A Design Theory Nexus for Situational Enterprise Architecture Management – Approach and Application Example

Sabine Buckl, Christian M. Schweda, Florian Matthes
Software Engineering for Business Information Systems
Technische Universität München
Munich, Germany
{sabine.buckl,matthes,schweda}@in.tum.de

***Abstract*—Today's enterprises are confronted with an ever changing environment that demands continuous transformations. Globalized markets, disruptive technological innovations, and new legal regulations call for enterprises, which flexibly adapt to these requirements. A commonly accepted instrument to guide such enterprise transformations is enterprise architecture (EA) management. Enterprises seeking to introduce and establish such a management function see themselves confronted with a plethora of tools, approaches, and frameworks that claim to provide 'the definitive design prescriptions' for an EA management function. The applicability of the different prescriptions nevertheless heavily depends on the organizational context in which such a management function has to be embedded and the EA-related goals that the enterprise wants to pursue.**

This paper presents a *design theory nexus* for situational EA management, that helps enterprises to choose the EA management approach best suited for their specific situation, i.e. their goals and organizational context. In order to build such a nexus, related work from the fields of situational method engineering and the viable systems model are revisited. Furthermore, prominent approaches to EA management are discussed in the paper and the explicitly or implicitly stated organizational contexts, in which they can operate, as well as the EA management goals, which the approaches are dedicated to, are elicited. Based on the thereby identified constraints a design theory nexus for situational EA management is developed. Utilizing the design theory nexus an enterprise can specify its application context as well as its EA management goals, and is provided with a selection of suitable approaches. The EA management pattern catalog of the TU Munich is used to exemplify the applicability of the approach. Finally an outlook critically reflects the findings of the paper and discusses areas of future research.

***Keywords*-Enterprise architecture management function, situational method engineering, viable systems model, design theory nexus**

I. Motivation

Modern organizations find themselves confronted with ever changing economic, regulatory, and technical environments that they are forced to continuously adapt to (cf. [42], [51]). Performing the changes that are necessary or could help to leverage opportunities is a complex task, aggravated by the intricate and highly interwoven architecture of the overall organization. Thereby, architecture or the enterprise architecture (EA) respectively is understood in accordance with the ISO Std. 42010 as the "fundamental organization of a system [the organization] embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution." [29, page 3] Local adaptations of one component of this architecture, e.g. a organizational unit, business process, or business application, might have unforeseen global consequences at and potentially detrimental impacts on related artifacts.

According to Aier et al. in [2], Enterprise adaptations can be differentiated into optimization (incremental change) and transformation (fundamental change). Whereas support for the former type of change is typically provided by functional methods of business administration, e.g. human resources, distribution, or marketing, the latter requires a holistic management function to systematically support organizational transformation [53]. A commonly accepted instrument to support and guide such transformations is EA management, whose main goal is to enhance and maintain the mutual alignment of business and IT (cf. [34], [26]). An effectively established EA management function leads to a) reduction of local maintenance costs due to increased standardization (cf. [50]), b) increased responsiveness via reduced project duration (cf. [48], [50]), c) facilitates risk management through reduced complexity and a organization-wide view on organizational changes (cf. [48], [42]), and d) enhances strategic business outcomes by increasing effectiveness of business processes, applications, etc. through standardization (cf. [9]).

Although the topic of EA management has been approached from various directions like engineering perspectives e.g. by Aier et al. in [2], systemic perspectives by Buckl et al. in [15] or Wegmann in [52], and business-centric perspectives, e.g. Scheer in [43], no commonly accepted understanding or definition of EA management has yet emerged. Schönherr gives in [46] an overview on existing approaches to EA management and shows, that 34% of the analyzed publications on EA management are not complemented with a definition of the term *enterprise architecture*. Building on the idea of language communities, Schelp and Winter explain this circumstance in [44] in the sense that the major research groups concerned with the area of EA management, form partially independent language communities, meaning that only within the groups a common terminology and understanding of the respective terms, EA or EA management respectively, exists. This phenomena can be exemplified utilizing the term *EA*. While some authors aim at a clear distinction between the artifact *enterprise architecture* and the management function concerned with the evolution of the EA (*enterprise architecture management*) (cf. [50], [37], [48]). Other communities regard the term EA to be more *normative* and hence consider the planning process as an integral part of the EA itself (cf. [24], [42], [51]). Thereby, many of the latter approaches stay on a rather abstract level with in respect to the description of the planning function.

The absence of commonly accepted step-by-step guidelines for managing the EA 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 an *one-size-fits-them-all* approach, but expect the management function to be organization-specific (cf. [13], [49], [32], [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 connection paths 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" [28]. The objectives of an EA management initiative in contrast vary widely. While typical goals of EA management can be summarized on a very abstract level, they have to be substantiated during the development and establishment of an appropriate 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. Existing approaches to EA management often originate from practical backgrounds, as e.g. the ones proposed by Niemann in [37], Hanschke in [23], and Schekkermann in [45] and therefore describe success stories within predefined organizational contexts, which are more often than not only implicitly defined, e.g. in an expository example. Other origins of existing EA management approaches are academic institutions, e.g. Aier et al. in [3], Ross in [41], and Frank in [21], or standardization bodies, as e.g. The Open Group Architecture Framework (TOGAF) in [48]. The latter approaches 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 abstain from providing information on how to perform this task. 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 paper addresses the aforementioned gap of organization-specificity of EA management by proposing a *design theory nexus* for situational EA management, which facilitates the selection of a suitable EA management approach based on the constraining organizational context and the pursued goals of the respective organization. The design theory nexus for situational EA management, is based on the work of Pries-Heje and Baskerville who initially introduced the construct of a design theory nexus for competing solutions in the context of decision support systems (cf. [39]), a systemic perspective as discussed by Beer in [6], [7], [8], and the idea of situational method engineering presented by Harmsen in [25]. Both approaches are revisited in Section II to illustrate their applicability in the context of EA management. Based on this prefabrics, Section III proposes a design theory nexus for situational EA management and details on its constituents, which are exemplified alongside a collection of best practice EA management patterns as contained in the *Enterprise Architecture Management Pattern Catalog (EAMPC)* of the TU Munich [16]. Final Section IV gives a critical reflection of the approach and an outlook on future topics of research.

II. Related work

Developing and establishing an EA management function, which suits the specific needs of the respective organization is a challenging task, which is characterized by evaluating the increasing set of competing solutions offered by existing EA management approaches (cf. Langenberg and Wegmann in [33]). Thereby, the evaluation should be based on a) the goals pursued by the respective EA management initiative and b) take the organizational context, in which the EA management function is embedded into account. Two different approaches to facilitate the aforementioned challenge are discussed subsequently. First, the construct of a design theory nexus, which provides "a set of constructs and methods that enable the construction of models that connect numerous design theories with alternative solutions." [39, page 733], is introduced and second, the idea of situational method

[1] For a in-depth discussion of different software development process models see [36].

engineering is presented, which describes how a method can be "tailored and tuned to a particular situation." (cf. Harmsen in [25, page 25]).

A. A design theory nexus for competing solutions

In [39] Pries-Heje and Baskerville present the idea of a *design theory nexus* as means to connect existing approaches, i.e. design theories, which provide competing solutions for a problem domain. A design theory nexus thereby not only connects competing solutions, but further helps "decision makers in choosing which of the theories are most suitable for their particular goals and their particular setting" [39, page 733], i.e. the organizational environment. Pris-Heje and Baskerville discuss that the presented approach is especially useful in cases of solving so-called *wicked problems*. Wicked problems, according to Rittel and Weber in [40] share the following characteristics: i) they can only be formulated in terms of a solution, ii) solutions are value laden and can only be denoted "good" or "bad" instead of "true" or "false", iii) their solution space is unbound, iv) solutions are irreversible, and v) wicked problems neither can be approached nor can alternatives be evaluated without engaging considerable uncertainty.

Based on the above understanding of wicked problems, developing and establishing an organization-specific EA management function can be understood as wicked problem. As already mentioned in the motivating Section I EA management approaches in literature are mostly derived from success stories in practice. This characteristic of EA management solutions can be found in academic approaches as well as in practitioners' approaches (i). Applying such a best practice solution in another setting can only be evaluated as being good or bad, more precisely as "working good" or "working bad" (ii). Furthermore, a common agreement on the main activities of the EA management function in the meantime exists (cf. Aier et al. in [1]), although, the realization of these activities is unbound, e.g. analysis of different states of the EA can be performed in various ways ranging from expert-based analysis (cf. [37], [55]) via pattern-based ones (cf. [17], [12]) to quantitative assessments via metrics (cf. [27], [22], [31]). Similarly, the main objective of EA management – fostering mutual alignment of business and IT by providing decision-support regarding the enterprise transformation – on the one hand is commonly agreed upon, while on the other hand the more detailed objectives of an particular EA management endeavor in an organization change over time (iii). Finally, the establishment of an EA management function cannot be simulated or tested due to the complex management subject (iv) or at least requires a simplification of the examined interrelations, which makes predictions on the appropriateness of a management approach only possible with considerable uncertainty (v). Following this understanding of establishing an EA management function as a wicked problem, for which a plethora of competing solutions exist, a design theory nexus for EA management[2] can be developed, which provides assistance in choosing a suitable EA management approach according to the organizations' goals and organizational context.

[2]Following the terminology as introduced by Pries-Heje and Baskerville in [39], the correct naming would be *design theory nexus instantiation*. We omit the term *instantiation* in the following due to reasons of readability.

According to Pries-Heje and Baskerville in [39, page 743], a design theory nexus consists of the following five constructs:

- **Goals** describe what the system is intended for.
- **Environment** refer to contingencies, which are 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 highly dissimilar decision alternatives.

Figure 1 provides an overview about the single components of a design theory nexus and illustrates their relationships.

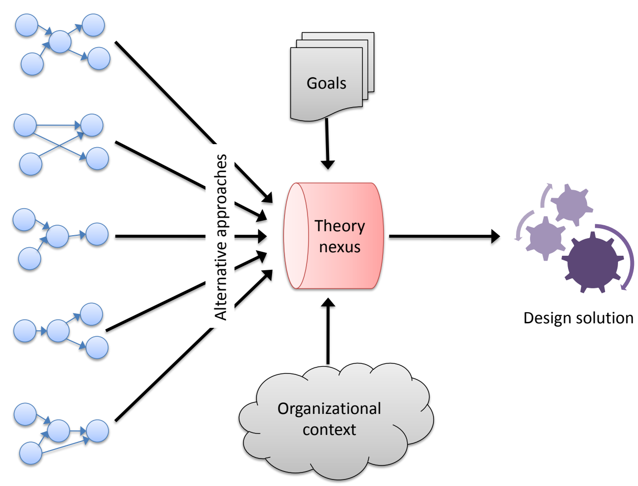

Fig. 1. Components of a design theory nexus according to Pries-Heje and Baskerville in [39]

The construction of a design theory nexus according to Pries-Heje and Baskerville (cf. [39]) 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, which must hold for the approach to achieve the highest utility. Here, it has to be noted that these conditions might be unequal 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 developed assertions is undertaken. Final step five combines the approaches, conditions, assertions, and the process into a tool, which supports the evaluation regarding the fit for each approach in a given situation.

B. A systemic perspective on EA management

Enterprises form complex systems consisting of various elements with a large number of interdependencies. In order to survive, an enterprise has to adapt to changes in the environment, e.g. changing markets or legal regulations. The viable system model (VSM), developed by Beer [6], [7], [8], provides a framework to describe such systems. According to the model, five interacting subsystems – *operation*, *coordination*, *control*, *planning*, and *identity* – can be identified. The VSM has been applied in various contexts, e.g. project management [10] or organizational modeling [11], [20].

Similar to EA management, the VSM aims according to [11] to support an enterprise during the implementation of large scale organizational transformations. Whereas a definition and description for each of the systems of the VSM is given in e.g. [6] no such common understanding about the constituents of the function of EA management exists. Therefore, the five subsystems of the VSM are subsequently detailed and mapped to typical activities performed as part of an EA management function.

System one – operation – contains the primary activities of the system under consideration, which directly interact with the environment. In the context of EA management these primary activities are identified with enterprise-level management functions, e.g. project lifecycle management, project portfolio management, synchronization management (cf. [54]). The enterprise-level management functions align with the projects that actually change the EA. These projects are initiated in the demand management, aligned in the strategies and goals management, selected in the project portfolio management, scheduled in the synchronization management, and realized with standards from the IT architecture management. A description of the function of EA management therefore must consider the role of related enterprise-level management functions.

System two – coordination – includes the information channels and bodies, which ensure that the primary activities of **System one** work harmoniously in coordination. EA management provides a common basis and the means for communication between the various stakeholders with business and IT background involved in the enterprise-level management functions. Therein, especially visualizations to support *communication* are used and exchanged between the different management functions to coordinate their activities. All project proposals originating from the demand management for example, are used as input to create possible planned landscapes to prepare the project portfolio management [35], [54]. Accordingly, the EA management function mus encompass a communication dimension.

System three – control – represents the structures and controls, which establish the responsibilities and rights to maintain the resource allocation of the operating system **System one**. Thereby, **System three** monitors the primary activities as well as the communication and coordination tasks of **System two** and adapts them according to the holistic view on the primary activities. If, for example, newly agreed standards from IT architecture management are not available for the project portfolio management, the projects considered therein cannot be checked for standard compliance. **System three** should therefore set up a structure, e.g. a wiki, where the standards can be viewed and communicated to the respective stakeholders. **System three** can be referred to as *reactive* EA management.

System four – planning – contains the EA intelligence function. The system is concerned with a holistic and future-oriented perspective to support strategic decision making. Whereas **System three** is capable of dealing with immediate concerns, **System four** focuses on future aspects, which emerge from the system's environment and also considers strategic opportunities, threats, and possible future directions. Typical processes in **System four** in the context of EA management include the analysis of the status quo of the architecture, the development of a target architecture representing the envisioned state in the future, and planning the transformation of the enterprise to pursue the target. Alongside the reactive aspect, an EA management approach must cover the aforementioned *proactive* aspect, containing a vision how a possible target enterprise should look like.

System five – identity – is responsible for managing the overall policy decisions. It should provide clarity about the overall direction, values, and purpose of the system under consideration. The main goal of **System five** is to balance present and future efforts, and to steer the system as a whole. In the context of EA management, **System five** addresses concerns like the scope and reach of EA management. Typically, a piloting project is performed in the initial phase of an EA management endeavor, e.g. starting with a limited number of concerns, e.g. compliance issues, availability aspects, or with restricted reach e.g. within one business department. Nevertheless, after the initial phase, when the EA management has matured and become more adopted, an *EA management governance* is established to redefine EA management scope and reach. Following typical quality control cycles [18], [47] the EA management governance aspect is concerned with measuring the achievement of the pursued goals.

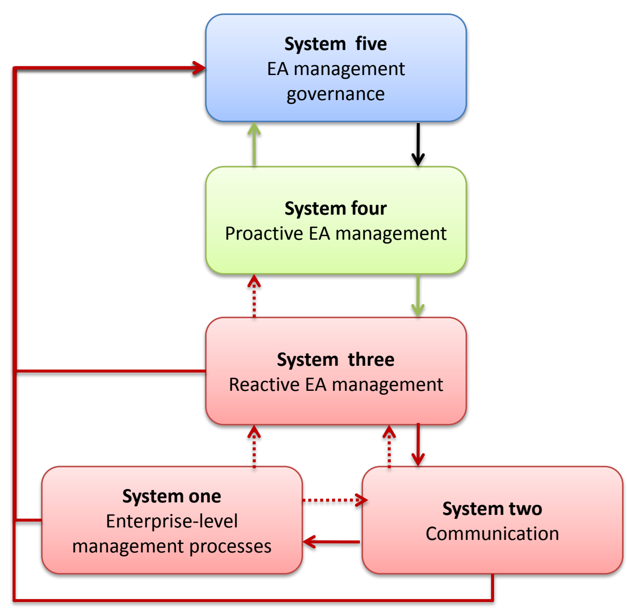

Fig. 2. Applying a viable system perspective to EA management

Summarizingly, the **Systems one** to **three** can be regarded as managing the *inside and now* of the EA whereas **Systems four** and **five** manage the *outside and the future* of the EA. In the context of EA management, the former systems relate to the operative EA management tasks – *running the enterprise* – while the latter ones consider the strategic EA management ones – *changing the enterprise*. The application of the VSM to EA management as described above is illustrated in Figure 2.

This systemic view on EA management is further complemented with the concept of the *algedonic signals* from the VSM. These signals, triggered up **Systems one** to **three**, provide an alerting mechanism, which is employed, if one of these systems is not able to perform as intended in the current situation. Such a signal is escalated to **System five**, which then can adapt the overall management function and can provide guidance to maintain the identity, i.e. the purpose of the EA management system. To exemplify these considerations, one may think of an EA getting increasingly heterogeneous albeit a standardization board has been established. At the point, this board notices that it has no means to counteract the tendency, an alert is escalated to the EA management governance. The governance function then has to e.g. empower the board to stop non standard conform projects, in order to enact the envisioned homogenization, or to rise the question, if a standardized EA is necessary in the future.

C. Situational method engineering

Motivated by the plurality of proposed methods for standardizing information system engineering on the one hand and the increasing application area diversification and complexity on the other hand, Harmsen presented in [25] 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." [25, page 6] Introducing the term *controlled flexibility* Harmson elicits requirements for a method engineering approach, which accomplishes standardization and at the same time is flexible enough to math the situation at hand. A *situation* thereby refers to the combination of circumstances at a given point in time in a given organization [25]. In order to address these requirements, for each situation a suitable method – so-called situational method – is *constructed*, which completely takes into account the circumstances applicable in the respective situation. In the construction process uniform method fragments are selected, which can be configured an adapted with the help of formally defined guidelines.

The generic process to constructing situational methods consists of four steps, which are detailed subsequently. Input to the configuring 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 resulting in a situational method. During assembling method fragments, aspects like completeness, consistency, efficiency, soundness, and applicability is accounted for (cf. [25, page 240seqq]). 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 illustrates the relationships between the different steps.

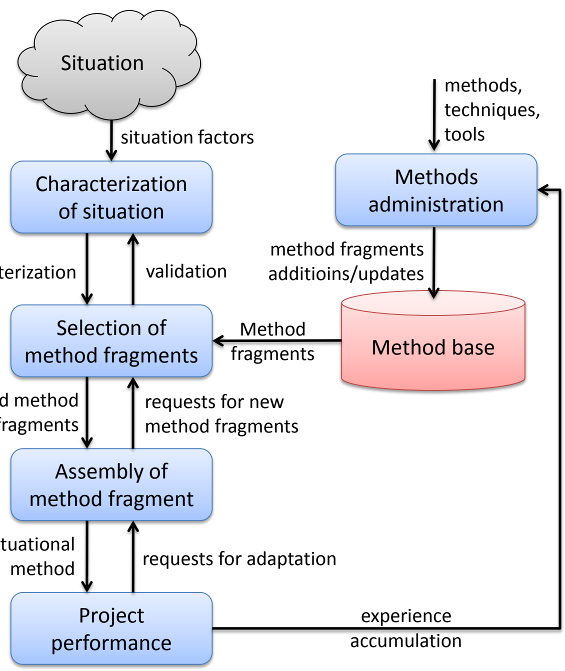

Fig. 3. The process of situational method engineering according to Harmsen in [25]

Complementing the construction process of a situational method for a given situation, Harmsen introduces in [25] the activity *method administration*, which captures methodological 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 [25]. An objective, which we likewise stated in the area of EA management in the motivating section of this paper. The situation described by Harmsen in information system engineering 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, which picks up the idea of the design theory nexus presented by Pries-Heje and Baskerville in [39] and the approach of situational method engineering presented by Harmsen in [25].

III. A DESIGN THEORY NEXUS FOR SITUATIONAL EA MANAGEMENT

The design theory nexus for situational EA management is heavily influenced by the work presented in Section II. Utilizing the concept of a design theory nexus as introduced by Pries-Heje and Baskerville in [39], we detail on the single components of a design theory nexus for situational EA management in the following. The design theory nexus for EA management is developed utilizing the five-step method as introduced before. Furthermore, the process of situational method engineering is used to enhance the construction of the design theory nexus and provides guidance for the application of the nexus as sketched at the end of this section.

While the framework developed by Zachmann in 1978 (cf. [56]) is commonly regarded to be the hour of birth of the topic EA management, the number of researchers and practitioners targeting this area of interest since that time has increased [33]. An overview on the current state-of-the-art in EA management is given by Aier et al. in [4] and the most active research groups in the area are determined by Schelp and Winter in [44] and Aier and Schelp in [5]. We utilize the thereby identified "major players" and their approaches to designing an EA management function in step one of the construction of the design theory nexus for situational EA management as input for the competing theories. Accordingly, the approaches of the following groups form the basis of our subsequent elaborations[3]:

- EPFL Lausanne, Switzerland
- Telematica Institute, 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 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 establishment of an organization-specific EA management function, e.g. an engineering based approach as presented by Aier et al. in [2], a pattern-based approach presented by Buckl et al. in [14], [16], or an analysis-focused approach introduced by Johnson and Eksted in [30]. These different approaches or the contained methods represent the input for our design theory nexus for situational EA management. Based on the systemic perspective on EA management discussed in Section II, the approaches can be analyzed and evaluated for their coverage of all systems from the operation level to the governance system.

For reasons of brevity, the construction of the design theory nexus for situational EA management is exemplified alongside the EAMPC of TU Munich, which provides a catalog of best practices gathered from industry and academia and therefore can itself be regarded as a collection of competing design theories. Other prominent approaches as e.g. listed by Aier et al. in [4] are only sketched in the following, while the examples are taken from the EAMPC.

The EAMPC contains four different types of patterns [16]

- **Methodology patterns** contain activities, which need to be conducted in order to solve a given problem.
- **Viewpoint patterns** describe diagrams, figures, tables, listings, which have been proven helpful in communicating EA-related information.
- **Information model patterns** define the information, which is required in order to solve a certain problem or to generate a particular viewpoint.
- **Anti patterns** illustrate approaches that have proven not to be successful in the context of EA management.

While the former three types of pattern provide the input for our design theory nexus for situational EA management, the latter type, anti-pattern contributes to the construction of the evaluation guidelines as detailed below.

Following the idea of patterns as e.g. introduced by Alexander et al. in Al77 different types of relationships between pattern may exists (cf. Noble in [38]). 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 design theory nexus for situational EA management.

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 approach presented by Ernst in [19][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.

[3]For a more detailed discussion of the contributions of these groups as well as for literature references see Schelp and Winter in [44].

[4]The approach presented by Ernst in [19] represents an excerpt, which is also contained in the EAMPC [16]

- 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 our theory nexus for situational EA management:

- 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 approaches. The suitability of the competing approaches for any combination of the conditions can then be defined utilizing a *fitting matrix* with the competing approaches 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 [19], 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 approaches 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 tool supporting the utilization of the design theory nexus for situational EA management is developed in final step five. Thereby, the competing approaches, goals, organizational contexts, a well as the process, which applies the fitting matrix, are reflected in the design of the tool. Possible realizations of the tool may range from simple excel-based approaches in line with the scoring matrix of Pries-Heje and Baskerville (cf. [39]) to more sophisticated tools, which cannot only be used for selecting an appropriate EA management approach, but provide further assistance for performing single activities of the EA management function, like e.g. documenting a current state of the EA.

Based on the developed design theory nexus for situational EA management, 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 respective 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 nexus 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 Harmson in [25], a performance measurement process should be set up, which ensures sustainability of the EA management function. According to the typical management cycle as e.g. discussed by Deming in [18] or Shewart in [47] a governance function should be established, which 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 design theory nexus for situational EA management in order to integrate new or update existing design theories if necessary.

IV. Conclusion and Outlook

This paper addresses a common but intractable problem faced by managers of organizations willing to introduce an EA management approach. While the importance of EA management as competitive advantage in times of ever changing environments for organizations is unquestioned, a plethora of different EA management approaches has been developed in academia and practice. These approaches reflect the various goals and organizational contexts available, but propose highly dissimilar, competing solutions with different focus. This reflects the diversity of EA management as enacted in different organizations, which can further be illustrated from a viable system perspective. In particular system five (identity) stresses the importance of defining the scope and reach of an EA management endeavor.

The design theory nexus for situational EA management proposed in this paper addresses the challenge of selecting "the right" EA management approach by providing a decision support system. With this system the approach(es) optimally suited under the constraints of the EA management goals pursued and the environmental context of the organization can be selected. The development of such a design theory nexus for situational EA management has been sketched in the paper alongside the competing approaches of the most active research groups in the area of EA management. The presentation was further illustrated alongside an application example taken form the EAMPC, which further publicized by Ernst in [19]. Although the approach promises to provide better guidance for managers faced with the challenge to design and develop an organization-specific EA management function, an evaluation of the utility of the proposed nexus in practice has not yet been undertaken.

We regard such an evaluation as greatly benefiting from a tool-based realization of the design theory nexus for situational EA management. Therefore, we are currently developing a tool, which enables managers in selecting and, if necessary, combining different EA management approaches. Nevertheless, as no standardized modeling language for method or model descriptions in the context of EA management has yet been developed, such a tool needs to be capable to integrate constituents from different approaches, which are described in different languages – a challenge not yet addressed.

Furthermore, the literature survey, which forms the basis for the development of the design theory nexus for situational EA management, needs to be critically reviewed. The selection of approaches is based on the activity of academic research groups in the field. Besides the question, if all relevant academic groups were included in the survey, also active communities of practitioners exist in the area of EA management. These communities promote and evolve widely-known approaches in the area (e.g. The Open Group Architecture Framework in [48]). In order to ensure usability and generality of the approach, the design theory nexus should be enhance with an extension mechanism to include further approaches to EA management, based on changing EA management goals or newly identified organizational constraints.

References

[1] S. Aier, S. Buckl, U. Franke, B. Gleichauf, P. Johnson, P. Närman, C. M. Schweda, and J. Ullberg. A survival analysis of application life spans based on enterprise architecture models. In *3rd International Workshop on Enterprise Modelling and Information Systems Architectures*, pages 141–154, Ulm, Germany, 2009.

[2] S. Aier, S. Kurpjuweit, J. Saat, and R. Winter. Enterprise Architecture Design as an Engineering Discipline. *AIS Transactions on Enterprise Systems*, 1:36–43, 2009.

[3] S. Aier, S. Kurpjuweit, O. Schmitz, J. Schulz, A. Thomas, and R. Winter. An engineering approach to enterprise architecture design and its application at a financial service provider. In *Modellierung betrieblicher Informationssysteme (MobIS 2008) – Modellierung zwischen SOA und Compliance Management 27.-28. November 2008 Saarbrücken*, pages 115–130, 2008.

[4] S. Aier, C. Riege, and R. Winter. Unternehmensarchitektur – literaturüberblick stand der praxis. *Wirtschaftsinformatik*, 50(4):292–304, 2008.

[5] S. Aier and J. Schelp. A reassessment of enterprise architecture implementation. In *Trends in Enterprise Architecture Research*, pages 53–68, 2009.

[6] S. Beer. *The Heart of Enterprise*. John Wiley, New York, NY, USA, 1979.

[7] S. Beer. *Brain of the Firm*. John Wiley, New York, NY, USA, 2nd edition, 1981.

[8] S. Beer. *Diagnosing The System for Organisations*. John Wiley, New York, NY, USA, 1985.

[9] G. B. Bird. The business benefit of standards. *StandardView*, 6:76–80, 1998.

[10] G. A. Britton and J. Parker. An explication of the viable system model for project management. *Systems Practice*, 6(1):21–51, 1993.

[11] J. Brocklesby and S. Cummings. Designing a viable organization structure. *Long Range Planning*, 29(1):49–57, 1996.

[12] T. Bucher, R. Fischer, S. Kurpjuweit, and R. Winter. Analysis and application scenarios of enterprise architecture: An exploratory study. In *Tenth IEEE International Enterprise Distributed Object Computing Conference (EDOC 2006), 16-20 October 2006, Hong Kong, China, Workshops*, Washington, DC, USA, 2006. IEEE Computer Society.

[13] S. Buckl, A. M. Ernst, J. Lankes, and F. Matthes. Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008). Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany, 2008.

[14] S. Buckl, A. M. Ernst, J. Lankes, K. Schneider, and C. M. Schweda. A pattern based approach for constructing enterprise architecture management information models. In *Wirtschaftsinformatik 2007*, pages 145–162, Karlsruhe, Germany, 2007. Universitätsverlag Karlsruhe.

[15] S. Buckl, F. Matthes, and C. M. Schweda. A viable system perspective on enterprise architecture management. In *2009 IEEE International Conference on Systems, Man, and Cybernetics*, 2009.

[16] Chair for Informatics 19 (sebis), Technische Universität München. Eam pattern catalog wiki. http://eampc-wiki.systemcartography.info (cited 2010-02-25), 2010.

[17] F. S. de Boer, M. M. Bonsangue, J. Jacob, A. Stam, and L. van der Torre. Enterprise architecture analysis with xml. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS 2005)*, volume 8, page 222b, Los Alamitos, CA, USA, 2005. IEEE Computer Society Press.

[18] E. W. Deming. Out of the crisis, 1982.

[19] A. Ernst. Enterprise architecture management patterns. In *PLoP 08: Proceedings of the Pattern Languages of Programs Conference 2008*, Nashville, USA, 2008.

[20] R. Espejo and R. Harnden. *The Viable System Model: Interpretations and Applications of Stafford Beer's VSM*. John Wiley, Chichester, UK, 1989.

[21] U. Frank. Multi-perspective enterprise modeling (memo) – conceptual framework and modeling languages. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 2002)*, pages 1258–1267, Washington, DC, USA, 2002.

[22] U. Frank, D. Heise, H. Kattenstroth, and H. Schauer. Designing and utilising business indicator systems within enterprise models – outline of a method. In *Modellierung betrieblicher Informationssysteme (MobIS 2008) – Modellierung zwischen SOA und Compliance Management 27.-28. November 2008*, Saarbrücken, Germany, 2008.

[23] I. Hanschke. *Strategic IT Management – A Toolkit for Enterprise Architecture Management*. Springer, Berlin, Germany, 2010.

[24] K. Harmon. The "systems" nature of enterprise architecture. In *IEEE International Conference on Systems, Man and Cybernetics 2005*, pages 78–58, 2005.

[25] A. F. Harmsen. *Situational Method Engineering*. PhD thesis, University of Twente, Twente, The Netherlands, 1997.

[26] J. C. Henderson and N. Venkatraman. Strategic alignment: leveraging information technology for transforming organizations. *IBM Systems Journal*, 32(1):472–484, 1993.

[27] M.-E. Iacob and H. Jonkers. Quantitative analysis of enterprise architectures. In D. Konstantas, J.-P. Bourrires, M. Lonard, and N. Boudjlida, editors, *Interoperability of Enterprise Software and Applications*, pages 239–252, Geneva, Switzerland, 2006. Springer.

[28] S. G. International. The chaos report 2009. http://www.standishgroup.com/newsroom/chaos_2009.php (cited 2010-02-25), 2009.

[29] International Organization for Standardization. Iso/iec 42010:2007 systems and software engineering – recommended practice for architectural description of software-intensive systems, 2007.

[30] P. Johnson and M. Ekstedt. *Enterprise Architecture – Models and Analyses for Information Systems Decision Making*. Studentlitteratur, Pozkal, Poland, 2007.

[31] P. Johnson, L. Nordström, and R. Lagerström. Formalizing analysis of enterprise architecture. In *Interoperability for Enterprise Software and Applications Conference*, page 10, Bordeaux, France, 2006. Springer.

[32] S. Kurpjuweit and R. Winter. Viewpoint-based meta model engineering. In M. Reichert, S. Strecker, and K. Turowski, editors, *Enterprise Modelling and Information Systems Architectures – Concepts and Applications , Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'07), St. Goar, Germany, October 8-9, 2007*, LNI, pages 143–161, Bonn, Germany, 2007. Gesellschaft für Informatik.

[33] K. Langenberg and A. Wegmann. Enterprise architecture: What aspect is current research targeting? Technical report, Laboratory of Systemic Modeling, Ecole Polytechnique Fdrale de Lausanne, Lausanne, Switzerland, 2004.

[34] J. N. Luftman, P. R. Lewis, and S. H. Oldach. Transforming the enterprise: The alignment of business and information technology strategies. *IBM Systems Journal*, 32(1):198–221, 1993.

[35] F. Matthes, S. Buckl, J. Leitel, and C. M. Schweda. *Enterprise Architecture Management Tool Survey 2008*. Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany, 2008.

[36] J. A. McDermin. *Software engineer's reference book*. Butterworth Heinemann, Oxford, Great Britain, 1994.

[37] K. D. Niemann. *From Enterprise Architecture to IT Governance – Elements of Effective IT Management*. Vieweg+Teubner, Wiesbaden, Germany, 2006.

[38] J. Noble. Classifying relationships between object-oriented design patterns. In *Australian Software Engineering Conference (ASWEC)*, pages 98–107, Los Alamitos, CA, USA, 1998. IEEE Computer Society.

[39] J. Pries-Heje and R. Baskerville. The design theory nexus. *MIS Quarterly*, 32(4):731–755, 2008.

[40] H. W. J. Rittel and M. M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4(2):155–169, June 1973.

[41] J. W. Ross. Creating a strategic it architecture competency: Learning in stages. *MIS Quarterly Executive*, 2(1), 2003.

[42] W. Ross, Jeanne, P. Weill, and C. Robertson, David. *Enterprise Architecture as Strategy*. Harvard Business School Press, Boston, MA, USA, 2006.

[43] A.-W. Scheer. *ARIS – Modellierungsmethoden, Metamodelle, Anwendungen*. Springer, Berlin, Germany, 4^{th} edition, 2001.

[44] J. Schelp and R. Winter. 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 (DESRIST2008), May 7-9, 2008, Westin, Buckhead, Atlanta, Georgia, USA,*, pages 79–92, Georgia State University, Atlanta, Georgia, USA, 2008.

[45] M. Schönherr. Towards a common terminology in the discipline of enterprise architecture. In S. Aier, P. Johnson, and J. Schelp, editors, *Pre-Proceedings of the 3^{rd} Workshop on Trends in Enterprise Architecture Research*, pages 107–123, Sydney, Australia, 2008.

[46] M. Schönherr. Towards a common terminology in the discipline of enterprise architecture. In G. Feuerlicht and W. Lamersdorf, editors, *Service-Oriented Computing – ICSOC 2008 Workshops*, pages 400–413, Berlin, Heidelberg, Germany, 2009. Springer.

[47] W. A. Shewart. *Statistical Method from the Viewpoint of Quality Control*. Dover Publication, New York, NY, USA, 1986.

[48] The Open Group. TOGAF "Enterprise Edition" Version 9. http://www.togaf.org (cited 2010-02-25), 2009.

[49] M. van den Berg and M. van Steenbergen. *Building an Enterprise Architecture Practice – Tools, Tips, Best Practices, Ready-to-Use Insights*. Springer, Dordrecht, The Netherlands, 2006.

[50] B. van der Raadt and H. van Vliet. Designing the enterprise architecture function. In *4^{th} International Conference on the Quality of Software Architectures (QoSA2008)*, pages 103–118, Karlsruhe, Germany, 2008.

[51] R. Wagter, M. van den Berg, J. Luijpers, and M. van Steenbergen. *Dynamic Enterprise Architecture: How to Make IT Work*. John Wiley, 2005.

[52] A. Wegmann. The Systemic Enterprise Architecture Methodology (SEAM). Technical report, EPFL, 2002.

[53] R. Winter. Business engineering – betriebswirtschaftliche konstruktionslehre und ihre anwendung in der informationslogistik. In *Integrierte Informationslogistik*, pages 17–38, Berlin, Heidelberg, Germany, 2008. Springer.

[54] A. Wittenburg, F. Matthes, F. Fischer, and T. Hallermeier. Building an integrated IT governance platform at the BMW Group. *International Journal Business Process Integration and Management*, 2(4), 2007.

[55] E. Yu, M. Strohmaier, and X. Deng. Exploring intentional modeling and analysis for enterprise architecture. In *Proceedings of the EDOC 2006 Conference Workshop on Trends in Enterprise Architecture Research (TEAR 2006)*, pages 32–32, Hong Kong, 2006. IEEE Computer Society Press.

[56] J. A. Zachman. A framework for information systems architecture. *IBM Syst. J.*, 26(3):276–292, 1987.