research, which aims at developing a design theory that may guide organizations to successfully implement EAM. Based on three field studies, we first analyze the issues that arise when implementing EAM in practice. We find that EAM often suffers from being regarded as a separate and parallel initiative, although it needs to be embedded in established management processes and organization. We then suggest a design theory for architecture-driven IT management (ADRIMA) that synthesizes prescriptive knowledge related to embedding EAM practices, artifacts, and roles in the existing IT management processes and organization. By consolidating both IT management and EAM perspectives, our research goes beyond existing EA literature and EA frameworks which describe EAM as a stand-alone management concept focusing on EA models and the EA life cycle.

J. Löhe
Institute of Research on Information Systems (IRIS),
EBS Business School, Konrad-Adenauer-Ring 15,
65187 Wiesbaden, Germany
e-mail: jan.loehe@ebs.edu
URL: http://www.ebs.edu/iris

C. Legner (✉)
Faculty of Business and Economics (HEC),
University of Lausanne, Campus Dorigny,
1015 Lausanne, Switzerland
e-mail: christine.legner@unil.ch
URL: http://www.hec.unil.ch/isi

Keywords Enterprise architecture management (EAM) · EAM implementation · IT management · IS design theory · Design science research

1 Introduction

Enterprise architectures (EAs) are considered promising means to align the required changes in corporate strategy and business processes with an increasingly complex IT landscape (Schmidt and Buxmann 2011; Aier et al. 2008b). They have become one of the CIO's top priorities and are also used as a vehicle to help reduce IT expenses under the increased pressure from business leaders (Luftman and Ben-Zvi 2011). The publication of the Zachman Framework (Zachman 1987) in the 1980s initiated an extensive discourse on EA in research and practice (see Schelp and Winter 2009; Schönherr 2009): Early work focused on enterprises' fundamental components, their relationships as well as their appropriate representations (e.g., Scheer 1991; Österle 1995; Ferstl and Sinz 1997; Frank 1995). While this work centered on the "what" in terms of enterprise modeling methods and notations (see Aier et al. 2008b), little attention was paid to the set-up and implementation of EA concepts in organizations (the "how").

In the meantime, enterprises have gathered practical experience with EA concepts: they started documenting their EA on different layers (Lankhorst et al. 2005; Winter and Fischer 2007) and assigned responsibilities for the further development of EA to dedicated architecture teams as well as roles, such as enterprise architects (Strano and Rehmani 2007). Today, enterprise architecture management (EAM) is acknowledged as a discipline to drive organizational change, to improve IT landscapes' transparency, and to align business and IT (Winter and Schelp 2008; Kappelman et al. 2009; Schmidt and Buxmann 2011).

Despite the diversity of approaches that define either EAM processes and governance regimes or derive EA viewpoints and applications based on stakeholder concerns, EAM implementation remains a challenging topic for organizations However, EAM initiatives are associated with substantial challenges, and often fail as researchers (Morganwalp and Sage 2004; Seppanen et al. 2009) and practitioners (Roeleven and Broer 2009) report. Some of the criticism against EAM is that it requires a lot of effort (Morganwalp and Sage 2004) and that the benefits thereof are not directly measurable and are realized with some time lag (Schmidt and Buxmann 2011). Moreover, EA's strategic alignment potential is hindered by lacking governance, insufficient support for the EA development from business and IT management, as well as by inadequate resources and skills (Seppanen et al. 2009; Winter and Schelp 2008). These EAM implementation issues motivate our research, which will first analyze the key challenges when implementing EAM in practice. From three field studies, we conclude that EAM often suffers from being regarded as a separate and parallel initiative, although it needs to be embedded in established management processes and organization. In order to address this issue, our research aims at developing prescriptions that may guide organizations to successfully implement EAM. We formulate our results as a design theory (Gregor and Jones 2007; Walls et al. 1992), which we call the design theory for architecture-driven IT management (ADRIMA). This article extends ADRIMA's first version (Löhe and Legner 2012), which focused on the constructs, representing domain-specific vocabulary and conceptualizations, and a set of design principles, i.e. principles of form and function, to embed EAM in IT processes and the existing organizational structures. The extended version of ADRIMA is a comprehensive and well-defined design theory which encompasses all eight design theory components specified by Gregor and Jones (2007): (1) purpose and scope, (2) constructs, (3) a set of design principles, (4) artifact mutability, (5) justificatory knowledge, (6) testable propositions, (7) expository instantiation, and (8) principles of implementation. Thereby it elaborates on how artifacts could be constructed and why the design works, (the first six "core components" of a design theory), but also addresses the design theory's implementation in specific circumstances. More specifically, this work enhances ADRIMA's initial version by anticipating artifact changes ("artifact mutability"), describing an instantiation as proof-of-concept ("expository instantiation") and developing specific guidelines for building further instantiations ("principles of implementation"). It is noteworthy that ADRIMA focuses on EAM as an integral part of IT management, although we also acknowledge EAM's broader enterprise focus. Our research setting concentrates on IT management, which—similar to other empirical studies in the field (Schmidt and Buxmann 2011)—involves IT-led EAM initiatives.

In the remainder of the paper, we first summarize the current state of research on EAM implementation and analyze to what extent IT management literature has picked up EAM concepts. We then motivate and introduce our research approach. In Sect. 4, we present the observations from three field studies. Based on these practical insights, we derive general design requirements and evaluate to what degree current approaches satisfy the requirements. We then introduce our design theory for ADRIMA. We conclude with a brief summary and discuss our contribution to the existing body of knowledge as well as the implications for future research.

2 Current state of research and related work

Early publications related to enterprise architectures centered on the "what" in terms of enterprise modeling methods and notations. With the more widespread adoption of EAM in practice, there is increasing interest in how EA concepts should be implemented in organizations (the "how"). The following sections review how EAM implementation is conceptualized in the existing literature and subsequently reflects EAM's role in IT management.

2.1 EAM implementation

Enterprise architectures provide means to document and communicate an enterprise's fundamental components and their relationships through formal conceptualizations, i.e. architectural descriptions or models (Maier et al. 2004; Lankhorst et al. 2005). Whereas there is no common understanding of the terms EA or EAM (see Schelp and Winter 2009; Schönherr 2009), most authors agree that

EAM is a management and design function targeting the enterprise in a comprehensive manner (Buckl 2011; Winter 2004; Aier et al. 2009a). In doing so, EAM supports the enterprise's transformation from the current (as-is) towards intermediate and long-term planned (to-be) EA states (Schelp and Winter 2009; Aier et al. 2008b; Buckl et al. 2010a, 2011). Moreover, EAM sets clear directions for enterprises through plans and roadmaps as well as principles and standards that guide the transformation (Spewak and Tiemann 2006; Ross et al. 2006).

While the benefits are widely acknowledged, EAM implementation in a company is challenging (Zink 2009; Seppanen et al. 2009) and may take several years to produce the expected outcomes (Schmidt and Buxmann 2011). In order to leverage EAM, companies must adopt and institutionalize the concept in their organizations. This implies that they explicitly manage the EA life cycle, conceptualize and document the EA in the form of models, and introduce new governance regimes. From our qualitative content analysis (Mayring 2000) of prior EAM literature, we have identified four research streams with different conceptualizations of EAM implementation (Table 1):

1. EAM initiatives, i.e. the setup of an EAM program that centers on EA models and their life cycle,
2. EAM processes, i.e. approaches that prescribe recurring processes or activities that enterprises should establish for using EAM,
3. EAM application scenarios, i.e. stakeholder-oriented approaches that focus on specific concerns and viewpoints, as well as
4. EAM governance, i.e. approaches that emphasize EAM roles and governance regimes.

The first research stream includes established EA frameworks and literature proposing building blocks of EAM programs with a strong focus on architecture development and transition. For example, the Federal Enterprise Architecture Framework (FEAF) advises a top-down EAM implementation for US authorities' architectural segments (e.g., tactical defense, energy supply or education). It contains a general EA process (CIO Council 2001) covering the EA program initiation, EA development with baseline and target architectures, as well as EA use, maintenance and control. FEAF comprises performance, business, service component, data, and technical reference models (FEA Program Management Office 2007a), and focuses on aligning EA with capital planning and investment control as well as with system development processes. The Open Group Architecture Framework (TOGAF) suggests an architecture development method (ADM) that represents a continuous cycle (The Open Group 2009). Starting with preliminary steps and an architecture vision, the ADM goes on to develop baseline and target architecture for business, information systems, and technology, followed by opportunities and solutions, as well as migration planning. Finally, it closes with implementation governance and architecture change management. Bricknall et al. (2006) identify critical factors for EAM initiatives: Among others, they refer to top management buy-in, implementation of an EA governance process and EAM's alignment with other enterprise life cycle processes, such as the investment process. They advise to start small and use a step-by-step approach, with an agreed relevant scope and understandable deliverables between business and IT.

Table 1 EAM implementation in the literature

Approach	Description	Examples
1. EAM initiatives	Setup and building blocks of EAM programs. Mostly, model-driven, i.e. focusing on a dedicated EA life cycle (as-is EA documentation, to-be EA planning, etc.)	*EA frameworks and model-driven EA approaches*: TOGAF (The Open Group 2009)*, FEAF (CIO Council 2001; FEA Program Management Office 2007b)*, SEAM (Wegmann et al. 2007), Zachman (Zachman 1987; Sowa and Zachman 1992)* *EAM initiatives' scope and success factors*: Bricknall et al. (2006), Bussells (2006), Janssen and Hjort-Madsen (2007), Matthee et al. (2007), Seppanen, et al. (2009)
2. EAM processes	Typical EAM processes and activities that enterprises should establish when introducing EAM	*EAM activities and functions*: Buckl et al. (Buckl et al. 2009b, 2010a), Schmidt and Buxmann (2011), van der Raadt and van Vliet (2008) *EAM processes*: Dern (2007)*, Hafner and Winter (2008), Keller (2007)*, Niemann (2005)*.
3. EAM application scenarios	Stakeholder-oriented approach defining the relevant concerns and viewpoints as well as application of EA models	*EAM patterns*: Buckl et al. (Buckl et al. 2008; Buckl et al. 2010b), Moser et al. (2009)* *Concern-driven EA modeling*: Lankhorst et al. (2005) *Exemplary usage of EA models*: Bucher et al. (2006), Aier et al. (2008a), Hjort-Madsen and Pries-Heje (2009).
4. EAM governance	EAM roles and committees as well as the introduction of EAM principles and standards	*EAM roles*: Strano and Rehmani (2007), Niemi (2007), Sauer and Willcocks (2002). *Organizational structures for EAM*: Findlay (2006), Venkatesh et al. (2007), Hoogervorst (2009) *EAM principles and standards*: Boh and Yellin (2007), Winter and Schelp (2008), Greefhorst and Proper (2011)

* Practitioner-oriented literature

Process-driven approaches provide descriptions of recurring EAM processes and activities. Certain authors (Buckl et al. 2009b, 2010a) concentrate on individual EAM activities, for example, describing and developing target state EA as well as analyzing and evaluating EA. Others define EAM-related processes on a macro (Keller 2007) or micro level (Dern 2007). Keller (2007) suggests high-level architecture processes for IT strategy, modeling, IT application portfolio management, project monitoring, policy development and enforcement, as well as project portfolio monitoring, and introduces a comparison matrix between his architectures processes and COBIT. Dern (2007) describes detailed workflows for IS portfolio analysis and planning, a comprehensive EA planning, as well as EA development including its initialization and the system development life cycle.

The third stream of research ("EAM application scenarios") emphasizes the necessity to address specific stakeholder concerns and viewpoints when implementing EAM. These works represent exemplary usage of EA models in specific application scenarios, for example, EA impact or coverage analysis to support risk management and sourcing decisions (Aier et al. 2008a; Bucher et al. 2006). In line with this, EAM patterns (Buckl et al. 2008) and concern-driven EA modeling (Lankhorst et al. 2005) relate EA artifacts and methods to a particular stakeholder's perspective. These approaches focus on EAM best-practice solutions to specific needs, without providing an organizational implementation method. To address this gap, Buckl et al. (2011) recently suggested a situational EAM design for selecting appropriate EAM patterns.

Finally, the argument of governance-driven approaches is that introducing EAM requires new organizational or governance structures (Winter and Schelp 2008; Strano and Rehmani 2007; Hoogervorst 2009). They concentrate mainly on the positioning and responsibilities of EAM roles and committees, whereas Hoogervorst (2009) also provides a comprehensive framework for integrating enterprise (architecture) governance into the broader scope of IT and corporate governance. The other governance-driven approaches emphasize defining, implementing, and monitoring EA principles (Greefhorst and Proper 2011) and standards (Ross 2003; Boh and Yellin 2007). They argue that the enforcement of EA principles and standards will coordinate technology choices and project-level decisions across different business units and IT departments in order to keep the over-arching architecture consistent.

2.2 IT management and EAM

Despite the increasing relevance of EAM for IT executives (Luftman and Ben-Zvi 2011), the interplay between EAM and IT management is not well understood. IT management deals "with a broad cross-section of issues spanning the activities from system inception, through design, development, system implementation, and beyond to post implementation evaluation" (Booth and Philip 2005). The plan, build, and run phases are very popular to structure these activities (Zarnekow et al. 2006). Although the plan and build phases comprise architectural design and development, IT management literature has paid very little attention to EAM so far. It either focuses on the IT department's organization and processes in detail and in their functional context (Österle et al. 1993; Boddy et al. 2005; van Schaik 1985) or concentrates on specific realms, such as IT strategy (Riempp et al. 2008; Earl 1988). Among the few approaches that explicitly mention architectural issues and tasks are the popular practitioner-oriented IT management frameworks (see van Bon and Verheijen 2006; ISACA 2011), notably capability maturity model integration (CMMI), IT infrastructure library (ITIL), and control objectives for information and related technology (COBIT).

CMMI (Software Engineering Institute 2010) is a maturity model aimed at improving the software development process. It suggests various related IT practices, such as project planning, requirements development, and technical solutions. While it mentions product architectures, architecture patterns, as well as architecture standards and documentation for a concrete system, it does not consider other architectural layers apart from technology and it does not extend its focus to enterprise-wide transformation.

As a best practice approach, ITIL (Cabinet Office 2011) describes strategic and operational processes to plan, design, develop, and provide high-quality IT services. ITIL has developed from an IT support and operation context into a holistic IT management perspective, and accordingly has recently started to integrate some first EAM aspects into its current version 3. As part of its service design principles, ITIL includes the technology architecture design, which is defined as "the development and maintenance of IT policies, strategies, architectures, designs, documents, plans and processes for the deployment and subsequent operation and improvement of appropriate IT services and solutions throughout an organization" (Cabinet Office 2011). For the details, it refers to well-known EA frameworks, such as TOGAF, FEAF, and Zachman. Furthermore, the service design principles stipulate to work within the agreed architectural framework and standards, work closely with all architectural roles to ensure maximum benefit from the work done in creating the architecture, and to reuse many of the assets created as part of the architecture. Thereby, ITIL determines the real benefit of the EA from an organization's ability to design and implement projects and solutions in a rapid and consistent manner. However, ITIL does not further detail the EA contents of plans, design documents, and standards.

COBIT (ITGI 2007) is a governance and management framework for information and related technology, which describes IT processes, roles, and artifacts along the following phases: plan and organize, acquire and implement, deliver and support, and monitor and evaluate. Its version 4.1 explicitly mentions an architectural process and roles. The "Define the Information Architecture" process aims at establishing and maintaining an enterprise information model and data dictionary as EA artifacts to enable applications development and decision-supporting activities. Other COBIT processes use those EA artifacts as inputs. Moreover, the COBIT role model involves the chief architect in a RACI matrix (Smith and Erwin 2007) and introduces with the "Determine technological direction" process an architecture board that provides architecture guidelines, gives advice on their application, and verifies compliance. Whereas COBIT integrates certain architectural activities and roles as well as EA artifacts, their description and application remains broadly generic, without detailing EA models, and restricted to the information architecture.

Although existing IT management literature hardly considers EAM, there is an increasing awareness that EAM implies changes to IT management functions and domains: Fischer et al. (2005) and later Wittenburg et al. (2007) consider EAM the glue between IT management-related functions. For the specific case of the BMW group, they provide a holistic and exemplarily overview of the IT management processes. Specifically, they explain how EA information, for example business process models, master maps, and blueprints, are used in and provide inputs and outputs for the adjacent IT management processes.

Riempp and Gieffers-Ankel (2007) analyze and describe how EAM concepts support IT managers' decision-making in implementing, integrating, operating, and further developing complex application portfolios. They suggest multiple relevant viewpoints, notably the IT strategy and IT project management viewpoint, with tailored EA models, such as strategy maps, organization charts, and business process models. Finally, the authors recommend integrating and aggregating the information provided by the EA models into a decision-oriented "dashboard". They propose different indicators, such as architectural standard compliance, functional readiness, and operational excellence indicators, to assess each application and support the application portfolio management.

Correia and Brito e Abreu (2009) derive a formal specification of service-level agreements by integrating IT service management (ITSM) into the EA. From their point of view, integration can be accomplished by sharing a common meta-model between the ITSM and EA repositories and by aligning ITSM and EA components. However, as they do not present their repositories' sharing nor their components' alignment, their concepts remain on an abstract level.

2.3 Research gap

To conclude, most of the EAM literature describes EAM as a stand-alone management concept focusing on EA models and the EA life cycle. Although there is an increasing awareness that EAM affects IT management functions and domains, the current literature has left the connection between EAM and IT management open. Only some practitioner-oriented publications mention EAM's application in the IT management context, but their description and application remains broadly generic or support only specific IT management domains. We identify this gap as a shortcoming in the current literature that provides research potential on how to implement EAM as an integral part of IT management processes.

3 Research approach

The main purpose of our research is to develop prescriptions that may guide organizations to successfully implement EAM. In view of this research objective, we choose design science as research approach which is ideally suited to create artifacts and in order to discover new knowledge how to best design and use artifacts (Baskerville et al. 2009; Iivari 2007). Design science is very popular among EA researchers, who mostly aim at constructing design artifacts, such as EA models and methods. As outlined in our literature review, it is also applied in the context of EAM implementation. While the current stream of constructive EA research is focusing on the design artifacts, we purposefully decided to synthesize our research results as a design theory (Gregor and Jones 2007). This approach allows us to concentrate on the principles inherent in the design of the artifacts and thereby focus on the most critical decisions underlying EAM implementations. Hence, the artifacts constructed in the context of a specific EAM implementation allow for "testing" the design theory, whereas the design theory allows the prescription of guidelines for further artifacts of the same type. We also find that this approach better copes with mutable design artifacts, such as situational artifacts, which are very common in EAM implementations (Buckl et al. 2011; Aier et al. 2009b).

The following section will elaborate on our understanding on design theorizing and its documentation. Based thereon, it will describe the research process that we applied for developing the design theory for ADRIMA.

3.1 Design theorizing

Design theorizing is a problem-driven approach to theory development, which aims at designing purposeful artifacts to solve relevant problems (Baskerville and Pries-Heje 2010; Gregor 2009; Baskerville 2008). Design theories synthesize prescriptive knowledge that is actionable, communicable, and can be developed mutually (Gregor and Jones 2007). Compared to natural and behavioral science theories, they not only represent systems of statements targeted at describing, explaining, and predicting real world phenomena (Bacharach 1989; Dubin 1976), but comprise sets of prescriptive statements to guide effective and feasible design (Walls et al. 1992). Although the development of design knowledge is of high importance (Kuechler and Vaishnavi 2008; Winter 2008) and design science has become a popular stream in IS research (Iivari 2007; Hevner et al. 2004; Hevner and Chatterjee 2010), there is as yet no commonly accepted way of designing and documenting design theories (see Baskerville and Pries-Heje 2010; Fischer et al. 2010). For the purpose of this study, we adopt the recommendations of Gregor and Jones (2007), who propose eight documentation components to provide a "systematic specification of design knowledge":

The *purpose and scope* specifies what the theory is for and what its boundaries are. ADRIMA's purpose and scope emanates from four design requirements, which we derived from field studies. In presenting our design theory, we will concentrate on the *constructs*, which represent domain-specific vocabulary and conceptualizations, and the *principles of form and function*, i.e. the design principles to embed EAM in IT processes and the existing organizational structures. The design principles are the design theory's central aspects, because they represent a reconceptualization from a specific artifact to a class of solutions and capture formalized knowledge that can be reused to develop other instantiations. The design principles are backed by *justificatory knowledge* and *testable propositions*. *Justificatory knowledge* represents kernel and "practitioner-in-use" theories as well as evidence-based justification to inform and explain the design decisions (Gregor and Jones 2007; van Aken 2004). *Testable propositions* form truth statements (Gregor and Jones 2007) or heuristic propositions (van Aken 2004) that can be tested during the theory's instantiations. In addition, we briefly illustrate the *expository instantiation* with an IT demand management process example. We also elaborate on the *principles of implementation* as the means by which the design is brought into being as well as the *artifact mutability,* discussing anticipated changes to the artifact encompassed by the theory. By addressing all eight components, ADRIMA can be considered a well-defined and comprehensive design theory.

3.2 Research process

We developed our design theory using a research process—inspired by the work of Hevner et al. (2004) and Hevner (2007)—covering relevance, rigor, and a design cycle (Fig. 1).

To analyze current EAM implementation in practice and identify the associated challenges, the first step of our research comprised interpretive field studies (Klein and Myers 1999). We observed three large German companies' EAM initiatives for periods of between 10 and 18 months and worked in close cooperation with their EAM teams to develop and implement EAM. Owing to the organizations' size, their regional distribution, and their dependence on the extensive use of IT, these companies were good candidates for EAM implementation. The selection of the companies was driven by purposeful sampling, i.e. their need for a holistic approach to EAM, as well as their willingness to cooperate and make multiple information sources available to researchers. In the course of the field study, we gained in-depth insight into the EA documentation and tools, the setup of the EA initiative, and the methodologies applied. In addition to direct observation, we conducted semi-structured interviews and workshops to grasp the EAM implementation process and its challenges. We thereby followed a pluralistic research approach (Mingers 2001) and used multiple data sources help to verify the "truth" of the results (Lincoln and Guba 1985). From the companies' specific EAM experiences, we were able to generalize typical implementation challenges through triangulation. We completed the first phase—the *relevance cycle*—by deriving a set of design requirements to address these challenges.

During the course of the *rigor cycle*, we compared our explorative insights to the EAM and IT management literature. We found that EAM studies mention similar challenges—which acknowledged the validity of our design requirements—and used the literature to identify justificatory knowledge that can support the development of the design theory.

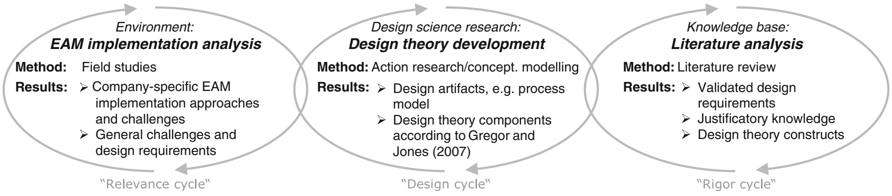

Environment:
EAM implementation analysis
Method: Field studies
Results: ➢ Company-specific EAM implementation approaches and challenges
➢ General challenges and design requirements
"Relevance cycle"
Design science research:
Design theory development
Method: Action research/concept. modelling
Results: ➢ Design artifacts, e.g. process model
➢ Design theory components according to Gregor and Jones (2007)
"Design cycle"
Knowledge base:
Literature analysis
Method: Literature review
Results: ➢ Validated design requirements
➢ Justificatory knowledge
➢ Design theory constructs
"Rigor cycle"

Fig. 1 Research process based on Hevner (2007)

During the *design cycle*, we developed and evaluated the design theory in an iterative action research approach which comprised repeating sequences of diagnosis, action and learning (Baskerville and Wood-Harper 1998). In order to ensure the quality of our research results, we carefully followed the guidelines applied in design theorizing (Hevner et al. 2004; Gregor and Jones 2007). We recorded all comments, design changes, and evaluation results by maintaining a research database to ensure the transparency, consistency, and dependability of our research findings. In the first phase of the design cycle, we helped two of the three companies solve their immediate EAM implementation challenges by conceptualizing IT process models and the supporting EAM practices and documentation ("design artifacts"). We used a consensus-oriented conceptual modeling approach (Becker and Niehaves 2007) to construct and evaluate these company-specific artifacts. In periodic, one-to-four-week intervals, we collaborated on-site with company representatives, notably enterprise architects. In addition, we conducted workshops with executives, experienced project managers and methodology experts to review and validate the artifacts, for example by means of process model walkthroughs. The practitioners contributed their experience from prior (successful and failed) EAM initiatives as well as extensive knowledge about the different organizational contexts in which EAM was to be applied. Their comments helped continuously improve each artifact's design until it was complete, consistent, and viable for implementation in the companies. The continuous review and validation workshops made also sure that the artifacts were being shaped by the investigation objects and not subjectively biased by the researchers. To improve the artifacts' credibility and validity, we corroborated the different design alternatives with the academic and practitioner literature, as well as with the archival documentation of our partners' IT units. We then generalized the findings as a formal design theory by relying on the design and evaluation of purposeful artifacts, since these are regarded as the central basis of design theories (Gregor 2009). Since we were working with different companies and had insights into several EAM implementation contexts within these companies, we were able to analyze the transferability of the artifacts into other settings and could generalize the principles inherent to the design of the artifacts. Based on these data, pattern matching, and a comparison between the findings from multiple real-world settings, we synthesized the findings into a formal design theory. Applying the design theory, by means of expository instantiation, to different EAM implementations allows us to further proof the applicability and credibility as well as to enhance the findings (Gregor and Jones 2007).

4 EAM implementation's challenges and design requirements

In the course of our field studies, we analyzed the EAM implementation approaches in three large Germany-based companies with a historically grown, complex IT landscape (Table 2). Despite several years' experience in EAM, all three companies were facing significant challenges which necessitated major adjustments to their EAM approach. From our observations in the field we generalize the typical challenges they faced and derive a set of design requirements. Finally, we contrast

Table 2 EAM implementation and challenges in three companies

	Company A	Company B	Company C
Company characteristics			
Industry	Real estate	Automobile	Professional services
Description	Real estate service provider with presence in Europe, division of an international group	Large automobile manufacturer with several independent brands and national sales organizations	German subsidiary of a multinational accountancy firm
Annual turnover (EUR)/IT budget %	€950 million/4.6 %	>€100 billion/1.0 %	€1.5 billion/5.2 %
Employees (FTE)	7,000	>300,000	9,000
EAM implementation			
EAM organization	Architecture group in central IT organization unit at division level	Architecture groups with distinct focus (technology, applications) at group IT level	Architecture group in central IT organization unit at subsidiary level
EAM approach	Modeling-driven, based on EAM tool	Driven by modeling, planning, method, and technological standards	Modeling-driven, based on EA framework
EAM methods/EA framework	Method orientation towards EAM tool	Self-developed methods, partly based on EAM tool and TOGAF	Customized, based on FEAF and other approaches, including TOGAF and ITIL
EA meta-model	One, based on EAM tool	Various, based on EAM tools	Customized, based on FEAF
EAM tool and repository	Central repository (Alfabet planningIT)	Several, central repositories (IDS Scheer ARIS, Alfabet planningIT, Sparx EA)	Central repository (IDS Scheer ARIS with own extension)

Table 2 continued

	Company A	Company B	Company C
EA application scenarios	IT operations: Contract management Inventory and document management IT maintenance and migration management	Business process and information management Demand management Application and service portfolio management IT standards and technology management IT landscape planning	Business process and information management Application and service portfolio management IT standards and technology management IT vendor management IT landscape planning
EAM challenges			
Challenges	High initial data collection effort Separated from established processes Unused and obsolete documentation; Unclear responsibilities and application scenarios	Limited top-down coordination Insufficient horizontal coordination Separated from established processes Technical focus Obsolete and inadequate documentation Low user acceptance and comprehension Missing EAM authority	Limited top-down coordination Insufficient horizontal coordination FEAF unsuitable for existing structures Separated from established processes Unused documentation Low user acceptance Missing EAM skills Missing EAM authority

the generalized design requirements of real-world EAM implementations with the existing literature.

4.1 Field studies

Company A, one of the largest real estate service providers in Europe, was busy with its third attempt to implement EAM. Its first attempt was aborted in the initial modeling phase. Documenting the complete as-is EA was simply overwhelming in terms of level of effort and detail. A second attempt was started with the goal of capturing the EA on a higher level of abstraction according to the meta-model of the EAM tool planning IT. It resulted in a nearly complete as-is EA documentation, which quickly became obsolete: missing EAM roles meant it was not updated and it was used neither in IT projects nor for analyses and reports. Moreover, a great deal of detailed EA information, for example in manuals, remained outside the EAM tool or with specific stakeholders, which meant the centrally collected EAM documentation was perceived as being of little use. Since the first two EAM initiatives occurred separately from the established IT processes in which the EA had regularly been planned or changed, most IT and business representatives saw little benefit in these initiatives. In the third attempt, the company pursued a pragmatic approach which concentrated on concrete EA application scenarios in the IT operations' process. In view of the high sourcing ratio (with more than 90 % of the annual IT budget being sourced from external providers), company A defined EA models as well as appropriate analyses and reports to use for IT providers' contract management as well as for error handling, migration planning, and maintenance processes.

Company B is one of the world's leading car manufacturers with several independent brands and local sales organizations covering more than 150 countries. It applied EAM to harmonize its group-wide IT landscape across different brands, production facilities and local markets. EAM was not driven by one top-down initiative, but by several individual initiatives in various IT units. In a first step, a group-wide core business process and organizational model was documented with the ARIS tool. This was complemented by technical infrastructure documentation, as well as the definition of technology standards and their enforcement when approving IT investments. In addition, Company B documented existing business applications and introduced a group-wide IT landscape planning methodology. More recently, the EAM gained further momentum from a service-oriented architecture (SOA) initiative in which documentation and modeling methods for domains, reusable IT modules, and technical IT services had been developed. Nevertheless, the anticipated EAM benefits, such as improved business-IT alignment and reusable application services, failed to appear. The organization realized there was a gap between its long-term EA planning and the numerous development projects that constantly change the EA. Additionally, the distributed EAM approach led to scattered EA documentation. The various EAM tools provided redundant and low quality information and inconsistent documentation versions. Company-wide, standardized EA documentation had not yet been implemented, as it was considered an overhead, and the EAM governance focused solely on enforcing technical standards. Motivated by the shortcomings, the organization intended to integrate EAM into a newly established end-to-end IT demand management process. By documenting and analyzing demands based on EA models, the goal was to operationalize EAM, maintain up-to-date EA documentation, and establish proactive IT planning.

The EAM in *Company C*, the German subsidiary of one of the world's leading accountancy firms, was driven by its global EAM strategy. A central EAM goal was to support the business strategy implementation; however, the activities still focused mostly on IT architecture. The German subsidiary was allowed to decide how EAM had to be implemented and selected a model-oriented approach. It developed a meta-model and reference models based on FEAF, covering all relevant areas from IT infrastructure assets to business processes. The initiative focused on assessing the existing application landscape, developing a catalogue of EA principles and standards, as well as establishing governance mechanisms to allow for long-term oriented IT management. During the EAM implementation, it became apparent that the FEAF framework approach was not sufficient, not only because FEAF's reference models are oriented towards US Federal Government, but also because it does not consider a firm's existing organizational structure. Furthermore, very little EA information was used in IT projects due to deficient EAM skills and a relatively slow central EA modeling process. The EAM team had only limited authority to direct the project managers who mainly drive the EA change. Project managers preferred familiar modeling approaches and usually showed resistance to the EAM team's additional documentation requirements. Moreover, a lack of coordination between the EAM initiative and a parallel ITIL initiative created contradictory perceptions and redundant documentation. The required adjustments generated additional effort, which in turn decreased the EAM initiatives' acceptance. As a consequence, the EAM team decided to concentrate on monitoring and reporting on the application and IT service portfolio, as well as IT vendor management.

4.2 General EAM implementation challenges

All three companies initially focused on a modeling-driven EAM approach, as proposed by many EA frameworks. They started by documenting their as-is architecture on different EA layers, but faced common challenges with the modeling-driven EAM implementation:

The first challenge relates to the *effort regarding the initial documentation* of the EA models and the architecture teams' definition of EA standards: The companies recognized that complete EA documentation was not feasible due to the many different stakeholders, the overall organizational complexity, and the too large scope. The high initial effort hampered the willingness to further maintain the EA artifacts. In addition, the companies did not consider mechanisms to update the EA documentation, such as after-project changes. Thus, the EA repositories rapidly became outdated and were perceived as being of low quality.

A second challenge was that *existing EA artifacts remained unused* in daily work and decision-making: The low utilization of the EA documentation was partly due to its poor quality and obsolescence. Additionally, the EAM initiatives provided EA documentation at the wrong levels of granularity and ignored the stakeholders' information needs. Indeed, all the EAM initiatives provided basic EA documentation, but without the appropriate EA artifacts' representation, for example in reports, to support the day-to-day work and decision-making.

A third challenge was the *lack of acceptance in the (IT) organization*: The IT employees thought EAM just required additional effort and that it had no benefits for their own work. The EAM initiatives' unclear goals and the lacking EAM knowledge hampered their support. There was a lack of EAM comprehension due to ambiguous vocabularies between the IT and EAM groups, and because EAM was often perceived as having only a technological focus. In addition, the architects had limited access to the IT decision-making committees, such as project steering committees or change advisory boards. Consequently, they were unable to promote and enforce EA policies and standards in the existing IT processes related to the IT life cycle, i.e. the planning, building, and running processes.

Since the EAM was set up as an independent initiative with a focus on the EA life cycle, it created a management cycle parallel to established IT processes, such as the IT strategy definition, budget and portfolio planning, the IT project delivery, and IT service management. The consequences were *coordination problems and rivalry between the EAM initiative and IT processes*, which already shaped decisions related to planning, changing, and managing the EA.

4.3 Deriving design requirements from general EAM challenges

From these typical and widespread EAM implementation challenges, we were able to derive four general design requirements (Table 3). These design requirements (DR) describe, first, the conditions needed, from a practitioner's viewpoint, to solve problems, and, second, the specification or constraint of the design (see IEEE 1990). Accordingly, the design requirements act as our design theory's purpose and scope, which the design theory's constructs, design principles, and testable propositions must address to overcome or reduce the general EAM implementation challenges.

Table 3 General EAM implementation challenges and design requirements

No.	EAM implementation challenges	Design requirements
1.	Great effort relating to the initial collection as well as EA artifacts being outdated and of low quality	Existing IT processes should continuously produce and maintain EA artifacts
2.	Low usage of existing EA artifacts in day-to-day work and decision-making	Existing IT processes should consume and use appropriate EA artifacts
3.	Lack of EAM acceptance in the (IT) organization and difficulties to enforce EA policies and standards	Existing IT roles and committees should understand EA artifacts as well as assume responsibilities for EAM tasks
4.	Coordination problems because the EAM initiative sets up processes for managing the EA life cycle parallel to established IT processes	EAM should be embedded in existing organizational structure and processes

The first design requirement that existing IT processes should continuously produce and maintain EA artifacts addresses the initial EA documentation efforts as well as the EA artifacts' timeliness and quality issues. In doing so, the design theory must provide functions for the continuous maintenance of EA artifacts. The second design requirement concentrates both on the EAM use and EA artifacts' adequacy. Hence, the design theory must consider EAM regularly as part of day-to-day work and decision-making in existing IT processes. The third design requirement addresses EAM's acceptance and comprehension problems due to IT staff's insufficient EAM awareness and skills. Thus, the design principles must tackle this problem by improving the existing IT roles' and committees' understanding of EA artifacts as well as establishing skills and responsibilities for EAM tasks. The fourth design principle derives from coordination problems between the EAM processes parallel to established IT processes. This design requirement generally emphasizes embedding EAM in established IT processes and organizational structures, instead of setting it up as a separate and parallel initiative.

4.4 Coverage of design requirements by existing EAM literature

We then matched the derived design requirements with appropriate EAM implementation approaches of our earlier review. Table 4 depicts by which means prior studies conceptualize EAM implementation and assesses to what extent the studies satisfy the design requirements.

With respect to DR1, only a few contributions (CIO Council 2001) describe a continuous maintenance concept for EA documentation, while others (Schmidt and Buxmann 2011; Aier et al. 2009a) at best mention its significance. With regard to the consumption and usage of appropriate EA artifacts, almost all identified approaches consider DR2, notably the approaches defining concerns and viewpoints as well as application of EA models ("EAM application scenarios" in our literature review). Aier et al. (2009a) specifically define criteria for adequate EA documentation.

DR3 is partly covered in governance-driven work, (e.g., Boh and Yellin 2007) as well as in practitioner-driven contributions and EA frameworks, although these contributions do not consider the EAM's interplay with existing organizational structures. There is a prominent contention that EAM should be integrated into established organizational structures, but with the exception of Hoogervorst (2009), remains a generic assertion and gives little attention to their EAM awareness and skills.

Some contributions that satisfy DR2 also build on DR4 by presenting specific IT processes with integrated EAM practices and EA artifacts. However, as they focus on specific EAM application scenarios, they have no general prescriptions how to embed EAM in existing processes, nevertheless their artifacts can help to derive design principles for a sub-class of solutions.

To conclude, this fidelity check (see March and Storey 2008) between the existing EAM approaches and the general design requirements demonstrates that there is no adequate solution in the extant knowledge-base. So far, no study has yet formalized design knowledge related to embedding EAM practices and EA artifacts into existing IT processes and organizational structures in an integrated and comprehensive manner. The corresponding studies, however, may contribute to

Table 4 Coverage of design requirements in EAM literature

EAM literature	Research stream	EAM implementation							Design requirements			
		Standard and Principle	Stakeholder Involvement	Org. structure	Roles	Introduction process	EAM processes	EAM application context	1	2	3	4
Bricknall et al. (2006)	I		+			o	o	Investment, system development process	o	+		o
FEAF (2001)	I	+		+	+	o	+	Capital planning and investment control	+	+	+	o
TOGAF (2009)	I	+		o	o		+	Enterprise transformation	o	o	o	
Buckl et al. (2009b, 2010a)	P						+	Portfolio, demand, project life cycle management	o	o		
Dern (2007)	P			+	+	+	o	IS planning, system development process	o	+	+	+
Keller (2007)	P	o		+	o		+	IT strategy, application portfolio management	o	+	+	
Aier et al. (2008a, 2009a)	A			o	o	o	o	Business process optimization, continuity mgt., IT consolidation	o	+		
Bucher et al. (2006)	A						o	Business continuity planning, risk management		+		o
Buckl et al. (2008, 2011)	A			o		o	+	Portfolio mgmt. (respect. EAM patterns)	o	o		
Boh and Yellin (2007)	G	o	+	+	+			Corporate IT management		+	+	
Ross (2003)	G	+	+	+		o	o	Business/IT alignment	o			o

Table 4 continued

EAM literature	Research stream	EAM implementation							Design requirements			
		Standard and Principle	Stakeholder Involvement	Org. structure	Roles	Introduction process	EAM processes	EAM application context	1	2	3	4
Schmidt and Buxmann (2011)	G			+	o		+	System development, IT planning	o	o	+	o
Greefhorst and Proper (2011)	G	+						Enterprise transformation, enterprise engineering		o		
Hoogervorst (2009)	G	o		+	+			Enterprise governance, enterprise engineering			+	o

Research stream: (A)pplication scenario, (G)overnance, (I)nitiative, (P)rocess

Element is addressed: (+) comprehensively, (o) partly/exemplarily identify additional design theory's justificatory knowledge that can be used to derive and explain suitable design principles.

5 A design theory for architecture-driven IT management (ADRIMA)

The design theory for ADRIMA is the formal representation of our research findings. Extending ADRIMA's initial version (Löhe and Legner 2012), we elaborate on the eight components proposed by Gregor and Jones (2007) to specify a well-defined design theory.

5.1 ADRIMA's purpose and scope

ADRIMA is motivated by the severe challenges that organizations face with implementing EAM as separate and parallel initiatives. The purpose of the ADRIMA design theory is to give prescriptions that will support companies implementing EAM by addressing the requirements that EAM should be embedded in existing IT organizational structure and processes (DR4), that existing IT processes should continuously produce and maintain EA artifacts (DR1) as well as consume and use appropriate EA artifacts (DR2), and that existing IT roles and committees should understand EA artifacts and assume EAM tasks (DR3). Our empirical studies reveal that, if applied as an integral part of IT management, EAM implementation impacts IT organizational structure and processes that take decisions about how the EA will evolve or change according to the defined IT goals. Thus, the scope of the proposed design theory encompasses all IT processes and roles that directly or indirectly govern (i.e. direct, monitor) and manage (i.e. plan, develop, operate) the EA.

5.2 ADRIMA's constructs

The *constructs* represent the theory's entities of interest and their relationships in the sense of the "causa materialis" (Gregor and Jones 2007). Figure 2 illustrates the main constructs, i.e. the basic vocabulary (March and Smith 1995) that enables our design theory's communication and description in the form of a conceptual model.

Since our design theory embeds EAM in IT processes, the model depicts how general IT management and EAM constructs are interrelated. On the one hand, it represents IT processes, tasks/activities, IT artifacts, IT goals, as well as the stakeholders and roles, along with their relationships. On the other hand, the conceptual model illustrates that the IT management constructs are supported by EA artifacts and EAM practices. According to van der Raadt and van Vliet (2008), EA artifacts comprise EA documentation and EA policies. EA documentation (architecture description) describes a current (as-is) or future (to-be) EA model through different views. Therefore, EA documentation contains a certain degree of detail and formalization and addresses a defined set of concerns and viewpoints (Aier et al. 2009a; Buckl et al. 2008; Maier et al. 2004). Viewpoints capture the rules or provide the means for focusing, constructing, and analyzing particular aspects of architecture descriptions (Maier et al. 2004; Lankhorst et al. 2005). EA

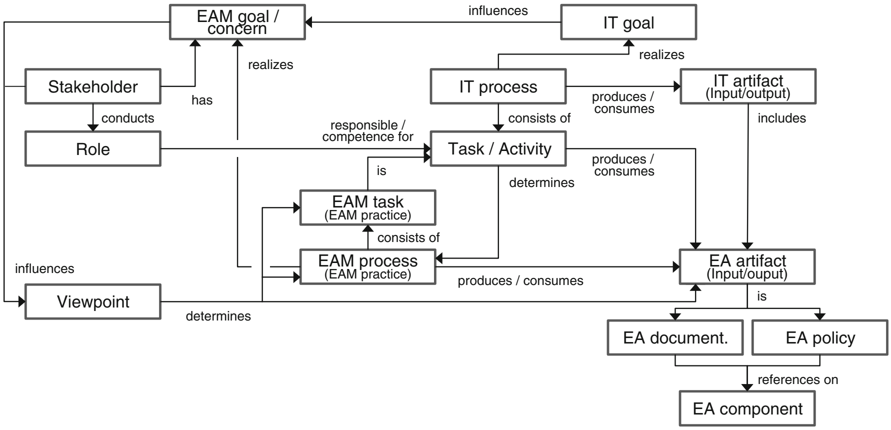


Fig. 2 ADRIMA's constructs in a conceptual model

policies are used to direct and control the EA development and typically refer to one or more EA components. Depending on their liability, they can be further divided into EA principles, guidelines, and standards (van der Raadt and van Vliet 2008). Moreover, an EA artifact can be part of an IT artifact, which is produced as an output or consumed as an input by an IT process. Since the literature uses different terminologies for EA/EAM-related activities, such as EA/EAM task, method, application, function, and process, we subsume them under the term EAM practice. Typical EAM practices (see Bucher et al. 2006; Buckl et al. 2010a; van der Raadt and van Vliet 2008; Schmidt and Buxmann 2011) include activities related to (1) EAM documentation (document, update or share an EA artifact); (2) EAM usage (analyze and evaluate EA artifacts or comply with an EA policy); (3) EAM governance (monitor, direct, and enact an EA policy or monitor, coordinate, and further develop EAM practices and EA artifacts or handle EAM escalation); as well as (4) EAM communication (communicate or provide feedback on EAM practices and EA artifacts). An EAM practice can comprise either a single EAM task or various EAM tasks, in order to realize a certain EAM goal or concern.

5.3 ADRIMA's principles of form and function and their testable propositions

> *DP1. Complement established IT processes with EAM practices and EA artifacts* Established IT processes that govern (i.e. direct and monitor) and manage (i.e. plan, develop, and run) substantial EA components and relationships should be complemented by EAM practices and EA artifacts.

The first design principle responds directly to DR4 and the need to generally embed EAM in the main IT processes that have a significant impact on the EA. Established IT processes must build the foundation on which to embed EAM practices and EA artifacts to better fulfill IT goals without reshaping the EAM as another management process cycle (e.g., The Open Group 2009; Buckl et al. 2009b). This implies that EAM practices and EA artifacts complement those IT processes that govern or manage substantial EA components, such as applications or infrastructure, and their relationships. We compare EAM, as a management function, with knowledge management (KM) (see Struck et al. 2010; Riempp 2004; Buckl et al. 2009a), since both EAM and KM need to have implicit and explicit knowledge effectively identified, documented, and communicated in enterprise-level management processes. Recent empirical EAM studies confirm this need in respect of EAM implementation in existing structures (e.g., Schmidt and Buxmann 2011). We suggest the following testable propositions:

(a) If the EAM practices and EA artifacts are embedded well in existing IT processes, EAM is more likely to succeed.
(b) If the EAM establishes parallel processes, it is more likely to fail.

> *DP2. Integrate EAM practices horizontally or vertically into specific IT tasks*
> (a) EAM usage and documentation practices that fit specific task characteristics should be horizontally embedded in an established IT process.
> (b) EAM communication and governance practices that fit specific task characteristics should be vertically embedded in an established IT process.

This principle aims to satisfy DR1, DR2 and DR4, and defines which kind of EAM practices to embed and where they should be embedded. Situational factors have been found to determine the application of EAM practices and act as preconditions to determine the most suitable EAM practices (Aier et al. 2009b) or what Buckl et al. (2011) refer to as method building blocks. These situational aspects in applying EAM practices can be explained by contingency theory which represents a dominant organizational and IS design approach, suggesting to adopt the appropriate level of a structural variable that fits the contingency to improve performance (Donaldson 2001; Weill and Olson 1989; Ginzberg 1980). Building on contingency theory and the work of Becerra-Fernandez and Sabherwal (2001) on knowledge management, we conclude that the impact of EAM is moderated by the context in which EAM is being used. From our field study, we have found existing IT tasks to be a determining situational factor. During our action research process, we learned that task characteristics determine appropriate EAM practices. Table 5 gives an overview and examples of the relationships between the IT task characteristic, EAM practice, and embedding direction. Regarding the embedding direction, an appropriate EAM practice should be either horizontally or vertically embedded in an existing IT process (see Riempp 2004):

- An EAM usage or documentation practice should be horizontally embedded, which means that it directly produces or consumes an EA artifact as a task of the existing IT process.
- An EAM communication or governance practice should be vertically embedded, which means a task of the existing IT process initiates the production or consumption of another EAM process's EA artifact.

Table 5 Task characteristic determining EAM practice embedding

IT task characteristic (example)	EAM practice	EAM practice (example)	Embedding
Analysis and evaluation of IT demands Decision-making on IT investments (e.g., project approval)	EAM usage	Analyze and evaluate impact on EA Check compliance with EA policy and EA targets	Horizontal
IT planning and goal setting Solution design (e.g., future CRM system)	EAM documentation	Develop and maintain, EA artifacts and components Modeling of EA artifacts	Horizontal
Publication of new IT standards and guidelines Productive start of new IT solution (e.g., new CRM system) Project marketing and communication	EAM communication	Provide feedback on EAM practices and EA artifacts Communicate EA artifacts	Vertical
Project review (e.g., quality assurance and compliance check) IT controlling	EAM governance	Monitor, direct, and enact EA policy Monitor and further develop EAM practices and EA artifacts Handle EAM escalation	Vertical

For DP2, we suggest subsequent testable propositions:

(a) If the applied EAM practices fit the IT tasks, they will increase the IT processes' efficiency and/or quality.
(b) If the EAM usage or documentation is not embedded horizontally in IT processes, they will be less effective.

DP3. Trigger EA maintenance continuously by means of an EA components' change Whenever an EA component changes, an IT process's feedback mechanism should be triggered to update the EA information base and secure its timeliness and quality.

This principle addresses DR1 and prescribes the conditions and mechanisms to ensure continuous maintenance of EA artifacts. A feedback mechanism should be triggered when an EA component changes to maintain updated EA documentation. The EA documentation has to be maintained, i.e. created or modified, validated and released, for example,

1. after a new IT planning round defines the to-be architecture;
2. after an IT governance process releases a new or modified EA policy;
3. when the as-is architecture is changed, i.e. when an EA component or relationship is created or modified.

Beside the organizational trigger and procedures, the feedback mechanism should also contain an interface specification for the consistent transfer from existing specialized repositories, such as a configuration management database, to the EA repository. This can, for example, be technically achieved by means of meta-model integration (see Fischer et al. 2007). In addition the organizational prerequisites, such as responsibilities, skills and procedures, have to be in place to link the specialized IT process's artifacts to the related EA artifacts. Our field studies reveal that the communication of a stakeholder's personal benefits from EA artifacts can boost continuous EA maintenance. For example, an EA model showing the impact and context of an IT project can be used as active marketing measures for a project manager. In respect of DP3, we propose the following testable propositions:

(a) If an IT process contains triggers for continuous maintenance, the EA documentation's up-to-dateness and quality will increase.
(b) If an IT process contains an interface specification to transform IT process-specific documentation into appropriate EA documentation, the EA documentation's up-to-dateness and quality will increase.
(c) The effort to ensure triggered, continuous EAM maintenance will be less than that required for various, single EAM initiatives to attain a similar EA information base quality.

> *DP4. Determine appropriate EA artifacts' granularity* An EA artifact should represent a defined set of stakeholders' viewpoints and satisfy the quality criteria of width, depth, and pragmatism.

This principle addresses both DR1 and DR2, and supports the application of EA artifacts in existing IT processes. As a fully covered EA information base is not feasible, appropriate EA artifacts should focus on major dependencies on a high level of abstraction and address a defined set of stakeholder viewpoints. An EA artifact forms an EA-specific in-/output that can be part of an IT artifact. Building on Aier et al. (2009a), three quality criteria should guide the definition and collection of appropriate EA artifacts:

- The width criterion specifies that an EA information base should only contain EA artifacts that are necessary to address stakeholders' viewpoints, which are defined by IT tasks and processes. Thereby, they also depend on task characteristics (see DP2).
- The depth criterion specifies that only holistic structures, i.e. those that reflect the entire organization or a group of similar components, are relevant. Detailed information is only relevant if an EA component's change has a significant impact on other components, has a significant influence on the behavior of the entire system or the details foster an EA component's reuse.
- According to the pragmatism criterion, the effort needed for continuous maintenance (see DP3) should be less than the benefits relating to the use of the EA artifacts. This implies that EAM usage and maintenance should use appropriate cost and benefit measures in order to make adjustments if required.

In respect of DP4, we suggest the following testable propositions:

(a) If EA artifacts fit the viewpoints, they will increase IT processes' quality.
(b) If EA artifacts fit the viewpoints, they will increase IT processes' degree of EAM usage.
(c) The EA information base will contain fewer unnecessary EA artifacts if the criteria of width, depth, and pragmatism are applied.

DP5. Enhance the established IT organization by EAM responsibilities and competencies Established roles and decision-making committees should be complemented by EAM-specific responsibilities and skills.

This design principle addresses DR3 and complements existing approaches focusing on the role of the architect and architecture boards (Keller 2007; Strano and Rehmani 2007; The Open Group 2009). However, our field study observations, as well as recent studies (Schmidt and Buxmann 2011; Winter and Schelp 2008), reveal that architecture topics have to be covered and become part of established IT roles and committees, such as project steering or IT budgeting committees. Consequently, architects have to serve on these committees to support decision-making. The procedures and rules of existing committees must be complemented by EAM feedback and escalation paths, as well as appropriate EAM decision-making rights and criteria, for example, to handle EA waiver requests or escalate to another committee (see van der Raadt and van Vliet 2008). Moreover, existing roles must be responsible for executing individual EAM tasks, such as updating EA documentation or escalating conflicts with EA standards, and must have the appropriate skills to do so. Consequently, the EA documentation's scope and contents to be handled, adherence to EA policies, as well as the EAM feedback and escalation paths should be part of the role profiles. In respect of DP5, we suggest the following testable propositions:

(a) If existing IT role profiles are complemented by EAM-specific responsibilities and skills, it is more likely that EA-related responsibilities will be carried out.
(b) If architects serve on existing IT committees, the enforcement of EA policies, i.e. standards and principles, will increase.

To conclude, Table 6 summarizes the design principles and determines how each of these addresses the design requirements and uses justificatory knowledge:

5.4 ADRIMA's expository instantiation

During the design cycle, we applied our design theory to a new IT demand management process at company B. The following section will illustrate how we applied the constructs and design principles in this expository instantiation "for the purpose of theory representation or exposition" (Gregor and Jones 2007). Figure 3 shows the first phases of the IT demand management process as BPMN diagram (OMG 2011). This diagram comprises the main constructs of our design theory, i.e. IT tasks, artifacts, and roles, as well as embedded EAM practices and EA artifacts.

Table 6 Mapping of design principles to design requirements and justificatory knowledge

Design principle	Design requirement	Justificatory knowledge
DP1. Complement established IT processes with EAM practices and EA artifacts	DR4	EAM compared to knowledge management (Struck et al. 2010; Riempp 2004; Buckl et al. 2009a)
DP2. Integrate EAM practices horizontally or vertically with specific IT tasks	DR4, DR1, DR2	Situational EAM (Buckl et al. 2011; Aier et al. 2009b) Contingency theory in general, and more specifically task characteristics as contingencies (Donaldson 2001; Becerra-Fernandez and Sabherwal 2001; Weill and Olson 1989; Ginzberg 1980)IT task characteristics as identified during field study
DP3. Trigger EA maintenance continuously by means of an EA components' change	DR1	Organizational procedures as identified during field study Meta-model integration for synchronizing repositories (Fischer et al. 2007)
DP4. Determine appropriate EA artifacts' granularity	DR2, DR1	Heuristics of an engineering-based EA approach (Aier et al. 2009a) Concern-driven EA modeling (Buckl et al. 2008; Lankhorst et al. 2005)
DP5. Enhance the established IT organization by EAM responsibilities and competencies	DR3	Enhancement of IT roles and committees by EAM responsibilities as identified during field study EAM governance structures (Schmidt and Buxmann 2011; Winter and Schelp 2008)

Regarding DP1, we selected the IT demand management process as an IT planning process that is highly relevant to EAM since it identifies, collects, evaluates, and prioritizes demands of future IT solutions. It thereby prepares architectural decisions and shapes the future states of the EA. The first design principle implies that this process is to be complemented by EAM practices and EA artifacts.

In order to apply DP2 and identify the relevant IT tasks, we analyzed how demands are processed in company B. After a requestor articulates a new demand (either in the planning process or ad-hoc), a dedicated person, the demand manager, ensures that the demand description is complete and analyzes its contents. This analysis is directly complemented (i.e. horizontal integration in DP2a) by an EA impact analysis, for example, to identify the affected business processes, organizational units, and applications (EAM usage).

To identify similar demands from other requestors, the demand manager categorizes the demand into an organization-wide domain model. In the exceptional case that domain categorization is not possible, the demand manager informs the central architecture team to maintain or enhance the domain model (see DP3). After domain model adaptions have been made, the central architecture team communicates the changes (i.e. vertical integration in DP2b) back to the organization. In

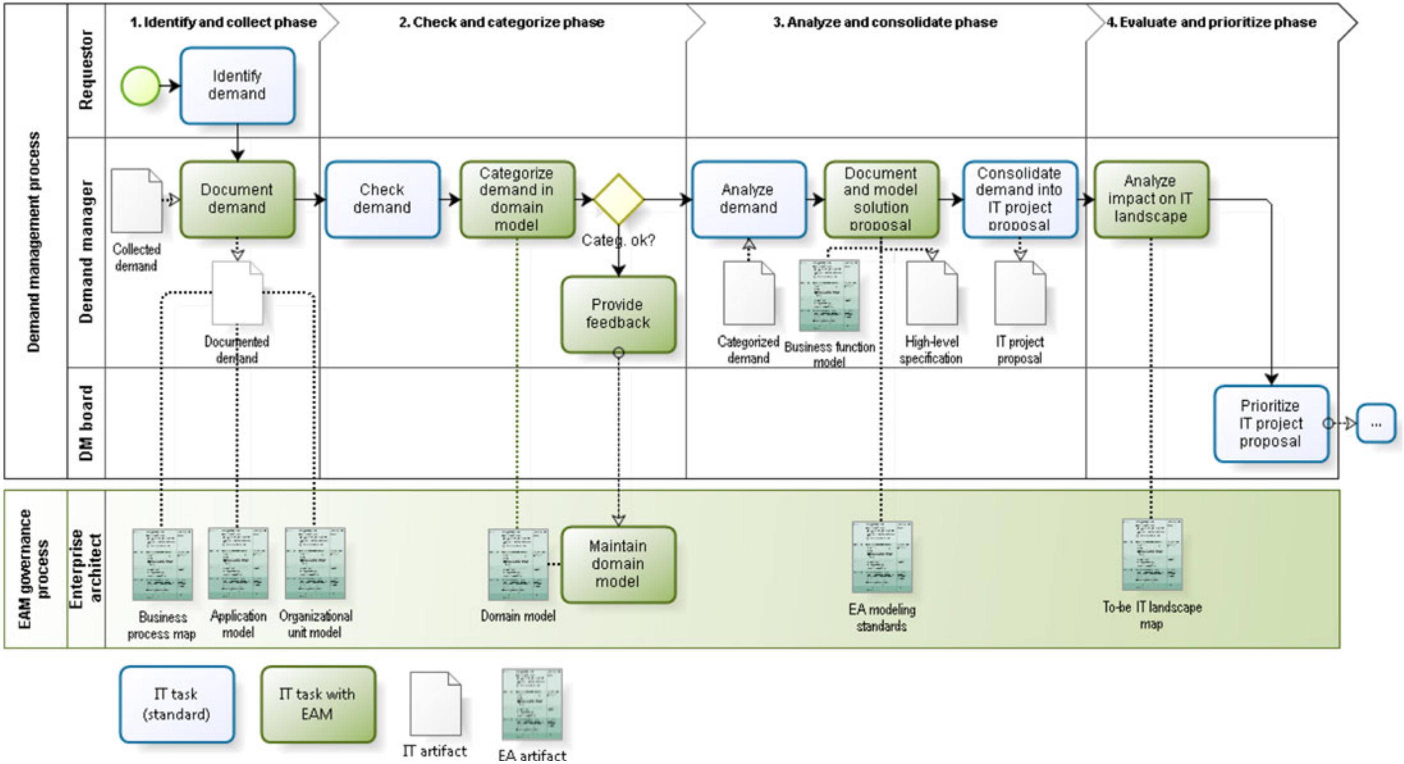


Fig. 3 Expository instantiation of demand management process (excerpt)

another example, the central architecture team defines and monitors (i.e. vertical integration in DP2b) EA standards to document demand's solution alternatives by means of a to-be business function model. When considering the go-live (i.e. the trigger in DP3) of the solution that implements the demand, which is not covered in the BPMN diagram, all changed EA artifacts must be updated. For example, the to-be business function model is transformed into an as-is business function model.

Regarding DP4, IT demand managers and requestors need well-arranged, schematic views to analyze the demand's problem and solution space. EA models, such as the to-be business function models, may be used to document a demand's proposed solution from a business perspective. It illustrates the new or changed business functions, as well as the affected business objects. Additionally, application landscape maps document a solution alternative's impact on the to-be IT landscape.

Lastly, DP5 is applied by establishing that an enterprise architect, as a member of the demand management committee, has a veto right. The enterprise architect participates during the demand evaluation and prioritization to avoid conflicts with the target EA and EA goals, for example, the global IT landscape harmonization goals. In doing so, not only the enterprise architect's role profile, but also the demand management committee's rules of internal procedure should reflect the veto right competency. This instantiation is not visible from the BPMN diagram, but in the organizational chart of the IT committees.

5.5 ADRIMA's principles of implementation

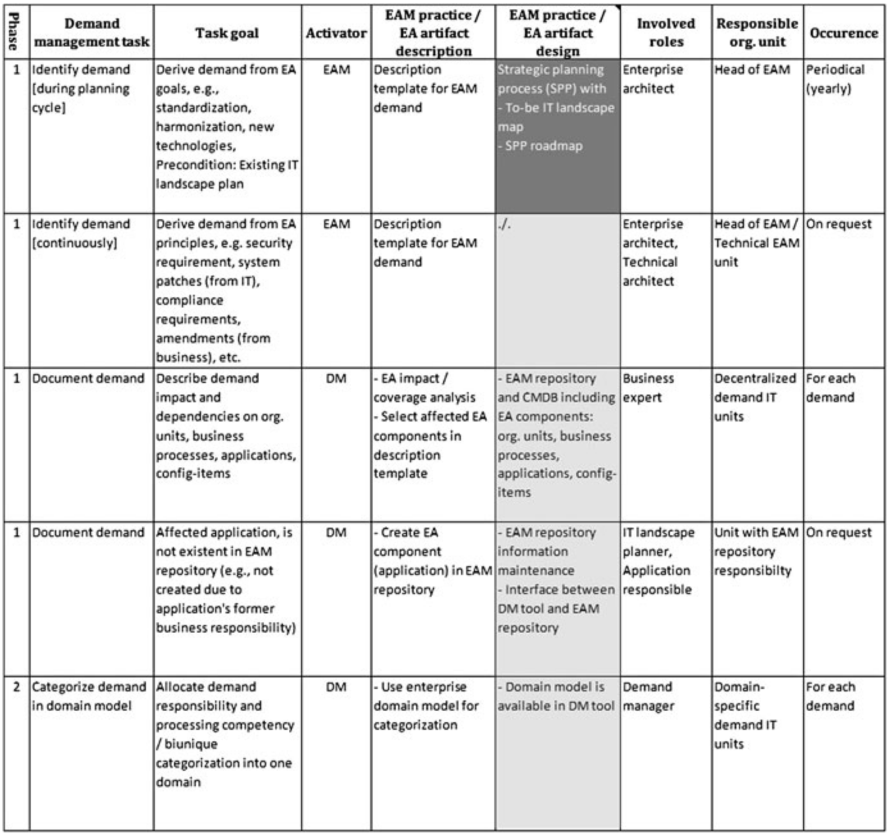
In this section, we will discuss issues that concern the means by which the design is brought into being, i.e. "the implementation process involving agents and actions" (Gregor and Jones 2007). We will focus on the preparation steps needed to embed EAM into a concrete instantiation of an established IT process. The main challenge in implementing our design theory is to ensure organizational change and a continuous EAM execution with the respective IT process. Basically, we suggest the following three implementation principles (IP), but do not claim for exhaustiveness:

IP1. Conceptual modeling involving IT managers and enterprise architects to define EAM-embedded IT processes: The first implementation principle follows the argumentation that the purposeful design of an EAM-embedded IT process is a critical first step in EAM implementation and concentrates on the means to do so. We suggest that appointed IT and EAM personnel work together to develop a conceptual model of the EAM-embedded IT processes. Changes in established IT processes and organizational structures can only occur if the responsible personnel (e.g., process or application owners) and the architects work together. Both parties' skills are required to discuss the IT's and architects' concerns.

In our field study, we applied a consensus-oriented conceptual modeling approach (Becker and Niehaves 2007) involving different IT managers and employees, as well as IT methodology experts, during the design cycle. Moreover, based on the BPMN diagrams, we conjointly discussed the EAM embedding potential of each IT task involving the IT demand management program manager, two enterprise architects, and a business analyst (see IP2). Consequently, we regard conceptual modeling of EAM-embedded IT processes, as generally described in business process management (e.g., Delen et al. 2005; Melao and Pidd 2000) and reference modeling (e.g., Winter and Schelp 2006; Fettke and Loos 2007), as a valuable first step in implementing our design theory. It provides useful means to define some kind of formal representation of the IT process and organizational structures. Thus, formal representations can provide the necessary guidance for organizational change and positive evaluation results increase their potential to be applied in the organization.

IP2. Analyze task characteristics, roles, and viewpoints to determine appropriate EAM practices and EA artifacts: Second, following DP2 to DP5, the established IT process needs to be analyzed in order to determine suitable EAM practices and EA artifacts. The detailed BPMN diagrams helped us discuss the task characteristics and analyze whether each step should be complemented by EAM practices or EA artifacts. A spreadsheet, as shown in Fig. 4, documents the decision to embed EAM.

For every task in the demand management process, the table documents the task's goals and activators (e.g., the EAM or the demand management team). The following two columns describe the EAM practices to be used. The color code



Phase	Demand management task	Task goal	Activator	EAM practice / EA artifact description	EAM practice / EA artifact design	Involved roles	Responsible org. unit	Occurence
1	Identify demand [during planning cycle]	Derive demand from EA goals, e.g., standardization, harmonization, new technologies, Precondition: Existing IT landscape plan	EAM	Description template for EAM demand	Strategic planning process (SPP) with - To-be IT landscape map - SPP roadmap	Enterprise architect	Head of EAM	Periodical (yearly)
1	Identify demand [continuously]	Derive demand from EA principles, e.g. security requirement, system patches (from IT), compliance requirements, amendments (from business), etc.	EAM	Description template for EAM demand	./.	Enterprise architect, Technical architect	Head of EAM / Technical EAM unit	On request
1	Document demand	Describe demand impact and dependencies on org. units, business processes, applications, config-items	DM	- EA impact / coverage analysis - Select affected EA components in description template	- EAM repository and CMDB including EA components: org. units, business processes, applications, config-items	Business expert	Decentralized demand IT units	For each demand
1	Document demand	Affected application, is not existent in EAM repository (e.g., not created due to application's former business responsibility)	DM	- Create EA component (application) in EAM repository	- EAM repository information maintenance - Interface between DM tool and EAM repository	IT landscape planner, Application responsible	Unit with EAM repository responsibilty	On request
2	Categorize demand in domain model	Allocate demand responsibility and processing competency / biunique categorization into one domain	DM	- Use enterprise domain model for categorization	- Domain model is available in DM tool	Demand manager	Domain-specific demand IT units	For each demand

Fig. 4 Reconcilement sheet (example)

determines whether an EAM practice is established and can be used (dark grey cells) or has to be developed/enhanced (light grey cells). In addition, during the discussion, we captured the responsible roles and organizational units, as well as the task frequency (periodically, on request, for each demand).

IP3. Adapt or design EAM practices and EA artifacts: Based on IP2's analysis, the enterprise architect is able to shape appropriate EAM practices and EA artifacts that complement the IT process and satisfy the relevant information requirements. As ADRIMA introduces only design principles to EAM practices and EA artifacts, they require operationalizations—for example by means of EAM patterns (see Buckl et al. 2008). When an adequate EAM practice and EA artifact was already available, its design, i.e. its viewpoints and application, was reviewed with the IT process owner. Moreover, the role profiles of the person in charge were redefined so that he or she was capable of applying the destined EAM practice or EA artifact. If there was no adequate existing EAM practice or EA artifact, a new one had to be designed and developed (e.g., by triggering the vertically embedded EAM governance process).

5.6 ADRIMA's artifact mutability

While ADRIMA's constructs and design principles provide a general and abstract blueprint for EAM implementation in established processes and organizational structures, the corresponding instantiation in a concrete organization is dependent on mutual factors. With artifact mutability, we aim at anticipating those factors and address their degree of change in the state of the artifact and the expository instantiation. We identified the following four factors: organizational setting, process type, degree of formalization of the process description, and organizational change.

First, we anticipate different EAM practices for different organizational settings, for instance depending on the size of the organization and its management structures. For example, EAM communication practices may differ between small and large-scale organizations. Whereas, in small settings, an application owner may take over EAM communication tasks, such as publishing the application's context and scope in a repository, in large-scale settings, a dedicated team, such as a program or project management office, may conduct dedicated EAM communication tasks as part of project marketing.

Second, for different process types, i.e. IT planning, development, and operations processes, we anticipate differences between EA artifacts' granularity. For example, we anticipate that the EA documentation to be used in the IT project portfolio management process will have a higher granularity, covering the whole organization, whereas IT projects will capture the detailed solution design using EA documentation.

Third, we anticipate that the embedding of appropriate EA artifacts will differ for those processes, whose process description can be formalized in advance, compared to emergent or ad hoc processes, whose process description cannot be formalized. The determination of the width, depth, and pragmatism criterion will be vaguer for the latter because it is not possible to describe an adequate formal representation for them in advance.

Fourth, due to organizational changes over time, the instantiated operationalization of EAM practices and EA artifacts will certainly evolve, which might require further adaptions without circumventing ADRIMA's design principles.

6 Conclusion and implications

6.1 Contribution

This paper's main scientific contribution is a design theory for ADRIMA which is summarized in Table 7. The ADRIMA design theory synthesizes prescriptive statements to embed EAM into existing IT processes; it is based on insights from three field studies, prior research and practical experiences in EAM projects. Its most important contribution consists of five design principles for integrating EAM practices and EA artifacts in the IT organization and processes that plan, change, and manage the EA. ADRIMA addresses the practical challenges that emerge with EAM implementation and provide means for organizational change by prescribing how IT organizational structures and processes, which incorporate the constructs and design principles, can be constructed and implemented in specific circumstances. Our research goes beyond previous scientific and practical approaches in its focus on design principles and emphasis on the EAM embedded in existing IT management; the existing EAM literature and EA frameworks describe EAM as a stand-alone management concept focusing on EA models and the EA life cycle. Conversely, existing IT management approaches, such as ITIL or COBIT, do not (yet) or only partly, consider EAM. As a design theory, ADRIMA complements and integrates recent EA research that deals with situational EAM introduction, EAM patterns and EAM application scenarios (Buckl et al. 2009b, 2010a, 2011; Aier et al. 2009b). It synthesizes the most critical design knowledge underlying EAM implementations and provides theoretical as well as evidence-based justification to inform and explain the design decisions. The formulation of our research findings as a design theory provides some advantages compared to the construction of methods, reference models or patterns; it allows for coping with mutable design artifacts and deriving different instantiations for specific organizational settings' processes.

6.2 Limitations

Our research has certain limitations: Since our focus was on developing a design theory for ADRIMA, our present research approach mirrors an IT-driven EAM perspective. This IT-led EAM focus is common not only in the literature in which the discipline evolved from system architectures (see Zachman 1987) into a holistic enterprise management function (see Ross et al. 2006; Kappelman et al. 2009), but also in the majority of EAM initiatives that start in the IT department. However, we consider EAM a holistic organizational approach and anticipate that the design theory could be extended to embedding EAM in other management structures and

Table 7 Anatomy of the ADRIMA design theory

Design theory component	Consideration in ADRIMA design theory
1. Purpose and scope	The main purpose of our research is to propose a design theory that will support companies implementing EAM by addressing the requirements (DR4) that EAM should be embedded in existing organizational structure and processes, (DR1) that existing IT processes should continuously produce and maintain EA artifacts as well as (DR2) consume and use appropriate EA artifacts, and (DR3) that existing IT roles and committees should understand EA artifacts as well as assume skills and responsibilities for EAM tasks
2. Constructs	ADRIMA's constructs describe how IT management and EAM are interrelated. They comprise IT-related constructs (IT processes, tasks, stakeholders and roles) as well as EAM-related constructs (EAM goal/concern, viewpoint, EAM practice, EA artifact, EA component)
3. Principles of form and function	The design theory defines five design principles for complementing established IT processes with EAM practices and EA artifacts (DP1), for vertically and horizontally embedding EAM (DP2), for triggered continuous EA maintenance (DP3), for appropriate EA artifact granularity (DP4), as well as for EAM-complemented IT roles and committees (DP5)
4. Artifact mutability	EA artifacts and EAM practices to be embedded are highly dependent on the concrete organizational setting, the specific IT processes and the degree of their formalization. In addition, their embedding evolves over time, which requires continuous adaption of the design theory's instantiation
5. Testable propositions	Propositions are formulated to test whether the implementation of ADRIMA's design principles result in better EAM implementation compared to alternative approaches, such as model-driven EAM initiatives
6. Justificatory knowledge	The design principles are built on general IS theories, such as contingency theory, the EAM-specific knowledge base ("practitioner-in-use" theories) as well as evidence-based justification from our field study
7. Principles of implementation	Three implementation principles are suggested, that build on conceptual modeling and the subsequent refinement of EA artifacts and practices in a collaborative approach involving EAM and IT stakeholders
8. Expository instantiation	An instantiation example of the IT demand management process is presented, which explicates the different design principles

processes. We encourage future research to extend ADRIMA to the broader enterprise management perspective.

A further limitation is that we developed our design theory in collaborative research projects with two organizations, and our findings may be constrained by this research setting. While design-oriented action research is a recommended research approach for the iterative development of design theory results (see Siponen et al. 2006; Walls et al. 1992), such studies do not prove their universal validity. We addressed this issue in our research approach, by conducting iterative design cycles with periodic review and validation workshops covering different organizational settings and EAM implementation contexts. In order to ensure the validity and generalizability of our design theory, we identified the relevant justificatory knowledge, including general and "practitioner-in-use" theories as well as evidence-based justification, that gives a basis and explanation for design decisions. To further increase the validity and generalizability of our research findings, especially in respect of other organizational outlets, future research should build on our findings to test and enhance the design principles and their testable propositions by means of multiple case studies or large-scale quantitative studies. Moreover, the design theory has so far only been instantiated in one organization. Investigating instantiations in other companies and processes will refine and complement our theory.

6.3 Implications for practice and research

Our work's implication for practice is that organizations should not introduce EAM as a separate initiative, but should aim at changing their established IT processes to become EAM-aware. In doing so, organizations can more transparently communicate the value of EAM use and avoid stakeholder resistance. Our design theory may guide practitioners in their efforts to implement EAM in their organization. By applying the design principles, they can systematically embed EAM practices and EA artifacts into their existing IT organizational structures and processes.

As an implication for research, our work draws the attention to the integration of EAM practices and EA artifacts with established organizational practices and artifacts. Keeping in line with Gregor and Jones (2007), we suggest that future research should address the further refinement of our design theory's components, notably the design theory's artifact mutability and expository instantiation, for example in the context of organizational change (see Whelan-Berry and Somerville 2010; Pettigrew et al. 2001; Visscher and Visscher-Voerman 2010). Instantiating the design theory might also identify advanced EAM patterns or complement existing design-oriented research on situational EAM approaches.

References

Aier S, Riege C, Winter R (2008a) Classification of Enterprise Architecture - An Exploratory Analysis. Enterp Model Inf Syst Archit 3(1):14–23

Aier S, Riege C, Winter R (2008b) Unternehmensarchitektur—Literaturüberblick und Stand der Praxis. Wirtschaftsinformatik 50(4):292–304

Aier S, Kurpjuweit S, Saat J, Winter R (2009a) Enterprise architecture design as an engineering discipline. AIS Trans Enterp Syst 1(1):36–43

Aier S, Riege C, Schönherr M, Bub U (2009b) Situative Methodenkonstruktion für die Projektbewertung aus Unternehmensarchitekturperspektive. Paper presented at the 9th International Conference Wirtschaftsinformatik, Vienna

Bacharach SB (1989) Organizational theories: some criteria for evaluation. Acad Manag Rev 14(4):496–515

Baskerville R (2008) What design science is not. Eur J Inf Syst 17(5):441–443

Baskerville R, Pries-Heje J (2010) Explanatory design theory. Bus Inf Syst Eng 2(5):271–282

Baskerville R, Wood-Harper T (1998) Diversity in information systems action research methods. Eur J Inf Syst 7(2):90–107

Baskerville R, Pries-Heje J, Venable J (2009) Soft design science methodology. Paper presented at the 4th international conference on design science research in information systems and technology (DESRIST 2009), Philadelphia

Becerra-Fernandez I, Sabherwal R (2001) Organizational knowledge management: a contingency perspective. J Manag Inf Syst 18(1):23–55

Becker J, Niehaves B (2007) Epistemological perspectives on IS research: a framework for analysing and systemizing epistemological assumptions. Inf Syst J 17(2):197–214

Boddy D, Boonstra A, Kennedy G (2005) Managing information systems: an organisational perspective, 2nd edn. Financial Times Prentice Hall, Harlow

Boh WF, Yellin D (2007) Using enterprise architecture standards in managing information technology. J Manag Inf Syst 23(3):163–207

Booth ME, Philip G (2005) Information systems management: role of planning, alignment and leadership. Behav Inf Technol 24(5):391–404

Bricknall R, Darrell G, Nilsson H, Pessi K (2006) Enterprise architecture: critical factors affecting modelling and management. Paper presented at the 14th European conference on information systems, Göteborg

Bucher T, Fischer R, Kurpjuweit S, Winter R (2006) Enterprise architecture analysis and application—an exploratory study. Paper presented at the 10th IEEE international enterprise distributed object computing conference workshops (EDOCW 2006), Hong Kong

Buckl S (2011) Developing organization-specific enterprise architecture management functions using a method base. Dissertation, sebis TU München

Buckl S, Ernst AM, Lankes J, Matthes F (2008) Enterprise architecture management pattern catalog (Version 1.0, February 2008). Research report, sebis TU München, http://wwwmatthes.in.tum.de/wikis/sebis/eampc. Accessed 9 Aug 2010

Buckl S, Matthes F, Schweda CM (2009a) Future research topics in enterprise architecture management—a knowledge management perspective. Paper presented at the 4th workshop trends in enterprise architecture research (TEAR 2009), Stockholm

Buckl S, Matthes F, Schweda CM (2009b) A Viable system perspective on enterprise architecture management. Paper presented at the IEEE international conference on systems, man and cybernetics (SMC 2009), San Antonio

Buckl S, Matthes F, Schweda CM (2010a) Towards a method framework for enterprise architecture management—a literature analysis from a viable system perspective. Paper presented at the 5th international workshop on business/IT alignment and interoperability (BUSITAL 2010), Hammamet

Buckl S, Schweda CM, Matthes F (2010b) A design theory nexus for situational enterprise architecture management. Paper presented at the 14th IEEE international EDOC conference (EDOC 2010), Vitoria

Buckl S, Matthes F, Schweda CM (2011) A method base for EA management. Paper presented at the IFIP working conference on method engineering (ME 2011), Paris

Bussells SE (2006) Assessment of a government agency's enterprise architecture program. J Enterp Archit 2(1):43–50

Cabinet Office (2011) ITIL Service Design. 2 edn. TSO (The Stationery Office), London

Correia A, Brito e Abreu F (2009) Integrating IT service management within the enterprise architecture. Paper presented at the fourth international conference on software engineering advances (ICSEA 2009), Porto

CIO Council (2001) A practical guide to federal enterprise architecture. http://www.gao.gov/bestpractices/bpeaguide.pdf. Accessed 09 Aug 2010

Delen D, Dalal NP, Benjamin PC (2005) Integrated modeling: the key to holistic understanding of the enterprise. Commun ACM 48(4):107–112

Dern G (2007) Management von IT-Architekturen: Informationssysteme im Fokus von Architekturplanung und -entwicklung. Vieweg, Wiesbaden

Donaldson L (2001) The Contingency Theory of Organizations. Sage Publications, Thousand Oaks

Dubin R (1976) Theory building in applied areas. In: Dunette M (ed) Handbook of Industrial and Organizational Psychology. Handbook of Industrial and Organizational Psychology. Rand McNally, pp 17–8

Earl MJ (1988) Information management: the strategic dimension. Clarendon Press, Oxford

FEA Program Management Office (2007a) FEA Consolidated Reference Model Version 2.3. http://www.whitehouse.gov/sites/default/files/omb/assets/fea_docs/FEA_CRM_v23_Final_Oct_2007_Revised.pdf. Accessed 14 Aug 2010

FEA Program Management Office (2007b) FEA Practice Guidance. http://www.cio.gov/Documents/FEA_Practice_Guidance_Nov_2007.pdf. Accessed 14 Aug 2010

Ferstl OK, Sinz EJ (1997) Modelling of business systems using the semantic object model (SOM)—a methological framework. Bamberger Beiträge zur Wirtschaftinformatik 43

Fettke P, Loos P (2007) Perspectives on reference modeling. In: Fettke P, Loos P (eds) Reference modeling for business systems analysis. Idea Group Publishing, Hershey, pp 1–21

Findlay D (2006) Implementing enterprise architecture in the federal aviation administration air traffic organization. J Enterp Archit 2(2):29–33

Fischer F, Matthes F, Wittenburg A (2005) Improving IT management at the BMW group by integrating existing IT management processes. Paper presented at the 9th IEEE international EDOC conference (EDOC 2005), Twente

Fischer R, Aier S, Winter R (2007) A federated approach to enterprise architecture model maintenance. Enterp Model Inf Syst Archit 2(2):14–22

Fischer C, Winter R, Wortmann F (2010) Design theory. Bus Inf Syst Eng 2(6):387–390

Frank U (1995) MEMO: A tool supported methodology for analyzing and (Re-) designing business information systems. In: Ege R, Singh M, Meyer B (eds) Technology of Object-Oriented Languages and Systems

Ginzberg MJ (1980) An organizational contingencies view of accounting and information systems implementation. Acc Organ Soc 5(4):369–382

Greefhorst D, Proper E (2011) Architecture principles - the cornerstones of enterprise architecture. Enterprise engineering series. Springer, Berlin

Gregor S (2009) Building theory in the sciences of the artificial. Paper presented at the 4th international conference on design science research in information systems and technology (DESRIST 2009), Philadelphia

Gregor S, Jones D (2007) The anatomy of a design theory. J Assoc Inf Syst 8(5):313–335

Hafner M, Winter R (2008) Processes for enterprise application architecture management. Paper presented at the 41st annual Hawaii international conference on system sciences (HICCS 2008), Waikoloa, 07 Jan 2008

Hevner AR (2007) A three cycle view of design science research. Scand J Inf Syst 19(2):Article 4

Hevner AR, Chatterjee S (2010) Design research in information systems, vol 22., Integrated series in information systemsSpringer, New York

Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. MIS Q 28(1):75–105

Hjort-Madsen K, Pries-Heje J (2009) Enterprise architecture in government: fad or future? Paper presented at the 42th Hawaii international conference on system sciences (HICSS 2009), Waikoloa

Hoogervorst JAP (2009) Enterprise governance and enterprise engineering. Springer, Berlin

IEEE (1990) IEEE Std 610.12-1990 standard glossary of software engineering terminology. IEEE Computer Society, New York

Iivari J (2007) A paradigmatic analysis of information systems as a design science. Scand J Inf Syst 19(2):39–64

ISACA (2011) COBIT mapping: overview of international IT guidance, 3rd edn. http://www.isaca.org/Knowledge-Center/Research/ResearchDeliverables/Pages/COBIT-Mapping-Overview-of-International-IT-Guidance-3rd-Edition.aspx. Accessed 23 May 2012

ITGI (2007) COBIT 4.1: Framework, Control Objectives, Management Guidelines, Maturity Models. IT Governance Institute. http://praktische-informatik.fh-luebeck.de/sites/default/files/CobiT_4.1.pdf. Accessed 22 June 2012

Janssen M, Hjort-Madsen K (2007) Analyzing enterprise architecture in national governments: the Cases of Denmark and the Netherlands. Paper presented at the 40th annual Hawaii international conference on system sciences, 2007 (HICSS 2007), Waikoloa

Kappelman LA, McGinnis T, Pettit A, Salmans B, Sidorova A (2009) Enterprise architecture: charting the territory for academic research. In: Kappelman LA (ed) The SIM guide to enterprise architecture. CRC Press, Boca Raton, pp 96–107

Keller W (2007) IT-Unternehmensarchitektur. dpunkt.verlag, Heidelberg

Klein HK, Myers MD (1999) A set of principles for conducting and evaluating interpretive field studies in information systems. MIS Q 23(1):67–93

Kuechler B, Vaishnavi V (2008) On theory development in design science research: anatomy of a research project. Eur J Inf Syst 17(5):489–504

Lankhorst MM, Iacob M-E, Jonkers H, van der Torre L, Proper HA, Arbab F, de Boer FS, Bonsangue M, Hoppenbrouwers S, Veldhuijzen van Zanten GE, Groenewegen L, van Buuren R, Campschroer J, Steen MWA, Stam AW, Wieringa RJ, van Eck PAT, Krukkert D, Ter Doest HHWL, van Leeuwen

D, Fennema P, Jacob J, Bosma H, Cuvelier MJ, Penders PGM, Bekius SF, Janssen M (2005) Enterprise architecture at work. Springer, Berlin

Lincoln YS, Guba EG (1985) Naturalistic inquiry. Sage Publications, Newbury Park

Löhe J, Legner C (2012) From enterprise modeling to architecture-driven IT management—a design theory, Paper presented at the 20th European conference on information systems (ECIS 2012), Barcelona

Luftman J, Ben-Zvi T (2011) Key issues for it executives 2011: cautious optimism in uncertain economic times. MIS Q Executive 10(4):203–212

Maier MW, Emery D, Hilliard R (2004) ANSI/IEEE 1471 and systems engineering. Syst Eng 7(3):257–270

March ST, Smith GF (1995) Design and natural science research on information technology. Decis Support Syst 15(4):251–266

March ST, Storey VC (2008) Design science in the information systems discipline: an introduction to the special issue on design science research. MIS Q 32(4):725–730

Matthee M, Tobin P, van der Merwe P (2007) The status quo of enterprise architecture implementation in South African financial services companies. S Afr J Bus Manag 38(1):11–24

Mayring P (2000) Qualitative Content Analysis. Forum: Qualitative Social Research 1 (2):Art. 20

Melao N, Pidd M (2000) A conceptual framework for understanding business processes and business process modelling. Inf Syst J 10(2):105–129

Mingers J (2001) Combining IS research methods: towards a pluralist methodology. Inf Syst Res 12(3):240–259

Morganwalp JM, Sage AP (2004) Enterprise architecture measures of effectiveness. Int J Technol Policy Manage 4(1):81–94

Moser C, Junginger S, Brückmann M, Schöne KM (2009) Some process patterns for enterprise architecture management. Paper presented at the Fachtagung des GI-Fachbereichs Softwaretechnik, Kaiserslautern

Niemann KD (2005) Von der Unternehmensarchitektur zur IT-Governance. Vieweg, Wiesbaden

Niemi E (2007) Enterprise architecture stakeholders—a holistic view. Paper presented at the Americas conference on information systems (AMCIS 2007), Keystone

OMG (2011) Business Process Model and Notation (BPMN) Version 2.0. http://www.omg.org/spec/BPMN/2.0/PDF. Accessed 22 Sept 2011

Österle H (1995) Business in the information age - heading for new processes. Springer, Berlin

Österle H, Brenner W, Hilbers K (1993) Total information systems management: a European approach. Wiley, Chichester

Pettigrew AM, Woodman RW, Cameron KS (2001) Studying organizational change and development: challenges for future research. Acad Manag J 44(4):697–713

Riempp G (2004) Integrierte Wissensmanagement-Systeme - Architektur und praktische Anwendung. Springer, Berlin

Riempp G, Gieffers-Ankel S (2007) Application portfolio management—a decision-oriented view of enterprise architecture. Int J Inf Syst e-Bus Manag 5(4):359–378

Riempp G, Müller B, Ahlemann F (2008) Towards a framework to structure and assess strategic IT/IS Management. Paper presented at the 16th European conference on information systems, Galway

Roeleven S, Broer J (2009) Why two thirds of enterprise architecture projects fail: An explanation for the limited success of architecture project. IDS Scheer. http://its.a-cci.com/doc/EA_-_Roeleven_Broer_-_Enterprise_Architecture_Projects_Fail_-_AEP_en.pdf. Accessed 15 Sept 2011

Ross JW (2003) Creating a strategic it architecture competency: learning in stages. MIS Q Executive 2(1):31–43

Ross JW, Weill P, Robertson D (2006) Enterprise architecture as strategy. Harvard Business School Press, Boston

Sauer C, Willcocks LP (2002) The evolution of the organizational architect. MIT Sloan Manag Rev 43(3):41–49

Scheer A-W (1991) Architektur integrierter Informationssysteme - Grundlagen der Unternehmensmodellierung. Springer, Berlin

Schelp J, Winter R (2009) Language communities in enterprise architecture research. Paper presented at the 4th international conference on design science research in information systems and technology (DESRIST 2009), New York

Schmidt C, Buxmann P (2011) Outcomes and success factors of enterprise IT architecture management: empirical insight from the international financial services industry. Eur J Inf Syst 18(1):168–185

Schönherr M (2009) Towards a common terminology in the discipline of enterprise architecture. In: Service-Oriented Computing—ICSOC 2008 Workshops. pp 400–413

Seppanen V, Heikkila J, Liimatainen K (2009) Key issues in EA-implementation: case study of two Finnish government agencies. Paper presented at the 11th IEEE conference on commerce and enterprise computing, Vienna

Siponen M, Baskerville R, Heikka J (2006) A design theory for secure information systems design methods. J Assoc Inf Syst 7(11):725–770

Smith ML, Erwin J (2007) Role & responsibility charting (RACI). http://myclass.peelschools.org/sec/12/4268/Resources/RACI_R_Web3_1.pdf. Accessed 22 March 2012

Software Engineering Institute (2010) CMMI for Development, Version 1.3. Carnegie Mellon Software Engineering Institute (SEI). http://www.sei.cmu.edu/reports/10tr033.pdf. Accessed 11 March 2011

Sowa JF, Zachman JA (1992) Extending and formalizing the framework for information systems architecture. IBM Syst J 31(3):590–616

Spewak S, Tiemann M (2006) Updating the enterprise architecture planning model. J Enterp Archit 2(2):11–19

Strano C, Rehmani Q (2007) The role of the enterprise architect. IseB 5(4):379–396

Struck V, Buckl S, Matthes F, Schweda CM (2010) Enterprise architecture management from a knowledge management perspective—results from an Empirical Study. Paper presented at the 5th mediterranean conference on information systems (MCIS)

The Open Group (2009) TOGAF Version 9—The open group architecture framework. The Open Group, Zaltbommel

van Aken JE (2004) Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules. J Manage Stud 41(2):219–246

van Bon J, Verheijen T (2006) Frameworks for IT management: an introduction (ITSM library). Van Haren Publishing, Zaltbommel

van der Raadt B, van Vliet H (2008) Designing the enterprise architecture function. Paper presented at the 4th international conference on the quality of software architectures (QoSA2008), Karlsruhe

van Schaik EA (1985) A management system for the information business: organizational analysis. Prentice-Hall, Engelwood Cliffs

Venkatesh V, Bala H, Venkatraman S, Bates J (2007) Enterprise architecture maturity: the story of the veterans health administration. MIS Q Executive 6(2):79–90

Visscher K, Visscher-Voerman JIA (2010) Organizational design approaches in management consulting. Manag Decis 48(5):713–731

Walls JG, Widmeyer GR, El Sawy OA (1992) Building an information system design theory for Vigilant EIS. Inf Syst Res 3(1):36–59

Wegmann A, Regev G, Rychkova I, Le L-S, Cruz JDdl, Julia P (2007) Business and IT alignment with SEAM for enterprise architecture. Paper presented at the 11th IEEE international EDOC conference (EDOC 2007), Annapolis

Weill P, Olson MH (1989) An assessment of the contingeny theory of MIS. J Manag Inf Syst 6(1):59–85

Whelan-Berry KS, Somerville KA (2010) Linking change drivers and the organizational change process: a review and synthesis. J Chang Manag 10(2):175–193

Winter R (2004) Architektur braucht Management. Wirtschaftsinformatik 46(4):317–319

Winter R (2008) Design science research in Europe. Eur J Inf Syst 17:470–475

Winter R, Fischer R (2007) Essential layers, artifacts, and dependencies of enterprise architecture. J Enterp Archit 3(2):7–18

Winter R, Schelp J (2006) Reference modeling and method construction: a design science perspective. Paper presented at the the 2006 ACM symposium on applied computing (SAC 2006), Dijon

Winter R, Schelp J (2008) Enterprise architecture governance: the need for a business-to-IT approach. Paper presented at the 2008 ACM symposium on applied computing (SAC 2008), Fortaleza

Wittenburg A, Matthes F, Fischer F, Hallermeister T (2007) Building an integrated IT governance platform at the BMW Group. Int J Bus Process Integr Manag 2(4):327–337

Zachman JA (1987) A framework for information systems architecture. IBM Syst J 26(3):276–292

Zarnekow R, Brenner W, Pilgram U (2006) Integrated information management: applying successful industrial concepts in IT. Springer, Berlin

Zink G (2009) How to restart an enterprise architecture program after initial failure. J Enterp Archit 5(2):31–41