Building Blocks for Enterprise Architecture Management Solutions

Sabine Buckl, Thomas Dierl, Florian Matthes, and Christian M. Schweda

Technische Universität München, Institute for Informatics 19,
Boltzmannstr. 3, 85748 Garching, Germany
{sabine.buckl,dierl,matthes,schweda}@in.tum.de
http://www.systemcartography.info

Abstract. Enterprise architecture (EA) management has become a commonly accepted means to guide enterprises in transformations responding to their ever changing environment. Organizations seeking to establish an integrated and effective EA management function are typically faced with a challenging lack of standardization in the field. Although the topic is heavily researched by practitioners, researchers, standardization bodies, and tool vendors, no commonly accepted understanding of the scope, reach, and focus of EA management exists. This fact can be explained by the distinct organizational structures, contexts, cultures, and requirements, which are specific for each enterprise and therefore ask for an enterprise-specific realization of the EA management function.

In response to the aforementioned challenge this article presents *building blocks for EA management solutions* (BEAMS). BEAMS on the one hand provides practical guidance for organizations to support the design and development of an organization-specific EA management function by presenting method and language building blocks, which can be selected and configured based on the specificities of the organization under consideration, i.e. the organizational context and the goals pursued. On the other hand BEAMS gives hints for researchers willing to contribute to the discipline of EA management. The theoretic discussion on the developing BEAMS approach is complemented by an example to illustrate the applicability of the approach. Finally, a critical reflection of the achieved results is given and future areas of research are discussed.

Keywords: EA management function, building blocks, patterns, situational method engineering, design theory nexus.

1 Introduction

In a rapidly changing economic, technical, and regulatory environment, the flexibility to adapt to changes as well as the ability to implement new business capabilities quickly are both vitally important for companies regardless of their type and size. Emerging paradigms as service oriented architectures (SOA), domain-specific languages or model driven development claim to be helpful in this context, but when it comes to their implementation in an organization, subtle difficulties arise. This can be exemplified with the implementation of an SOA, but holds for other paradigms as well: restructuring the IT landscape of an organization towards services is a long-lasting endeavor, needing not only quite a few information on the applications, but also on the connected business processes and business objects. Most likely such information is not present at the beginning of an SOA transformation program, but has to be gathered in an extensive process. Even if the information is available, the transformation program cannot assume that 'the world keeps from turning', i.e., the organization does not stop changing. Therefore, the transformation program has to be continuously realigned with the change and maintenance projects that are executed in parallel. In this respect, realizing all benefits of an SOA transformation is only possible in an environment, where business and IT *development* are aligned.

This mutual alignment goes beyond a mere *provider role* of the IT, in which IT resorts itself to solely fulfilling business requirements. IT in contrast has to take an *enabler role*, proactively seeking to increase flexibility and adaptability to foster the agility of the overall organization. This two-fold role of the IT well illustrates the very core of business-IT alignment (cf. [23,35,52]), which could have also been described conversely from a business perspective. In consequence, mutual alignment is a goal best to be approached from both perspectives – a business and an IT perspective – and is hence not in the focus of the management functions for business or IT management, respectively. This calls for a management function with an embracing management subject spanning business- and IT-related concepts, but most preferably also accounting for *crosscutting aspects*, as strategies and projects. The latter is especially necessary as a managed evolution of the organization inevitably connects to the strategies as drivers of organizational change and the projects as its vehicles. This holistic understanding of the organization actually is the one incorporated in enterprise architecture (EA), i.e. the architecture of the enterprise which in accordance with the ISO Standard 42010 [26] can be defined as follows:

> Enterprise architecture (EA) is the fundamental conception of the enterprise in its environment, embodied in its elements, their relationships to each other and to its environment, and the principles guiding its design and evolution.

The management of the EA forms a management discipline that seeks to address the aforementioned topic of mutual alignment by taking the embracing perspective of the overall EA. This new management discipline has – not surprisingly – attracted practitioners and researchers seeking for guidance on how to conduct and perform EA management. Research in this area is typically conducted in close cooperation and interaction with an organization willing to practically apply the research results. On the one hand this opens the door for "developing case studies" (cf. van Aken in [49, page 232]) by employing an intrinsically motivated industry partner. On the other hand, industry-funded research projects usually underly the partnering organization's pace and hence often force early delivery of results, which aggravates the development of comprehensive theoretical underpinnings. Thus, researchers in the area of EA management are challenged to ensure that their research conducted in close interaction with organizations does not degenerate into "routine design" that according to Hevner et al. in [24, page 82] must be distinguished from design science. In contrast, the close cooperation can be used to contribute to theory building e.g. via extracting case studies (cf. van Aken in [49] as well as Eisenhardt and Graebner in [15]). Building on a figure from Gehlert et al. in [20, page 442] that illustrates the twofold relation between theory and design according to Nunamaker et al. in [40], we highlight the interplay between design and theory building (cf. Figure 1).

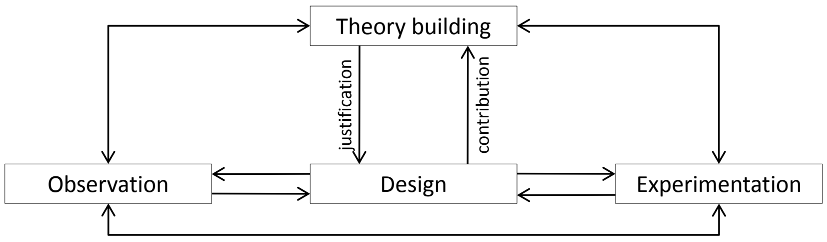

Fig. 1. The interplay of design and theory building [20,40]

In the light of this interplay and against the backing environment in which EA management research is typically carried out, the first research question guiding our subsequent considerations can be derived.

> *How can researchers on the one hand contribute to the knowledge base of EA management, and at the same time deliver early results applicable in practice?*

An interesting area for contributing to the knowledge base of EA management is concerned with the structure of the EA management function itself, as currently, no commonly accepted step-by-step guidelines for performing EA management exist. This absence might be caused by the fact, that no EA management process model detailing the management function has yet gained prominence. Some researchers even doubt the existence of a *one-size-fits-them-all* approach, but expect the management function to be organization-specific (cf. [5,31,50,48]). This situation is similar to the one in software development, where albeit a general agreement on important activities as e.g. *requirements elicitation* or *testing*, various process models exist, which strongly differ concerning the linkages between the different activities and the level of detail in which the different activities are described[1]. The situation of EA management is even more complicated than the one in software development. The goals of a software development process are typically agreed upon as "developing a software system in time, with the required functionality and quality, as well as within the planned budget" [47]. The objectives of an EA management initiative in contrast vary widely. While typical EA management goals can be summarized on an abstract level, they have to be substantiated during the establishment of an appropriate organization-specific

[1] For a in-depth discussion of different software development process models see [37].

management function in order to identify the elements of the EA relevant for the initiative. Reducing maintenance costs via standardization can for instance be performed on different levels, e.g. on business processes, business support provided by business applications, or on a more technical infrastructure level.

Besides the variety of different goals, which need to be appropriately addressed by the EA management function, the organizational context, in which the function has to be embedded and operated, influences the suitability of an EA management approach. While in a smaller company with a familiar atmosphere, the simple communication of architectural principles might be sufficient to ensure project compliance, a more hierarchical corporate culture might demand for the establishment of quality gates, e.g. architecture reviews prior to the project start as well as controls after realization of the project to ensure adherence to architectural principles and standards. Some of the existing approaches even stress the fact that they have to be adapted to the context of the applying organization (cf. "adapting the ADM [architecture development method]" in [48, page 56 seq]) but typically abstain from providing information on how to perform these adaptations. A better situation can be identified regarding the goals pursued by the different approaches, which are typically detailed on an abstract level as mentioned before. This leads us to the second research question of this article

How does a configurable approach to design EA management functions look alike?

In this article, we answer the aforementioned research questions by presenting *building blocks for EA management solutions* (BEAMS). The BEAMS approach is based on a conceptualization of Pries-Heje and Baskerville, who introduced the concept of a design theory nexus in [41], the prefabrics of the pattern-based approach to EA management presented by Buckl et al. in [6] and Ernst in [17], and the situational method engineering as discussed by Harmsen in [22]. The resulting approach is presented in Section 3 and complemented with a constructed case providing an example of how BEAMS can be applied (cf. Section 4). With BEAMS being a relatively new approach, currently a comprehensive practical evaluation has yet not been conducted. To provide indications on the suitability and applicability of the approach, a comparison with prominent approaches representing the state-of-the-art of EA management practice is given in Section 5. Final Section 6 provides a critical reflection of the achieved results and hints to further areas of research.

2 Prefabrics from Related Disciplines

Developing an EA management function, which suits the specific needs of the using organization is a challenging task. Taking into account the rich literature on the subject EA management as presented by Langenberg and Wegman in [32] as well as Schönherr in [45], the development activity can be understood as task in which different competing solutions offered by existing EA management approaches are compared and evaluated in respect to their suitability. The evaluation a) is thereby based on the goals pursued by the EA management initiative and b) accounts for the organizational context to embed the EA management function into. Two different approaches to perform such evaluation and to facilitate the aforementioned challenge are discussed below. First, the construct of a design theory nexus (DTN), which provides "a set of constructs and methods that enable the construction of models that connect numerous design theories with alternative solutions" [41, page 733], is introduced. Secondly, the idea of situational method engineering is presented, which describes how a method can be "tailored and tuned to a particular situation" (cf. Harmsen in [22, page 25]).

2.1 A Design Theory Nexus for Competing Solutions

In [41], Pries-Heje and Baskerville present the idea of a DTN as means to connect existing competing solutions, i.e. design theories, for a problem domain. A DTN is no simple framework connecting these solutions, but further helps "decision makers in choosing which of the theories are most suitable for their particular goals and their particular setting" [41, page 733], i.e. the organizational environment as well as the goals to pursue. Pries-Heje and Baskerville discuss that their approach is useful in cases of solving *wicked problems*. Establishing an EA management function forms a wicked problem, for which a plethora of competing solutions exist. A DTN instantiation for EA management can be developed, which provides assistance in choosing a suitable EA management approach according to the organizations' goals and organizational context. According to Pries-Heje and Baskerville in [41, page 743], a DTN instantiation consists of the following four constructs depicted in Figure 2:

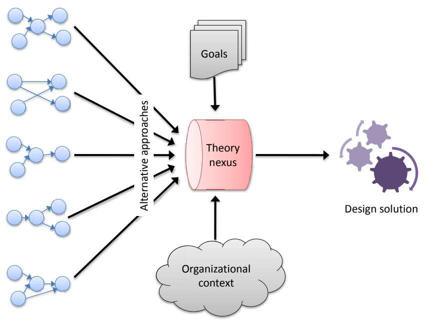

Fig. 2. Components of a DTN according to Pries-Heje and Baskerville in [41]

– **Goals** describe what the design solution is intended for.
– **Organizational context** refers to contingencies outside of the people involved.
– **Design theory nexus** defines the connection point at which the competing theories are bound with realities into a design solution.
– **Design solution** represents the result constructed from dissimilar decision alternatives.

The instantiation, i.e. construction, of a DTN according to Pries-Heje and Baskerville (cf. [41]) follows a five step approach. In the first step, the available approaches in the area under consideration are examined, e.g. via a literature analysis. In a second step, the identified competing theories are investigated for explicit or implicit conditions that must hold for the approach to achieve the highest utility. Here, it has to be noted that these conditions might not match, i.e. be *asymetric*, for any pairing of the theories. The third step assesses the identified conditions for practical relevance and formulates them to assertions. In the fourth step, a decision-making process for evaluating the assertions is undertaken. Final step five combines the approaches, conditions, assertions, and the process into a tool, which supports the evaluation of the fit for each approach in a given situation.

2.2 Situational Method Engineering

Motivated by the plurality of proposed methods for information system engineering as well as the increasing application area diversification and complexity, Harmsen presented in [22] an approach to *situational method engineering.* The driving idea behind situational method engineering can be summarized by the following quote: “There is no method that fits all situations” [22, page 6]. Introducing the term *controlled flexibility* Harmsen elicits requirements for a method engineering approach, which accomplishes method standardization and at the same time is flexible enough to match the situation at hand. A *situation* thereby refers to the combination of circumstances at a given point in time in a given organization [22]. In order to address these requirements, for each situation a suitable method – so-called situational method – is *constructed* that takes into account the circumstances applicable in the respective situation. In the construction process uniform method fragments are selected, which can be configured and adapted with the help of formally defined guidelines.

The generic process to constructing situational methods consists of four steps. Input to the configuration process is the specific situation in which the method should be applied, e.g. the environment of the initiative, including users, organizational culture, management commitment, etc. This situation is analyzed in the first step (*characterization of the situation*) to describe the application characteristics. This information is used in the second step (*selection of method fragments*) to select suitable method fragments from the method base. Heuristics can thereby be applied to foster the selection process. In the third step (*method assembly*) the method fragments corresponding with the situation characterization are combined to a situational method. During assembling method fragments, aspects like completeness, consistency, efficiency, soundness, and applicability are accounted for (cf. [22]). The actual use of the constructed situational method is performed in the last step (*project performance*). Figure 3 gives an overview on the construction process and the relationships between the different steps.

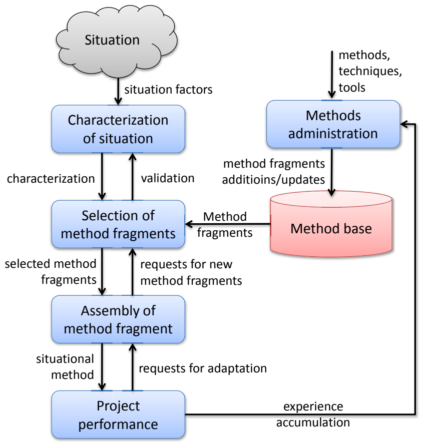
Situation
situation factors
Characterization of situation
characterization
validation
Selection of method fragments
selected method fragments
requests for new method fragments
Assembly of method fragment
situational method
requests for adaptation
Project performance
experience accumulation
methods, techniques, tools
Methods administration
method fragments additioins/updates
Method base
Method fragments

Fig. 3. The process of situational method engineering according to [22]

Complementing the construction process of a method situated for a given environment, Harmsen introduces in [22] the activity *methods administration* that captures methodical knowledge, i.e. adds or updates method fragments, if necessary, based on feedback from the project performance step.

Developing a harmonization in the area of method engineering and at the same time emphasizing on the influence of the particular situation a method should be applied in, represents the core idea in situational method engineering as presented by Harmsen in [22]. The state of IS engineering described by Harmsen is quite similar to the one in developing and designing an organization-specific EA management function. A multitude of approaches exists but none of these has gained prominence due to the situation- or organization-specificity of the subject. Therefore, we propose an approach that picks up the idea of the DTN presented by Pries-Heje and Baskerville in [41] and the approach of situational method engineering presented by Harmsen in [22].

2.3 A Pattern-Based Approach to EA Management

Patterns have a long history as useful means for documenting re-usable solutions for recurring problems in a complex domain, dating back to Alexander [3], who introduced patterns as "coherent and modular solutions to specific problems". Further publications (cf. Buschmann et al. [12] or Gamma et al. [19]) have refined the term pattern and put forward structuring guidelines for the description of patterns. A broadly accepted structure is presented by Buschmann et al. in [12], according to which a pattern is constituted of the following elements:

– **Context description**, which is concerned with causes and environmental factors that may have lead to the problem that the pattern solves.
– **Problem description**, which alludes to the issues and difficulties that occur in many contexts and may be solved with the pattern. Thereby, the description expatiates on conflicting forces that comprise the problem.

- **Solution description**, which explains the steps to be taken and the concepts to be used in order to solve the corresponding problem.
- **Consequence description**, which refers to consequences that may be caused by applying the pattern to the given problem.

Building on the idea of patterns, Buckl et al. [6] coined the term of the "EA management pattern" as a way to structure the domain of EA management. In [16], Ernst further develops this idea towards a pattern language for EA management. This pattern language introduces three types of patterns, namely *methodology pattern (M-Pattern)*[2], *viewpoint pattern (V-Pattern)*, and *information model pattern (I-Pattern)* that are used to develop an organization-specific EA management function. These three types of *EA management patterns* describe constituents of proven-practice solutions for EA management as found in literature but also in practice (cf. [13]). The different types of patterns contribute different parts to an EA management function, detailed as follows:

- **M-Patterns** describe management methods (and processes) that solve a specific EA-related problem. Thereby, a pattern provides step-by-step guidance and information on what and how to do.
- **V-Patterns** describe viewpoints, i.e., types of visualizations that are employed by an M-Pattern in order to communicate solution-relevant information about the EA.
- **I-Patterns** describe conceptual models, whose concepts are instantiated to documentations of solution-relevant parts of the overall EA.

Buckl et al. describe in [6] how the three types of EA management pattern can together be used to design an organization- and problem-specific EA management function. The context descriptions provided by the patterns are explored during this phase in order to select the appropriate patterns that optimally fit the organizational context. The problem descriptions are the starting points for selecting the "right" EAM pattern, i.e. those patterns that solve the organization-specific problems. If different patterns were applicable to similar problems and hence were to be decided upon during the design process, the context and consequence description could provide additional help to choose the patterns that optimally balance desired outcomes and side-effects. Finally, the interrelationships between the patterns, which are described as part of the pattern language, support the identification of patterns that might also apply in the given context or may be helpful for solving related problems. After having selected the appropriate patterns, the methods, viewpoints, and conceptual models described therein have to be integrated into a management function. This final design step requires method engineering capabilities, as especially the M-pattern as described in [13] do not provide integration artifacts. The same is true for the V- and I-Pattern, which have to be integrated into a comprehensive EA modeling language. While some issues on integration are discussed by Buckl et al. in [6], dedicated integration artifacts and mechanisms are not provided.

[2] Being more clear with respect to the terminology, these patterns should be alluded to as "method patterns".

The absence of integration related prescriptions may be explained with the focus and the nature of the approach. Patterns are solutions observed in practice, i.e. describe real-world solutions, and are not engineered or developed towards an integrated knowledge base. This can be exemplified with the M-patterns that describe methods and processes for conducting EA management, but are not concerned with designing an EA management function. A design method for EA management function would have to provide additional guidance for

– **selecting** the appropriate EA management patterns, especially in case different of them are applicable,
– **integrating M-patterns** into a consistent EA management process, especially avoiding process redundancies, and
– **integrating V- and I-patterns** into an EA modeling language, especially accounting for the information demands of the viewpoints.

The subsequently presented approach refines the EA management patterns described in [13] to address the aforementioned issues of integration. The patterns are reorganized and rewritten to redundancy free and composeable building blocks for EA management solutions.

3 BEAMS – Components and Design

The BEAMS approach presents a DTN instantiation for situational EA management based on the groundworks presented in Section 2. In Section 3.1 the structure and interplay of the components of BEAMS is described, reflecting the core dichotomy of EA management – method and language – as manifested in the EAM pattern approach (cf. Section 2.3) and discussed by Schelp and Winter in [44]. Based on the common understanding of the constituents, we discuss the construction process of BEAMS in Section 3.2 utilizing the five-step method as proposed by Pries-Heje and Baskerville in [41].

Prior to presenting BEAMS and its components, a central design principle of the approach should be introduced, namely the principle of "loose coupling" between the building-blocks. By this term, we describe a characteristic of the inner organization of BEAMS. As opposed to a pattern-language the building-blocks are not interlinked by *explicit*, i.e. *material*, relationships. The actual relationships are of *implicit* nature, i.e. constitute *formal* relationships, that may be derived from the building-blocks relations to the underlying framework and stratified terminology. Put in other words, the BEAMS' underlying framework supplies an ontology, on which the building-blocks are built in a way that their (formal) interrelations may be derived from linkages to the same framework constituents. As a consequence, this design principle helps to develop BEAMS (cf. Section 3.2) from manifold sources without having to intermesh the different design theories and prescriptions in a dense web of newly established relationships.

3.1 Components of BEAMS

The idea of interrelating competing solutions to design an organization-specific EA management function can only be realized against the basis of a common understanding and terminology. Multiple publications have targeted this topic. Schönherr showed in [45] that the discipline of EA management is not yet developed a fully consistent terminology, but is on the way to do so. In a similar vein, Schelp and Winter discuss in [44] that different "language communities" in EA management research exist, although certain degree of convergence has recently been reached. Nevertheless, when it comes to distinct aspects of EA management multiple approaches agree on a common understanding regardless of some terminological differences. One of these aspects is the question of the fundamental activities and tasks of EA management. Buckl et al. revisit in [11] the different perspectives on this topic as put forward in literature taking into account the pattern approach (cf. [13]) as well as the approaches of Frank [18], Wegmann [51], Hafner and Winter [21], Niemann [38], Schekkerman [43] and The Open Group [48]. Based on the literature, Buckl et al. devise a method framework for EA management consisting of four activities as shown in Figure 4:

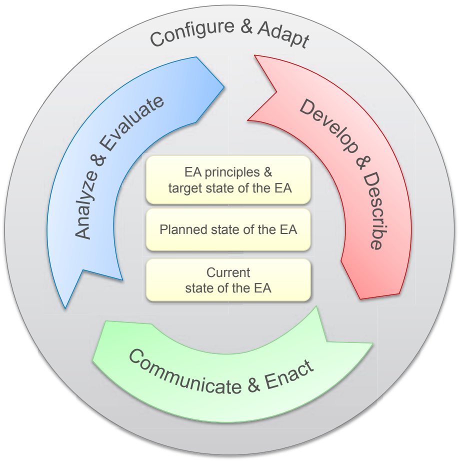

Fig. 4. Method framework of BEAMS

Develop & describe a state of the EA, either a current state describing the as-is architecture, a planned state or a target state, i.e. an EA vision.

Communicate & enact architecture states and principles to EA-relevant projects and to related management functions, as project portfolio management.

Analyze & evaluate architectural scenarios (planned states) or analyze whether a planned state helps to achieve the target state or not.

Configure & adapt the EA management function itself, i.e. decide on the management concerns, goals, and methods.

Against the background of the method framework reverberating through related EA management literature, we can devise the core structure of BEAMS in context as shown in Figure 5. The core constituents of BEAMS as a DTN instantiation for the field of EA management design are:

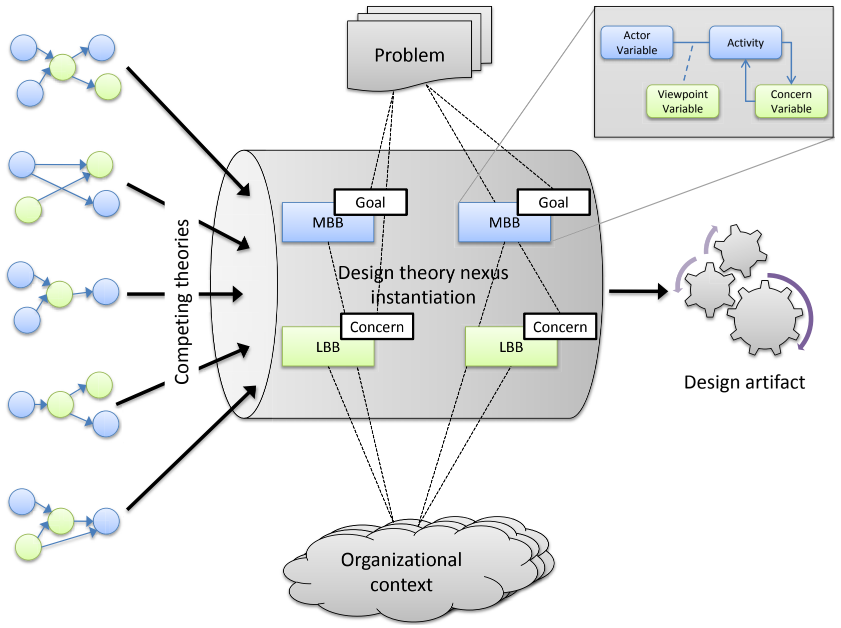

Fig. 5. Components of BEAMS

Competing theories: The competing theories represent the knowledge base from which BEAMS is built. Reflecting the nature of EA management as a practice-oriented field of research, BEAMS builds on solutions with have been proven valuable in practice, e.g. patterns, best practices, case studies.

Problem: A problem represents the issue to be solved by applying the theory. A problem in the area of EA management typically consists of a

- **goal** representing an abstract objective, e.g. increase homogeneity, provide transparency, and a
- **concern,** i.e. area of interest in the enterprise, e.g. business support, application systems.

Organizational context: The organizational context represents the situation in which the EA management function operates. Typical factors which are considered in the organizational context are the organizational culture, management commitment, stakeholders, etc.

Building block (BB): The building blocks form the solution models to be combined to an organization-specific EA management function. Reflecting the dichotomy of method and language, two kinds of building blocks exist,

- **method building block (MBB)** describing who has to perform which tasks in order to address a problem in the situated context and
- **language building block (LBB)** referring to which EA-related information is necessary to perform the tasks and how it can be visualized.

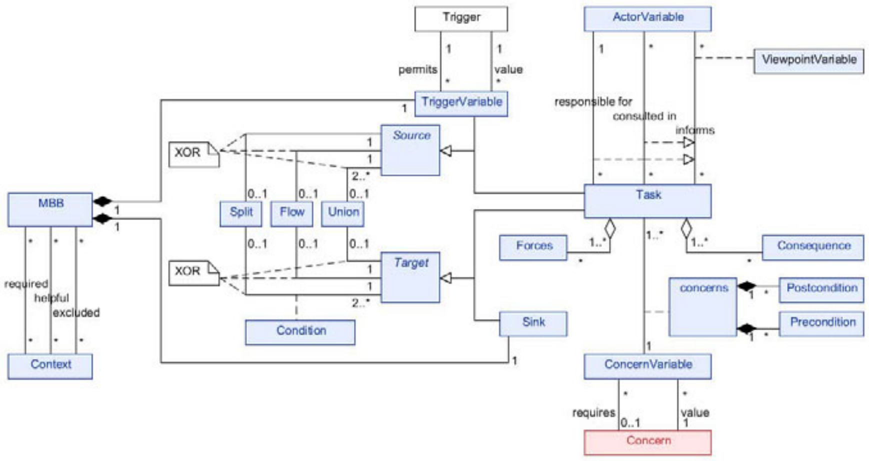

Fig. 6. Meta model of the method bulding blocks of the BEAMS approach

BEAMS actually distinguishes two subtypes of LBBs, namely information model building blocks IBBs and viewpoint building blocks VBBs. With the focus of this paper being on the MBBs, we abstain from giving in-depth information on these two types of building blocks and direct the interested reader both to Section 4, where examples of such building blocks are provided as well as to [7], where Buckl et al. discuss VBBs in more detail. Shortly summarizing the roles of these two types of building blocks, we may say that IBBs are used to define the *syntax* and *semantics* of the EA description language, i.e. to reflect the corresponding concern of the EA management function. VBBs are used to describe the language's *notation*[3], i.e. the way the EA-related information is presented.

Central to BEAMS is the notion of the MBB as re-usable solution for building an organization-specific EA management process mirroring the three phases of *develop and describe*, *communicate and enact* as well as *analyze and evaluate*. In this vein, we start with explaining the nature and inner organization of an MBB further taking into account the relationships to the other constituents of BEAMS. Figure 6 displays the constituents of an MBB and the relationship between these constituents. An MBB describes the different TASKs that are performed in order to achieve a certain goal under a given organizational CONTEXT. The MBB further specifies the ordering of the tasks and specifies SPLITs and UNIONs designating where tasks are alternative in their execution. For every SPLIT the MBB also describes the CONDITIONs that act during task execution. Complementing each MBB is started with a trigger represented in a TRIGGER VARIABLE. In configuring the EA management function this variable is filled with an actual TRIGGER, obeying the rules for doing so as supplied via the PERMITS relationship. Each task is executed by a corresponding actor represented by an

[3] The term "notation" is used in accordance with Kühn [30], whereas other publications refer to the notation as "concrete syntax".

ACTOR VARIABLE in the description of the method. The interplay of TASKs and FLOWs is described in a BPMN-like notation (cf. [10]) as exemplified in Figure 9. The notion of the "actor variable" is employed here to denote that the description of the MBB does not specify distinct actor or role in the using organization, but merely describes a responsibility of a person or group. Further, the MBB can specify that the actor variable is bound in respect to its organizational role, e.g. might express that an escalation based enactment mechanism only works, if a superordinate actor can be called upon. Beside to the mandatory relationship to the executing, i.e. *responsible* actor variable, each task may relate to other actor variables as well, namely variables representing actors that are *consulted* or *informed* during task execution. The distinction between the different levels of involvement pertaining to a single task is based on the RACI model of CobiT (see e.g. [27]), while a slightly different perspective is taken on the involvement level *informed*. For the purpose of describing MBBs, we assume that any actor involved in a task is informed, such that the responsible actor as well as consulted actors are counted as informed, too.

The *informed* relationship between a TASK and a corresponding actor (as represented in an ACTOR VARIABLE) is reified via a VIEWPOINT VARIABLE designating that the actor takes a specific viewpoint on the information relevant during the given task. The notion of the variable here again describes that the MBB does not make concrete prescriptions on the viewpoint to be used, but in turn allows to select an organization-specific viewpoint for accomplishing the task. Two remarks have to be added with respect to the VIEWPOINT VARIABLE. In the context of the BEAMS approach the concept of the viewpoint is discussed from a strongly notational perspective. This means that a viewpoint completely commits to the *syntax* and *semantics* specified by the underlying concern (IBB), while only the *notation* is specified in the viewpoint definition. This understanding of viewpoint is grounded in the work of Buckl et al. [8], who have discussed the relationships between viewpoints and concerns on a more formal basis. The notation of a viewpoint is specified via VBBs as transformation over the corresponding syntax via a pipe-based transformation language. Figure 7 shows an exemplary VBB defining a clustered visualization, where certain information objects describing the EA are converted to symbols (parameterization OUTER). Starting from these information objects the clustered visualization traverses a relationship (parameterization OUTER2INNER) to related information objects that are further converted to symbol (parameterization INNER). When exemplifying the approach in Section 4, we shall see more VBB-based transformations. A second remark pertains to the statement that an MBB does not make concrete prescriptions on the viewpoints to be used. While this is actually the case, an MBB may indeed recommend or discourage certain viewpoints, respectively. For example, a viewpoint variable used during an interview-related task may recommend textual viewpoints while on the contrary discouraging the utilization of viewpoints in the 'lines and boxes'-style. These statements nevertheless are no prescriptions on the level of the actual viewpoint to be used but rather on a meta-level, recommending or discouraging certain 'types' of viewpoints.

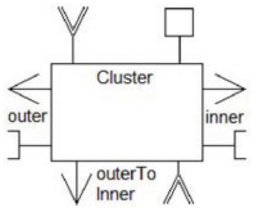

Fig. 7. VBB describing a clustered visualization

From an exterior perspective an MBB is associated with the architectural concern that its tasks cover. The concern specifies the area-of-interest in the enterprise on which the different steps taken in the MBB act. In line with the argumentation of Buckl et al. in [8] a concern may be identified with an information model, such that an MBB may during instantiation into a concrete process in the EA management function be parameterized with an according model. Nevertheless, many MBBs as e.g. ones associated with the activity of *develop and describe* can specify tasks without further knowledge on the actually associated area-of-interest. In this sense, an MBB does not directly link to a specific concern but to CONCERN VARIABLEs that are assigned with concrete concerns during configuration. Speaking more precisely, any MBB links to a CONCERN VARIABLE, specifying the MBB's or tasks *precondition* and *postcondition* concerning information demands. An exemplary precondition supplied as part of a concern variable would state that the MBB can only be executed, if information conforming to the given concern was documented. The same concern might conversely state as postcondition that the documented information was also cleared for communication. Any MBB may hence specify its dedicated set of pre- and postconditions, although these conditions must be specified using termini from the BEAMS terminology (or an extended version thereof). Finally, it has to be remarked that a CONCERN VARIABLE may specify a *lower bound*. By doing so the variable states a concern that is bound as value to the variable has to at least incorporate the concepts specified by the lower bound concern. Exemplifying this one should think of a task concerned with the assessment of projects. While no assumptions have to be made in respect to the exact area-of-interest during the assessment, it is nevertheless necessary for reasons of consistency to demand that the concern at least covers the project concept.

3.2 Development of BEAMS

The development of BEAMS demands due the DTN nature of the artifact input from related approaches. The very first approach concerned with EA management is the Zachman framework [53] dating back to 1987. Since that time, the number of researchers and practitioners targeting this area of interest has increased [32]. An overview on the current state-of-the-art in EA management is given by Aier et al. in [2] and the most active research groups in the area are determined by Schelp and Winter in [44]. We utilize the thereby identified 'major players' and their approaches to designing an EA management function in step one of the construction of BEAMS as input for the competing theories. Accordingly, the approaches of the following groups form the basis of our subsequent elaborations:

– EPFL Lausanne, Switzerland
– Novay, The Netherlands
– University of St. Gallen, Switzerland
– TU Berlin, Germany
– KTH Stockholm, Sweden
– TU Munich, Germany
– TU of Lisbon, Portugal

Step two involves analyzing the competing approaches identified in the first step following the method of hermeneutic text comprehension (cf. [54]) in order to determine their distinguishing characteristics. Thereby, we in particular focus on the essential goals of each approach and the respective means, i.e. processes, to achieve these goals. In this way, we identified the following goals:

1. reduce operating cost
2. increase disaster tolerance
3. reduce security breaches
4. ensure compliance
5. increase homogeneity
6. improve project execution
7. enhance strategic agility
8. improve capability provision
9. foster innovation
10. increase management satisfaction

Complementing, we identified different means to establish an organization-specific EA management function, e.g. an engineering based approach as presented by Aier et al. in [1], a pattern-based approach presented by Buckl et al. in [6,13], or an analysis-focused approach introduced by Johnson and Eksted in [28]. These different approaches or the contained methods represent the input for BEAMS. In the following the development of BEAMS, i.e. the instantiation of a DTN, is exemplified alongside the EAMPC of TU Munich that provides a catalog of best practices gathered from industry and academia and therefore can itself be regarded as a collection of competing design theories.

Following the idea of patterns as e.g. introduced by Alexander et al. in [4] different types of relationships between pattern may exists (cf. Noble in [39]). While patterns can provide alternative solutions, meaning they cannot be used in combination, i.e. represent competing solutions, other relationship types like *compatible, sub-, super-*, or *intersected* refer to patterns, which can be used in combination. Considering the patterns as contained in the EAMPC the different types of relationships as introduced above exists, especially within one type of patterns. These relationships should be considered in the construction of the

DTN for situational EA management, but are according to the design principle outlined above to be converted to formal relationships as far as possible. In the third step, we derive a number of assertions that are based on prominent characteristics of each approach as expressed in literature. For the patterns presented by Ernst in [16][4], for example, we formulated inter alia the following assumptions:

- Detailed information on applications and standardized technology needs to be available.
- A centralized IT organization is required to enable an architecture review process.
- Upper management support needs to be available to ensure architecture conformance of projects.

The assumptions formulated for the competing approaches are gathered and reformulated in order to use a common terminology. The following non-exhaustive list provides an overview on the thereby identified assumptions, which represent the organizational context descriptions of BEAMS:

- Centralized vs. decentralized IT organization
- Upper Management support for the EA management team
- EA management team has own budget, e.g. for architectural relevant project
- A dedicated tool for EA management is available or not
- Integration with other management function and processes, e.g. project portfolio management, is defined

The above identified goals of EA management and the organizational contexts are formulated in forms of conditions and mapped to the assumptions of the identified solutions. The suitability of the competing solutions for any combination of the conditions can then be defined utilizing a *fitting matrix* with the competing solutions on the y-axis, the identified conditions on the x-axis, and a scoring of the fitting function in the cell. The fitting function can thereby take a value form the set *required, excludes, helpful.* The patterns for enhancing standard conformity as proposed by Ernst in [16], for instance, would require a centralized IT organization, while the upper management support would only be helpful but is not necessarily required.

Based on this fitting matrix, a decision-making process for selecting one or more appropriate solutions for designing an situational EA management function is developed in step four. The appropriateness of the EA management function is heavily influenced by the goals pursued by the organization as well as by whatever pertinent issues are presented in the organizational context. Therefore, these constraints, i.e. goals and organizational context, determine whether a competing EA management approach succeeds or fails.

Finally, a technique supporting the utilization of BEAMS is developed in final step five. Thereby, the competing approaches, goals, organizational contexts,

[4] The patterns presented by Ernst in [16] represent an excerpt, which is also contained in the EAMPC [13].

a well as the process, which applies the fitting matrix, are reflected in the design of the technique. Possible realizations of the techniques may range from simple excel-based techniques in line with the scoring matrix of Pries-Heje and Baskerville (cf. [41]) to more sophisticated tools, which cannot only be used for selecting an appropriate EA management approach. Based on BEAMS, an organization-specific EA management functions can be constructed following the construction process of situational method engineering. Therefore, the following five steps have to be performed by the using organization.

– **Characterize situation:** The organizational context descriptions as introduced above have to be assessed and the goals of the EA management initiative have to be defined.
– **Tool-based assessment of method fragments:** A preselection and evaluation of the competing design theories contained in the DTN is returned by the tool, based on the provided information.
– **Selection of design theories:** The enterprise architect has to choose between the remaining theories or decide to use a combination.
– **Assembly of design theories:** The selected design theories need to be configured and adapted, e.g. regarding the ordering or the used terminology.
– **Establish situational EA management function:** The designed function has to be established, e.g. regarding governance structures, quality gates, etc.

Following the idea of *method administration* as discussed by Harmsen in [22], a performance measurement process should be set up that ensures sustainability of the EA management function. According to the typical management cycle as e.g. discussed by Deming in [14] or Shewart in [46] a governance function should be established that measures the achievement of objectives and if necessary adapts the EA management function accordingly by reentering the above presented configuration process. Furthermore, an extension mechanism needs to be implemented in the DTN for situational EA management in order to integrate new or update existing design theories if necessary.

4 Exemplifying BEAMS – Designing an Organization-Specific EA Management Function

In this section we describe an exemplary application of the BEAMS in a fictional organization, namely the financial service provider BSM.

The situation in respect to the organizational context of the EA management function of BSM can be characterized as follows: Over the years BSM purchased different other financial service providers, adapted their business processes, and incorporated their business applications. This has lead to a complex and highly heterogeneous application portfolio that BSM has difficulties in evolving and maintaining. In order to increase maintainability of its business applications, BSM wants to reduce their total number. For doing so, the organization decides to launch an EA management pilot project. With the organizational structure of BSM being grown over a series of acquisitions of other financial service providers, the organization has retained a number of independent IT units and hence a 'decentralized IT'. The EA management pilot is driven by a small EA management team located in a staff unit of the CIO's office. While this means that the EA management initiative can rely on high-level management support, especially the decentralized structure of the IT departments makes it necessary to use the pilot project for marketing and illustrating the benefit of the new and overarching management function. In addition, the EA management team has to deal with missing tool support for EA management as currently no further budget for the pilot is available.

Not aiming too high, BSM sets the goal of the EA management pilot to 'increased transparency' focused on the concern 'business applications used by organizational units'. This forms the initial EA-related problem that BSM seeks to address by the EA management initiative. This problem statement is covered by an IBB presented by BEAMS. This IBB comprises an information model that contains two classes BUSINESS APPLICATION and ORGANIZATIONAL UNIT as shown in Figure 8.

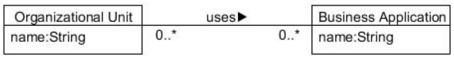

Fig. 8. Initial information model of BSM

Based on the identified organizational contexts an assessment of the BBs of BEAMS is performed, e.g. MBBs which build on the availability of an EA management tool are excluded from the selection process. The EA management team of BSM understands that the application owners would be pleased to deliver the information about their application's use in different organizational units to demonstrate the importance of the application they are responsible for. In contrast, stakeholders from the business departments may keep distance but can get in first contact with the EA management initiative. They decide to interview the application owners and according business departments to gather the corresponding information. The corresponding MBB is selected as it states (as CONSEQUENCE) to be beneficial for marketing, which can be performed prior to the interview by illustrating the objectives of the endeavor. The MBB shown in Figure 9 describes the steps of conducting interviews to gather information conforming to a concern. As a post-condition the MBB states that the corresponding concern is *documented* meaning the according information is available. Further, the MBB can be selected for the purpose that BSM wants to pursue, as interviews are well-suited to document current state EAs. Would the problem statement conversely aim at target or planned states of the EA, different MBBs had to be taken.

The MBB selected before states as a consequence that the application of the method may result in inconsistent information. Put in the context of our example, an application owner could state a business department, which according to his knowledge uses the respective business application but the corresponding

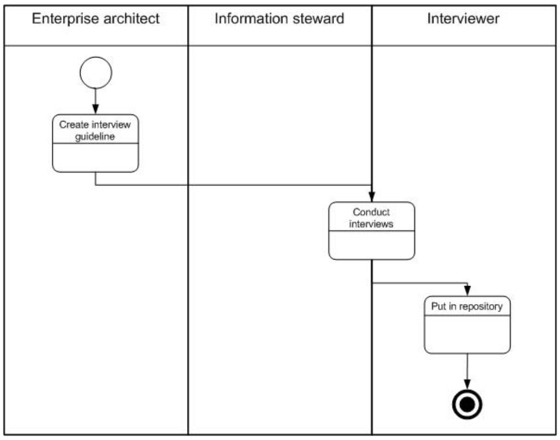

Fig. 9. MBB for gathering EA information via interviews

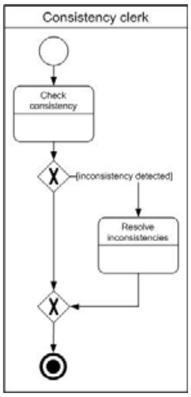

Fig. 10. MBB for checking consistency in the gathered information

business department denies that. In order to overcome this drawback of inconsistent information, another develop & describe-related MBB admissible in the given circumstances is available, which is concerned with tasks for consistency checking in EA documentations. The EA management team of BSM decides to make use of this MBB in order to ensure consistency in the documentation. The consistency-checking MBB as shown in Figure 10 takes a documented concern in its pre-condition and states that after the execution of the corresponding tasks, the concern, more precisely the corresponding information, is *consistent*.

BSM selects a communicate & enact MBB, which publishes the gathered information describing the current status of the EA via publishing it in the corporate intranet. Figure 11 illustrates the assembled method of BSM. During assembling the MBBs, the stakeholder variables are replaced by organization specific actors and roles, i.e. the 'information steward' is replaced by the 'application owners'

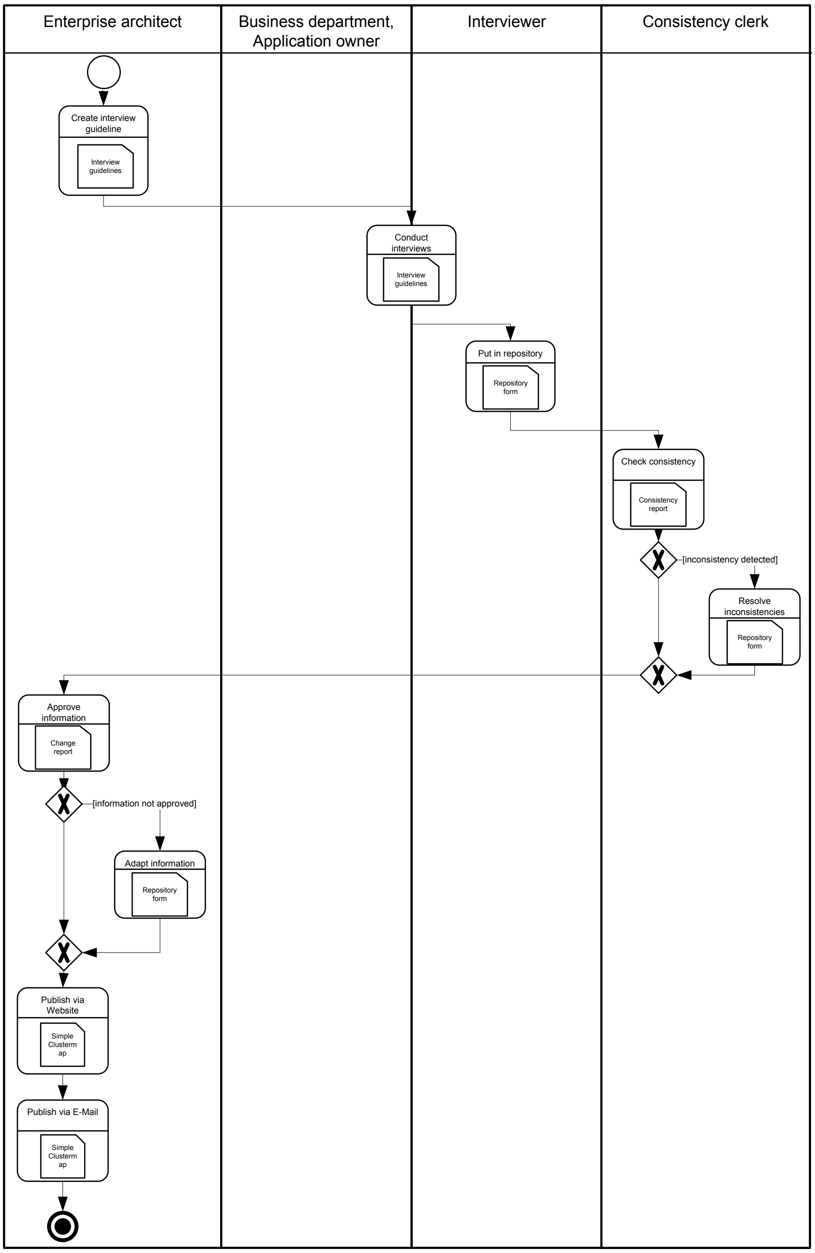

Fig. 11. Organization-specific method for increasing transparency

and ‘business departments’ respectively. Furthermore, the viewpoints utilized during the execution of the method are further specified utilizing VBBs as reports, forms, and maps. For the ‘publish via website‘ task for example, the VBB of a 'simple cluster map' is used. A graphical description of the VBB for a simple cluster map according to the information model given in Figure 8 is illustrated in Figure 12, while the resulting view is shown in Figure 13.

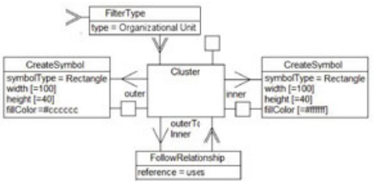

Fig. 12. VBB simple clustermap

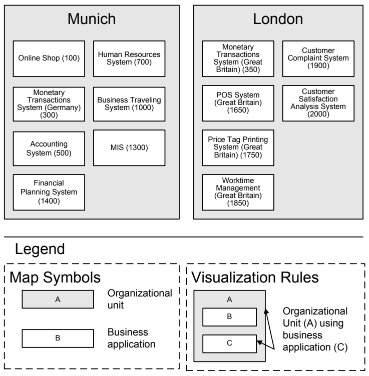
Munich
Online Shop (100)
Human Resources System (700)
Monetary Transactions System (Germany) (300)
Business Traveling System (1000)
Accounting System (500)
MIS (1300)
Financial Planning System (1400)
London
Monetary Transactions System (Great Britain) (350)
Customer Complaint System (1900)
POS System (Great Britain) (1650)
Customer Satisfaction Analysis System (2000)
Price Tag Printing System (Great Britain) (1750)
Worktime Management (Great Britain) (1850)
Legend
Map Symbols
A
Organizational unit
B
Business application
Visualization Rules
A
B
C
Organizational Unit (A) using business application (C)

Fig. 13. Simple clustermap ov BSM

Finally, BSM has to establish their organization-specific management function based on BBs of BEAMS. Therefore, the roles newly introduced by the EA management endeavor, e.g. the consistency clerk, have to be filled and the initial information gathering procedure has to be performed. As suggested in Section 3 a performance measurement process is set up. After publishing the current state of the EA via the corporate intranet, the CIO as the stakeholder of the problem statement as well as the business departments and the application owners as the participating users are asked for their satisfaction with the achieved results. Due to the positive feedback, the CIO decides to extend the scope of the EA management function. The goal of the EA management initiative is changed to 'increase homogeneity' of the application landscape. The concern is consequently extended, i.e. a superconcern in the sense of Buckl et al. [9] is used. This complementary leads to an extended information model for BSM to include the 'technologies used by a business application' as well as possibilities to define a 'technology as standard'. The corresponding IBB is illustrated in Figure 14.

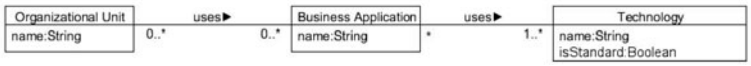

Fig. 14. Information model for increasing homogeneity

Performing further adaptations of the EA management function of BSM, the design team revisits the method framework of BEAMS (cf. Figure 4). In order to pursue the goal 'increase homogeneity' additional information about the organization's EA has to be gathered. In order to keep the investment for gathering the additional information low, the EA management team decides to send around questionnaires, in which the application owners are requested to state the technologies that their corresponding application uses. A corresponding MBB is used to incorporate this information. Based on this information an expert team assesses the technologies and decides on their standard conformity. More precisely, the the expert team decides which technologies should be supported as standards. In this sense, the EA management function is extended to incorporate activities for *analyzing and evaluating*, more precisely an MBB for this purpose. Finally, the EA management team decides to adapt the communication mechanisms by introducing an additional viewpoint conveying standardization relevant information. For doing so a specialized version of the cluster map is created decorating the symbol creation rules for the application with a color-coding indicating, whether all related technologies are standard-conform or not. Bringing together the different BBs a revised version of the EA management function as shown in Figure 15 is developed.

5 Related Work

The approach of BEAMS is not at lest due to his DTN nature different from the other approaches fro EA management as found in literature. But the DTN nature also explains the intricate relationship between BEAMS and different EA management approaches, from which BEAMS draws its proven-practice building blocks. Against that background a detailed comparison of BEAMS with the corresponding approaches is likely to fall short of any novel insights. We nevertheless take two selected 'traditional' EA management approaches as reference points of comparison (see Section 5.1) to highlight the different nature of BEAMS. Complementing these comparisons we further summarize two *contingency*-based approaches for EA management and show in Section 5.2 how these relate to BEAMS.

5.1 Traditional EA Management Approaches

The Open Group Architecture Framework (TOGAF) (cf. [48]) is perhaps the most well-known framework for EA management. In its most recent version 9.0, TOGAF presents both an architecture development method (ADM) and an information model for architectural description, the so called "enterprise content metamodel". The cyclic ADM is designed as reference method for performing an "architecture project", which in the sense of TOGAF is the natural way of performing EA management. Such architecture project is set up in a preliminary phase that decides over scope and reach of the project with respect to the EA, but also over the utilization of tools. Further, decisions are taken over the linkage of the EA management project to existing enterprise-level management processes, as e.g. business object management. During the first phsae of project execution, the reach and scope are further concretized via a selection of

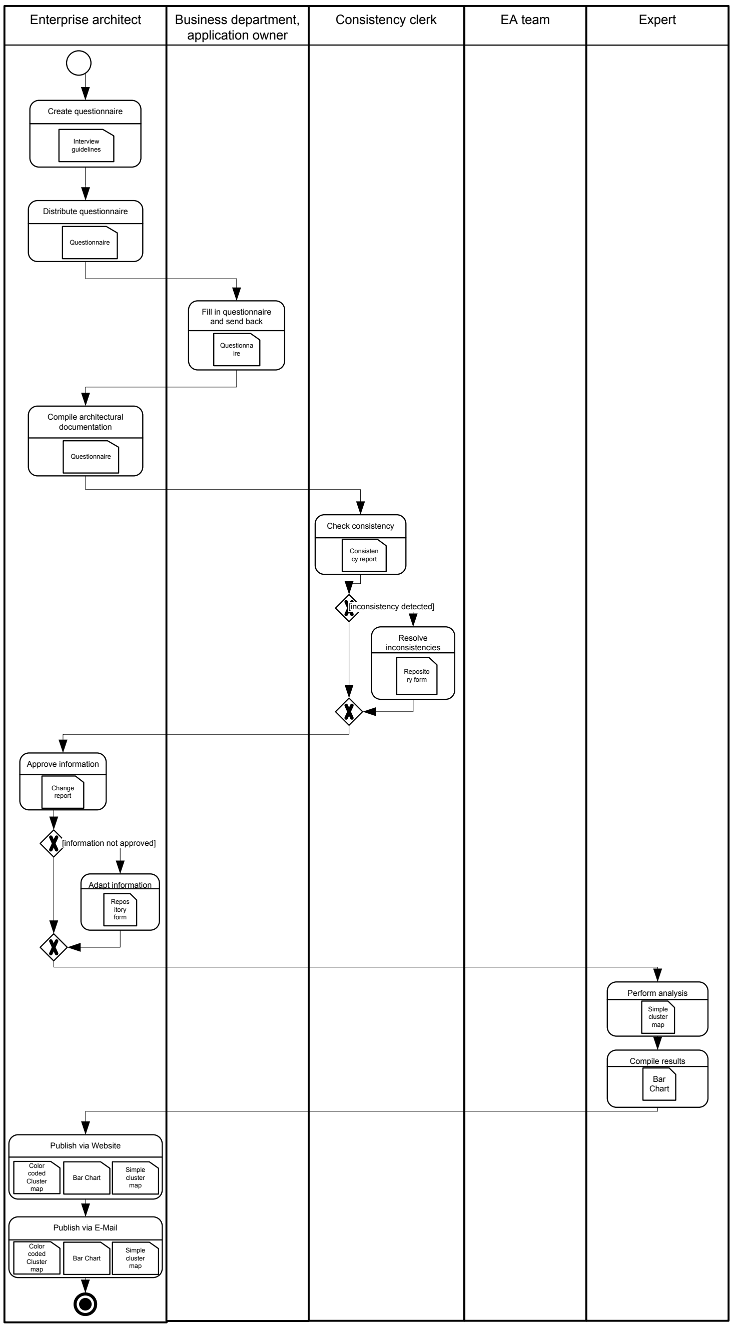

Fig. 15. Organization-specific method for increasing homogeneity

the corresponding stakeholders. Additionally, goals and requirements of the EA management project are defined making concrete prescriptions on the EA vision to pursue. In this sense the first two phases of the ADM relate to the activity of *configure and adapt* as described in the method framework of BEAMS (cf. Section 3). The EA vision is further detailed in three subsequent phases of the TOGAF ADM, namely the *business architecture*, *information systems architecture* and *technology architecture* development. During these phases the current state of the architecture is documented, a target state is developed, and gap analyses between these states are performed. In line with the prescriptions of TOGAF the documentation of the current state has to developed prior to the development of the target one. In the next two phases of the ADM (*opportunities and solutions* and *migration planning*) the results of the gap analysis are used to propose and plan transformation activities that change the overall EA. The implementation of this change is monitored in the *implementation and governance* phase, whereas the final phase *architecture change management* is concerned with assessing the overall performance of the EA management project. Complementing the description of the activities, TOGAF describes the input and output artifacts of the different phases, although the thereby utilized visualizations are only informally tied to the underlying information model. This content metamodel is presented as an extensible model centered around a monolithic core. The provided extensions introduce additional concepts such as KPIs or goals into the model, but no prescriptions are made, when to use which extension. Further, the information model remains on a rather abstract level abstaining from details as datatypes, multiplicities or constraints that would nevertheless be needed to ensure model consistency. Due to the project nature of TOGAF's EA management only a few remarks a spent on the establishment of a continuous EA management function, which is further reflected by the fact that maintenance activities for the EA documentation and continuous communication mechanisms are not discussed.

A very frequently quoted academic approach to EA management is the archimate approach presented by Lankhorst et al. in [33]. Central to this approach are the activities of documentation, communication and analysis of EAs. These activities are accounted for by introducing a specialized modeling language, the archimate modeling language, based on an information model (cf. e.g. [29]) covering the three facets *structure*, *behavior*, and *information* on the different architectural layers *business*, *application*, and *infrastructure*. This information model, while being monolithic in design, partially accounts for the diversity in the organizations understanding of their EAs by support 'short-cut' modeling. This means that, although the model assumes three distinct architectural layers, the layering is not strict, but intermediary concepts may be omitted, if necessary. This built-in flexibility of modeling does nevertheless not come without cost, but can lead to imprecise and inconsistent models, especially if concepts on an intermediary layer are added later. The lack of prescriptions in the field of model adaptation does not prevent the approach from giving comprehensive insights into both visualization and analysis techniques building on these models. The archimate approach presents different graphical notations for visualizing architectural information, as e.g. a “business support map”. Further, different quantitative analysis techniques building on the archimate meta-model are described (see e.g. [25]). Nevertheless, when it comes to prescriptions on how to adapt the archimate approach to a specific organization, literature becomes scarce and actual prescriptions are missing.

5.2 Contingency Approaches to EA Management

Only recently *contingency approaches* to EA management have gained some prominence as means to account for the organization-specificity of an EA management function. An early example of such approach is presented by Leppänen et al. in [34]. Central to their approach is the “EACon” framework, an organized collection of contingency factors of EA method engineering. This framework builds on rich literature in the field of EA management and devises the central contingency factors as found there, namely “EA method goals”, “enterprise” as well as “environment”, and “roles” as well as “resources”. These factors may well be identified either with the EA-related goals and the contextual factors of BEAMS, or with system of actor variables introduced therein. Leppänen et al. further detail on a contingency factor called “EA management” that is concerned with decisions rights and coordination means of EA management which are conversely covered by the actor modeling of BEAMS. Other aspects as “communication means” are only briefly alluded to in the contingency framework reflecting the strongly method centric perspective taken therein. This further reverberates in the rather short discussion on the concern of EA management showing that language aspects are not discussed by Leppänen et al. in [34]. Relating the work to the BEAMS approach, the contingency framework may well be used to structure and to organize the contextual influences that pertain to the approaches used to build BEAMS. When it nevertheless comes to concrete prescriptions or contingencies on aspects of EA description, the work of Lepänen et al. [34] may only serve as an abstract reference point providing possible dimensions of adaptation.

In [42], Riege and Aier outline a contingency approach to EA management with emphasis on the method aspect of the management activity. This is primarily reflected in discussion on the organizational setting that may constrain the implementation of an EA management function or project. Complementing these discussions, the approach presents abstract goal-like statements that may be helpful to frame the goal of the overall EA management activity. Concrete prescriptions on the method steps to be taken or actual EA-related goals for practical settings are conversely out of the scope of the contingency approach. The same is true, when it comes to language aspects, although Riege and Aier add a short side node on the “constitution of the EA”, which should adequately reflected as documentation fed to the management activities. In this sense, the BEAMS approach may be seen as stringent continuation of the work of Riege and Aier as presented in [42] accounting for both sides of the EA management coin, management methods and description languages.

6 Outlook

This article contributes a building block based approach to the field of EA management governance. With the BEAMS an organization seeking to establish a specific EA management function should be able to leverage operational and practice-proven design prescriptions. In this sense, the presented approach continues the work started by the contingency based EA management approaches of Leppänen et al. in [34] as well as Riege and Aier in [42]. With its grounding in the EAM pattern approach of Buckl et al. (cf [6,13]) and other practice-proven approaches from academic research as well as from standardization bodies, BEAMS provides a comprehensive approach to a highly relevant topic of information systems research and practice. With all the contributing approaches being successful in practice, one can sensibly assume that BEAMS is applicable in different practical settings, although a thorough evaluation on this topic is yet to be undertaken. First practice projects currently implemented building on the prescriptions of BEAMS nevertheless are developing in promising ways.

A future challenge in the context of BEAMS is associated with the evolvement of the approach itself. Up to this point, BEAMS is initialized with input from various sources, but as EA management-related research continues, future findings and results may provide a valuable addition to the approach. The method for constructing a DTN instantiation as presented by Pries-Heje and Baskerville (cf. discussions in Section 2.1) may be helpful in this context, but is by nature limited to one stream of evolution. In contrary we expect that BEAMS will attract a similar community as the EA management pattern catalog (cf. [13]) did, such that methods, mechanisms, and techniques for collaboratively evolving the knowledge base of BEAMS are needed. This especially applies with respect to the intermediary artifacts of the patterns and case descriptions that were used for constructing the initial set of building-blocks. The notion of these intermediary artifacts further relates to another future challenge – the tool support.

The development of an organization-specific EA management function is – even with the prescriptions and building-blocks provided by BEAMS – a complex task. The consistent integration of the MBBs keeping track of the pre- and post-conditions, respectively, requires careful attention. The same also applies for the integration of the information models and viewpoints that build the LBBs used in the EA management function. With this background and intricate interrelations, the design activity for organization-specific EA management functions calls for tool support. A building-block based a configurator for EA management functions may further be helpful as vehicle for evolving the knowledge base of BEAMS, as the configurations and adaptations made in such tool may be technically reflected to the knowledge base and analyzed using statistical means. On the contrary, the configuration tool must provide mechanisms to export the configured EA management function as a description file.

The work [36] of Matthes et al. describes that current tools for supporting EA management can be categorized into two large groups of tools, namely *meta-modeling* tools of high flexibility and *methodology-driven* tools delivering a pre-defined information model and method framework. Tools of the latter type may be regarded as 'traditional' EA management approaches in the sense of the discussion from Section 5.1, whereas tools of the former type may be used to implement arbitrary EA management approaches. In this sense, meta-modeling tools may interpret the description file exported from the BEAMS configurator. This conversely calls for a standardization of the exchange and configuration file format to facilitate the re-use of the defined configuration.

References

1. Aier, S., Kurpjuweit, S., Saat, J., Winter, R.: Enterprise Architecture Design as an Engineering Discipline. AIS Transactions on Enterprise Systems 1, 36–43 (2009)
2. Aier, S., Riege, C., Winter, R.: Unternehmensarchitektur – Literaturüberblick Stand der Praxis. Wirtschaftsinformatik 50(4), 292–304 (2008)
3. Alexander, C.: The Timeless Way of Building. Oxford University Press, New York (1979)
4. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S.: A Pattern Language. Oxford University Press, New York (1977)
5. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F.: Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008). Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany (2008)
6. Buckl, S., Ernst, A.M., Lankes, J., Schneider, K., Schweda, C.M.: A pattern based approach for constructing enterprise architecture management information models. In: Wirtschaftsinformatik 2007, Karlsruhe, Germany, pp. 145–162. Universitätsverlag Karlsruhe (2007)
7. Buckl, S., Gulden, J., Schweda, C.M.: Supporting ad hoc analyses on enterprise models. In: 4th International Workshop on Enterprise Modelling and Information Systems Architectures (2010)
8. Buckl, S., Krell, S., Schweda, C.M.: A formal approach to architectural descriptions – refining the iso standard 42010. In: 6th International Workshop on Cooperation & Interoperability – Architecture & Ontology, CIAO 2010 (2010)
9. Buckl, S., Matthes, F., Schweda, C.M.: Interrelating concerns in ea documentation – towards a conceptual framework of relationships. In: 2nd European Workshop on Patterns for Enterprise Architecture Management (PEAM 2010), Paderborn, Germany (2010)
10. Buckl, S., Matthes, F., Schweda, C.M.: A modeling language for describing ea management methods. In: Modellierung betrieblicher Informationssysteme (MobIS 2010) (2010)
11. Buckl, S., Matthes, F., Schweda, C.M.: Towards a method framework for enterprise architecture management – a literature analysis from a viable system perspective. In: 5th International Workshop on Business/IT Alignment and Interoperability (BUSITAL 2010) (2010)
12. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-oriented software architecture: a system of patterns. John Wiley & Sons, Inc., New York (1996)
13. Chair for Informatics 19 (sebis), Technische Universität München. Eam pattern catalog wiki (2010), `http://eampc-wiki.systemcartography.info` (cited 2010-07-01)
14. Deming, E.W.: Out of the Crisis. Massachusetts Institute of Technology, Cambridge (1982)

15. Eisenhardt, K.M., Graebner, M.E.: Theory building from cases: Opportunities and challenges. Academy of Management Journal 50(1), 25–32 (2007)
16. Ernst, A.: Enterprise architecture management patterns. In: PLoP 2008: Proceedings of the Pattern Languages of Programs Conference 2008, Nashville, USA (2008)
17. Ernst, A.M.: A Pattern-Based Approach to Enterprise Architecture Management. PhD thesis, Technische Universität München, München, Germany (2010)
18. Frank, U.: Multi-perspective enterprise modeling (memo) – conceptual framework and modeling languages. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 2002), Washington, DC, USA, pp. 1258–1267 (2002)
19. Gamma, E., Helm, R., Johnson, R., Vlissides, J.M.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional Computing Series. Addison-Wesley Professional, Munich (1994)
20. Gehlert, A., Schermann, M., Pohl, K., Krcmar, H.: Towards a research method for theory-driven design research. In: Hansen, H.R., Karagiannis, D., Fill, H.G. (eds.) Business Services: Konzepte, Technologien, Anwendungen, 9, Wien, Austria. Internationale Tagung Wirtschaftsinformatik, vol. 1, pp. 441–450. Österreichische Computer Gesellschaft (2009)
21. Hafner, M., Winter, R.: Vorgehensmodell für das management der unternehmensweiten applikationsarchitektur. In: Ferstl, O.K., Sinz, E.J., Eckert, S., Isselhorst, T. (eds.) Wirtschaftsinformatik, pp. 627–646. Physica-Verlag, Heidelberg (2005)
22. Harmsen, A.F.: Situational Method Engineering. PhD thesis, University of Twente, Twente, The Netherlands (1997)
23. Henderson, J.C., Venkatraman, N.: Strategic alignment: leveraging information technology for transforming organizations. IBM Systems Journal 38(2-3), 472–484 (1999)
24. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28(1), 75–105 (2004)
25. Iacob, M.-E., Jonkers, H.: Quantitative analysis of enterprise architectures. In: Konstantas, D., Bourrières, J.-P., Léonard, M., Boudjlida, N. (eds.) Interoperability of Enterprise Software and Applications, Geneva, Switzerland, pp. 239–252. Springer, Heidelberg (2006)
26. International Organization for Standardization. ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems (2007)
27. IT Governance Institute. Framework Control Objectives Management Guidelines Maturity Models (2009), `http://www.isaca.org/Knowledge-Center/cobit` (cited 2010-06-18)
28. Johnson, P., Ekstedt, M.: Enterprise Architecture – Models and Analyses for Information Systems Decision Making, Studentlitteratur, Pozkal, Poland (2007)
29. Jonkers, H., van Burren, R., Arbab, F., de Boer, F., Bonsangue, M., Bosma, H., ter Doest, H., Groenewegen, L., Scholten, J., Hoppenbrouwers, S., Iacob, M.-E., Janssen, W., Lankhorst, M., van Leeuwen, D., Proper, E., Stam, A., van der Torre, L., van Zanten, G.: Towards a language for coherent enterprise architecture descriptions. In: 7th International Enterprise Distributed Object Computing Conference (EDOC 2003), Brisbane, Australia. IEEE Computer Society, Los Alamitos (2003)
30. Kühn, H.: Methodenintegration im Business Engineering. PhD thesis, Universität Wien (2004)

31. Kurpjuweit, S., Winter, R.: Viewpoint-based meta model engineering. In: Reichert, M., Strecker, S., Turowski, K. (eds.) Enterprise Modelling and Information Systems Architectures – Concepts and Applications, Proceedings of the 2^{nd} International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2007), St. Goar, Germany, Bonn, Germany, October 8-9. LNI, pp. 143–161. Gesellschaft für Informatik (2007)
32. Langenberg, K., Wegmann, A.: Enterprise architecture: What aspect is current research targeting? Technical report, Laboratory of Systemic Modeling, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland (2004)
33. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Heidelberg (2005)
34. Leppänen, M., Valtonen, K., Pulkkinen, M.: Towards a contingency framework for engineering and enterprise architecture planning method. In: 30^{th} Information Systems Research Seminar in Scandinavia (IRIS), pp. 1–20 (2007)
35. Luftman, J.N., Lewis, P.R., Oldach, S.H.: Transforming the enterprise: The alignment of business and information technology strategies. IBM Systems Journal 32(1), 198–221 (1993)
36. Matthes, F., Buckl, S., Leitel, J., Schweda, C.M.: Enterprise Architecture Management Tool Survey 2008. Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany (2008)
37. McDermin, J.A.: Software engineer's reference book. Butterworth Heinemann, Oxford (1994)
38. Niemann, K.D.: From Enterprise Architecture to IT Governance – Elements of Effective IT Management. Vieweg+Teubner, Wiesbaden (2006)
39. Noble, J.: Classifying relationships between object-oriented design patterns. In: Australian Software Engineering Conference (ASWEC), pp. 98–107. IEEE Computer Society, Los Alamitos (1998)
40. Nunamaker, J.F., Chen, M., Purdin, T.D.M.: Systems development in information systems research. J. Manage. Inf. Syst. 7(3), 89–106 (1991)
41. Pries-Heje, J., Baskerville, R.: The design theory nexus. MIS Quarterly 32(4), 731–755 (2008)
42. Riege, C., Aier, S.: A contingency approach to enterprise architecture method engineering. In: Service-Oriented Computing - ICSOC 2008 Workshops. LNCS, vol. 5472, pp. 388–399 (2009)
43. Schekkerman, J.: Enterprise Architecture Good Practices Guide – How to Manage the Enterprise Architecture Practice. Trafford Publishing, Victoria (2008)
44. Schelp, J., Winter, R.: On the interplay of design research and behavioral research – a language community perspective. In: Proceedings of the Third International Conference on Design Science Research in Information Systems and Technology (DESRIST 2008), Westin, Buckhead, Atlanta, Georgia, USA, Georgia State University, Atlanta, May 7-9, pp. 79–92 (2008)
45. Schönherr, M.: Towards a common terminology in the discipline of enterprise architecture. In: Aier, S., Johnson, P., Schelp, J. (eds.) Pre-Proceedings of the 3^{rd} Workshop on Trends in Enterprise Architecture Research, Sydney, Australia, pp. 107–123 (2008)
46. Shewhart, W.A.: Statistical Method from the Viewpoint of Quality Control. Dover Publication, New York (1986)
47. Standish Group International. The chaos report 2009 (2009), `http://www.standishgroup.com/newsroom/chaos_2009.php` (cited 2010-02-25)

48. The Open Group: TOGAF "Enterprise Edition" Version 9 (2009), `http://www.togaf.org` (cited 2010-02-25)
49. van Aken, J.E.: Management research based on the paradigm of the design sciences: The quest for field-tested and grounded technological rules. Journal of Management Studies 41(2), 219–246 (2004)
50. van den Berg, M., van Steenbergen, M.: Building an Enterprise Architecture Practice – Tools, Tips, Best Practices, Ready-to-Use Insights. Springer, Dordrecht (2006)
51. Wegmann, A.: The Systemic Enterprise Architecture Methodology (SEAM). Technical report, EPFL (2002)
52. Wegmann, A., Balabko, P., Lê, Regev, G., Rychkova, I.: A method and tool for business-it alignment in enterprise architecture. In: Proceedings of the CAiSE 2005 Forum, Porto, Portugal, pp. 113–118 (2005)
53. Zachman, J.A.: A framework for information systems architecture. IBM Syst. J. 26(3), 276–292 (1987)
54. Zelewski, S.: Erkenntnisinstrumente der betriebswirtschaftslehre. In: Betriebswirtschaftslehre – Band1, München, Germany, Oldenbourg (2008)