# A Conceptual Framework for Enterprise Architecture Design

Sabine Buckl, Florian Matthes, Sascha Roth,
Christopher Schulz, Christian M. Schweda

Technische Universität München (TUM)
Chair for Informatics 19 (sebis)
Boltzmannstr. 3, 85748 Garching bei München, Germany
{buckls,matthes,rothsa,schulzc,schweda}@in.tum.de

**Abstract.** An ambiguous terminology as well as a lack of clarity prevail in information systems research when focusing on enterprise architecture (EA) and its corresponding management function. Sound definitions for key terms in the field of EA design, i.e. strategies, principles, and goals, are too often used interchangeably with slightly different meaning. Addressing this situation, the present article proposes a conceptual framework for EA design that covers the aforementioned terms and organizes them along two dimensions, namely their underlying EA conceptualization and their role in the design process. The framework is exemplified and mirrored against state-of-the-art literature in the realm of EA and the corresponding management function. Finally, the article sketches further research trends centering around the design of an EA.

**Key words:** enterprise architecture, enterprise architecture management, conceptual framework, strategy, goal, principle, standard

## 1 Introduction

In the past decade *enterprise architecture* (EA) and its corresponding management function have gained considerable attention from the academic and practical audience. With the main objective of the management function to align business and IT its management subject is the enterprise's architecture which according to the ISO/IEC 42010 (2007) [11], can be defined as "the fundamental organization of [the enterprise] embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution". Distinct states of an EA are typically developed during EA management, the *current state* describing the status quo, the *target state*, forming an envisioned long-term perspective, and intermediate *planned states*, representing the medium-term objectives to be achieved. Regarding both planned and target states, the business represents the main 'driver' of the organizational change whereas projects are the 'implementors'. In order to ensure a managed evolution (cf. Murer et al. in [15]), projects have to be evaluated according to the principles guiding an EA's design and evolution.

Albeit the frequent references to concepts as *principle, vision,* and *strategy* in recent EA management-related literature (see e.g. [6, 20, 23]), concise and tangible definitions as well as classifications are still missing. Focusing for example on the term EA principle, existing sources [21, 25] typically apply related terminologies in an ambiguous manner while highlighting the importance of principles in the same breath. Especially, the EA management community lacks clarity how EA principles relate to the also frequently used terms *vision* and *strategy* or the often synonymously used term *standard*. Furthermore, the relationship between the aforementioned terms and the *projects* realizing enterprise transformation is of interest. These terminological unclarities drive the two core research questions of this article:

– What are major concepts for the design of an enterprise architecture and how they are defined?
– How those concepts embed in an abstract design framework for EAs?

The present article seeks to answer these questions by providing a framework that both helps to define and to understand the interplay of these terms. The framework is based on the prefabrics of the OMG's Business Motivation Model [16], Simon's science of the artificial [19], and the work on language communities of Kamlah and Lorenzen in [13] (see Section 2). The elaborated structure is further linked to the approach of the managed evolution already evoked above (Section 3). Afterwards, the framework developed is used to revisit the state-of-the-art in EA management literature in Section 4. Based on the analysis results future research directions in the area of EA management are derived and discussed in the concluding Section 5.

## 2 Principles in the context of EA management

The management of the EA is an activity intended to evolve the architecture as well as to control the evolution thereof. In this sense EA management can be understood a design activity (cf. van der Raadt and van Vliet in [24]) targeting the enterprise in a comprehensive manner. Enterprise architects (*designers*), with a planned state (*end*) in mind, search for the *means* by which the EA will achieve those aims. As part of this search the architects develop different plan scenarios of the EA and evaluate these with respect to the achievement of the desired end. The design activity may thereby be understood as a purely 'mental' one operating on a *mental model* of the enterprise also incorporating the according *means-end*-relationships.

In [19] Simon calls for a more formal understanding of design involving an imperative style of logic. In particular he proposes to operationalize the means-end-relationships behind any design problem into logical statements relating

**command variables** describing objects (architecture elements) that may be changed by design activities,

**fixed parameters** describing architectural properties as well as environmental aspects that cannot be changed by design activities,

**constraints** limiting the space of changes that can be made by a design activity, and

a **utility function** evaluating a designed architecture in respect to the (experienced) utility for its stakeholders.

In above terms the search for the planned state to pursue may be reformulated as 'find values for the command variables fulfilling the given constraints in the context of the fixed parameters that they best satisfy the utility function'. For the field of EA management such reformulation may at first seem a worthless exercise, as finding and defining formal command variables may be regarded a highly sophisticated and not easy to accomplish task. Nevertheless, the core distinction between command variables, constraints, and a utility function can be beneficially applied to understand, distinguish, and relate concepts as principles, strategies, and goals.

Complementing above considerations, we briefly revisit the notion of the mental model as essential part of addressing design problems. Every enterprise architect (as designer of the EA) uses a mental model of the enterprise to plan and evaluate the corresponding design alternatives. This model covers a specific area-of-interest, i.e. *concern* in terms of the ISO/IEC 42010 (2007), in the overall architecture of the enterprise. In line with Buckl et al. (cf. [3]) such concern may on the one hand be identified with a specific *conceptualization* of the enterprise, i.e. with a problem- and designer-specific classification of relevant elements of the enterprise. On the other hand a concern commits to a specific *filter* determining which parts of the enterprise are considered relevant. Two mental models as employed by two enterprise architects may hence differ in respect to both the conceptualization, i.e. the classification of elements, and the filtering, i.e. the selection of elements. In order to form the basis for a collaborative EA management conducted by a group of enterprise architects and other EAstakeholders, these people have to agree on a shared conceptualization. They have to be in one *linguistic community* (in terms of Kamlah and Lorenzen [13]) in order to be able to communicate their architecture understanding to collaboratively design and evolve the EA.

In the context of EA design activities more formal conceptualizations of the enterprise are widely used to facilitate communication between different EA stakeholders. These conceptualizations are mostly reflected in corresponding EA modeling languages, more precisely their underlying *information models*[1]. These models are conceptual models committing to an agreed conceptualization (cf. Buckl et al. in [3]) of the EA or parts thereof as relevant in respect to important stakeholders and their design problems. These stakeholders form the corresponding linguistic community for which instantiations of the information model are accessible. In the light of the above discussions we devise a framework for distinguishing and relating the different concepts as discussed in Section 1. This

[1] The meta-models backing an EA modeling language are in line with Buckl et al. (cf. [2]) named information models here.

framework builds on two axes, namely one considering the level of operationalization (*mental conceptualization* vs. *information model*) and one distinguishing between *desired result*, *course of action*, *utility function*, and *constraints*. Figure 1 outlines this framework and shows how the different concepts can be related therein.

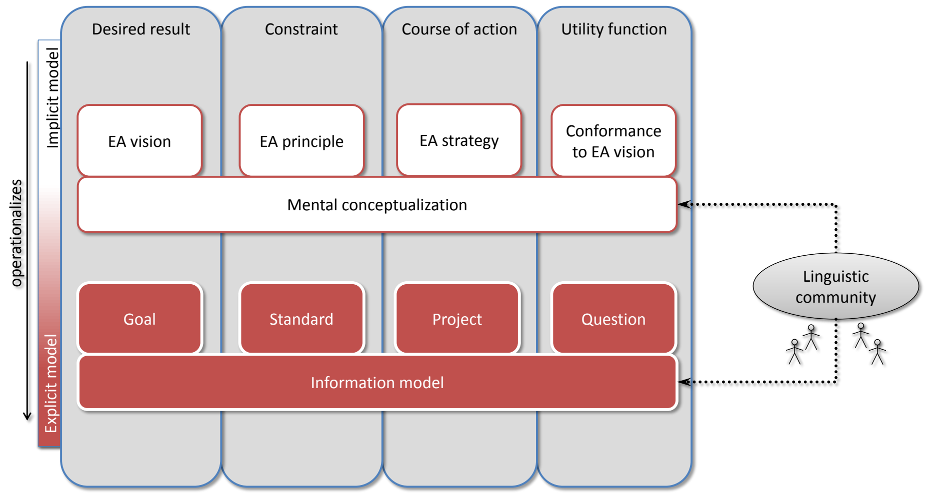


**Fig. 1.** Conceptual framework for EA design

Against the background of the framework we apply definitions for the different concepts as follows:

An **EA vision** is a distant target representing an ideal state, i.e. an implicit model and understanding of a target state of an EA. This aligns with the definition given by Lankhorst et al. in [14], saying that the "[EA] vision states an 'image of the future' and the values the enterprise holds".

An **EA principle** constrains and guides the design of the EA (cf. [18]) and may in turn provide justification for decision-making throughout an EA (cf. [22]). In general, principles are self-restraint and not externally obliged, e.g. by law in terms of compliances.

An **EA strategy** outlines a series of means (activities) to pursue a desired end, i.e. a dedicated target state of an EA. Refraining the complementary definition given in the OMG's Business Motivation Model (BMM) [16] a strategy is an "accepted [...] approach to achieve [desired ends], given the environmental constraints and risks".

**Conformance to EA vision** describes an intuitive understanding for the degree to which the current or a planned state of the EA matches the EA vision.

All the concepts mentioned before build on an informal and intuitive understanding as incorporated in the mental model of the corresponding linguistic community, e.g. the enterprise architects as designers of the EA. Refraining above argument on design in the EA context being a collaborative activity, these mental models, more precisely their backing conceptualizations, are frequently complemented with EA information models laying a formal basis for modeling the EA in an explicit manner. With the embracingness of the subject EA, it is sensible to assume that not the entire architecture of the enterprise but only relevant areas are covered by an information model. The decision on relevance may therein pertain to cover the areas-of-interest mirroring the visions, principles and strategies, but also to keep low the expenses for gathering and maintaining the information about the contained concepts. Put in other words, an information model has to balance the information needs and the expenses related to gathering this information. As consequence thereof, the mental concepts outlined above may only be partially operationalized to concrete pendants described below:

A **Goal** is a "statement about a state or condition of the enterprise to be brought about or sustained through appropriate means" (cf. [16]). In this sense, a goal describes an (intermediary) *end* operationalizing the EA vision or a part thereof, based on a corresponding information model. The goal is thereby "defined for an object [...] with respect to various models of quality [...]", i.e. is attached to a specific object in this model in terms of Basili et al. [1].

A **Standard** describes a "predefined design norm" based on principles and standards defined earlier (cf. [10]). In line with this argumentation standards describe a particular interpretation of an EA principle with respect to the information model.

A **Project** is concerned with the fulfillment of a goal. Consequently, any effort within a project should contribute to a goal, based on an information model. Thus, a project may be EA related in terms that it influences the EA, or an explicit EA transformation project.

A **Question** or a set thereof "is used to characterize the way the assessment/achievement of a specific goal is going to be performed" (cf. [1]). This means that a question measures the achievement of a goal based on the conceptualization mirrored in an information model.

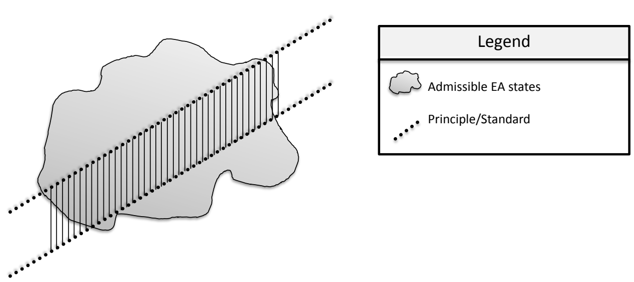


**Fig. 2.** EA principles constraining the EA design space

Both, principles and standards may be formulated in two different ways, namely as guidelines (respectively recommendations) or as restrictions (cf. [18]). In either way, a principle or a standard restricts the design space (cf. Figure 2), i.e. the set of accepted states, although the level of strictness may vary. Revisiting the relationships between a principle and the according standards one might say, that the standards operationalize the principle, while the latter provides the underlying rationale. The link between principle and vision is a more delicate one. While we in general can assume that the EA principles and EA vision do not contradict each other, i.e. that the vision is admissible against the background of the established principles, the same does not necessarily hold for standards and goals.

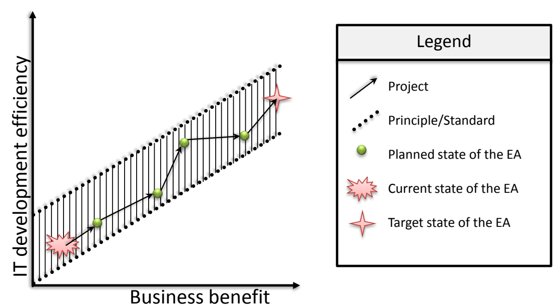


**Fig. 3.** EA principles in the paradigm of managed evolution

Figure 3 illustrates the role of EA principles within the paradigm of managed evolution [15] describing an iterative and controlled approach of EA development within multiple transformation projects. The EA evolves over time between the guidelines and constraints set by effective EA principles. In contrast to [15], we identified that these 'side rails' are manifested in EA principles as well as standards guiding on the one hand the evolution of the EA, and restricting on the other hand the possible EA states taken to evolve to an EAtarget state.

Over time, e.g. due to an increasing amount of technologies, the design space of EA management with regards to the set of admissible EA states widens. Since EA principles are static, this observation let us conclude the effectiveness of EA principles is degenerating on the long run. In conclusion, any applied standard, i.e. operationalized EA principle, has to generate feedback such that EA principles become dynamic and thus can evolve over time aligning to the needs of the EA evolution.

## 3 Exemplifying the approach

Making the above conceptual framework more explicit, an example leaning against Murer et al. [15] is provided in the this section. In Murer's example, a swiss bank has the EA vision (reflected in a target state of the EA) which supports the continuous implementation of new business requirements under high cost and time-to-market pressure. Furthermore, the EA vision states that IT systems must remain available, reliable, and secure as well as to assure effectiveness and economy of the business at the same time.

In our example, we assume that the bank has one important EA principle, namely to concentrate development competences of the IT departments. This principle is operationalized via a standard that defines a restricted set of programming languages to be used, namely Java, C++, and COBOL. With the EA vision of having an easily adaptable and flexible EA, the corresponding strategy is to transform the EA, more precisely the application landscape, into a service-oriented landscape that makes use of external services and service outsourcing where sensibly possible. This strategy has been pursued for some time by different projects that were able to implement service-orientation in certain parts of the overall EA.

At a certain point in time, the swiss bank decides to offer its customers an ‘app’ enabling mobile online banking which induces a positive influence on the bank’s distribution and communication channel. In this vein, business benefit also reflected in the bank’s overall reputation for cutting-edge technology is increased. With this short time goal set, resources for a new project are allocated targeting the development of such mobile application. Since the mobile application requires Objective C as the unique programming language, the above defined programming language standard is violated at the expense of short term business benefit. Figure 4 illustrates the impact of the ‘mobile app’-project on the overall state of the EA.

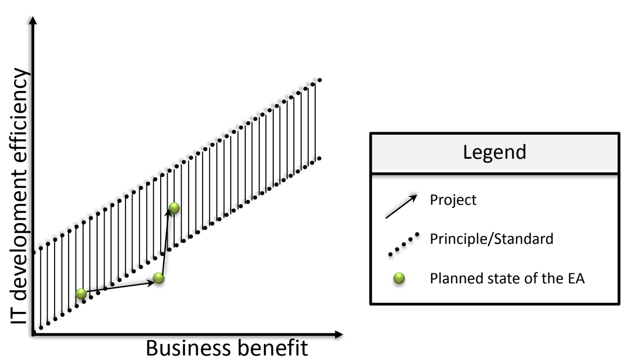


**Fig. 4.** EA evolution with a standard deviating project

After the successful completion of the project introducing a mobile application, a follow-up project is planned. This project replaces the mobile application developed in-house with a commercial of the shelf product leveraging the *Home Banking Computer Interface* (HBCI). In doing so, the application is reworked in a way that the actual development in Objective C is performed by a third party resulting in the conformance of the programming language standard.

## 4 Revisiting the state-of-the-art in EA management literature

In this section, six prominent approaches in the area of EA and the respective management function are discussed on the basis of the conceptual framework as presented in Section 2. Thereby, the terminology as employed in the original sources is used and compared to the concepts and understanding presented in this paper. Regarding the research method, the approach of hermeneutics (cf. [9]) is applied helping to interpret and map the identified concepts found in literature to the corresponding framework elements.

In his recent work [5, 6], Stelzer conducts an in-depth literature analysis on EA principles ranging from frameworks for EA and textbooks, to journals and conference publications. The author afterwards systematizes the investigated sources by decomposing an architecture principle into objects, context and description, scope of application, and its delimitation from other forms of principles. While he points out that there exists so far non-explained relationships between EA goals, principles, as well as means of their application, Stelzer refrains from providing a summarizing definition of these terms as well as from giving information how their relationship to each other is characterized. Notwithstanding, the author describes that architecture principles are embedded in a network of associated principles and highlights the fact, that their existence, differentiation, designation, and interrelation is contingent on the specific enterprise setting. Based on his profound literature analysis, Stelzer motivates to consider principles in a broader context and highlights that they are embedded in rationale and implications. However, by leaving out how these two concepts can be linked to an EA vision or EA strategy and by focusing on goals and EA principles, he refrains from explaining the broader context a principle is part of. Hence, additional concepts as proposed by the framework, e.g. different linguistic communities, projects, and questions are not examined by the author. Besides principles pertaining solely to the EA as indicated in both publications' titles (cf. [5, 6]), the author also implicitly refers to principles related to the management of an EA, IT, and business.

In *The Open Group Architecture Framework* (TOGAF) [21] is a prominent framework for developing an EA. Following the *Architecture Development Method* (ADM) which is subdivided into nine phases, an architecture vision describing "how the new capability will meet the business goals and strategic objectives and address the stakeholder concerns when implemented" is established in the first phase. Thereby TOGAF denotes an enterprise mission, vision, strategy, and goals as key elements of an architecture vision and points out that it provides "a first-cut, high-level description of the baseline and target Architectures, covering the business, data, application, and technology domains.". In addition, the vision also contains architecture principles which are defined as "general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an organization sets about fulfilling its mission". In differentiating between principles on enterprise-, IT-, and architecture level, the framework also proposes a specific format for defining architecture principles consisting of name, statement, rationale, and implications. Furthermore, architecture principles as subset of IT principles, are subdivided into a) principles that govern the architecture process and b) principles which govern the implementation of the architecture. Whereas former refer to the management of an EA, latter center around the guidance for designing and developing information systems. In this vein, the misleading term "architecture principles" in TOGAF rather pertains to main guidelines considering the management of an EA and not the elements of an EA itself. Furthermore, business principles, as a fourth type of principles are introduced separately and are linked uniquely to an architecture project, which also has to take current goals and strategic drivers into account. While TOGAF does not make any distinction between an implicit and explicit level, it also lacks in defining and creating concrete and consistent relationships among those concepts. For instance, the framework introduces the term blueprint as "collection of the actual standards and specifications" but leaves out to define the link regarding EA principles and strategy. Additional elements being part of the conceptual framework suggested above are not accounted for by TOGAF.

The approach of multi-perspective enterprise modeling (MEMO) of Frank [7] provides a framework consisting of a set of special purpose modeling languages, which can be leveraged to support different perspectives (strategy, organization, and information system) as well as various aspects relevant for enterprise modeling (EM). In their recent work, Frank et al. [8] elaborate requirements for a method aiming at the design and utilization of indicator systems within an enterprise environment. In leveraging the core concepts of SCORE-ML which have been embedded into the existing EM method supporting in turn the modeling of processes, resources, and goals, the authors associate indicators with the relevant business context. In the corresponding SCORE-ML meta model, indicators are directly related to goals which they are representing. Moreover, goals may be organized into a multi-layered hierarchy. Further concepts as proposed by the framework like strategy, principles, and visions of an EA as well as their operationalized counterparts (e.g. project, standard, and question) are neither considered by the SCORE-ML method nor embedded in the meta model.

Österle et al. [17] suggest a core-business-metamodel subsuming the main concepts of the strategy, organization, and system level. In depicting the relationships between concepts on these levels through a model which is restricted to the core elements of a business, the model supports the implementation of the St. Gallen's business engineering approach. The approach has been proven successful throughout more than 1000 consulting projects and is regularly reapplied for scientific publications (see e.g. [4]). Regarding the model in detail, goals as part of an enterprise's strategy are specified on a business level, although a strategy is only modeled implicitly in assigning distinct classes to the strategic level. Further concepts of above presented framework (e.g. EA principles, vision, project, and questions) are not in the scope of the model.

*Control Objectives for Information and related Technology* (COBIT) represents a prominent framework for IT management created by the Information Systems Audit and Control Association (ISACA) in close cooperation with the IT Governance Institute (ITGI) [12]. The framework is focused on what is required to achieve adequate management and control of IT, and is positioned on an abstract level. By being composed of four domains which are subdivided into 34 IT processes each consisting of a set of activities, COBIT provides a reference process model and common language for everyone in an enterprise to view and manage IT activities. In creating and explaining a linkage between an enterprise strategy, business goals for IT, IT goals, EA, and the IT scorecard, the framework emphasizes that an enterprise strategy influences the IT resources and capabilities, thus the EA. Furthermore, COBIT proposes an informal relationship between business-, IT-, process-, and activity goals which can be measured via the corresponding performance metrics driving in turn the respective higher-level goal. Although the framework states that suggested standards and good practices "are most useful when applied as a set of principles and as a starting point for tailoring specific procedures" COBIT does not include (EA) principles. Moreover, neither a formalized linkage between strategy and related goals is provided, nor a distinction between different levels of operationalization reflected in the different types of models is given. Additional concepts which are part of the presented framework (e.g. EA strategy, vision, project) are not addressed by COBIT.

A comprehensive picture of EA-related concepts including their definitions is provided by Schekkerman [18]. According to Schekkerman, a standard "is an agreement on how things should be done, or, in other words, a rule (or set of rules) on which an agreement exists". While a principle is defined on a higher level of abstraction by expressing "an idea, a message (culture / behavior) or value that comes from corporate vision, strategies, and business drivers, experience or from knowledge of a subject". Schekkerman also emphasizes, that principles (which similar to TOGAF consist of a name, statement, rationale, and implications) govern the business and organization as well as the development and implementation of the EA. However, the author does not consistently distinguish between the level of operationalization and means-end-relationships, as proposed by the framework presented in Section 2. In addition, Schekkerman speaks of a general vision and strategy referring to the main drivers and requirements of an enterprise but does not derive the according EA concepts and their pendants.

## 5 Future research questions

In this article, we devised a conceptual framework for EA design helping to overcome the ambiguity of terminology in the research field. In distinguishing between desired result, constraint, course of action, and utility function as well as an implicit and explicit model, we presented major concepts, defined them systematically, and finally sketched a framework into which these concepts embed. In doing so, the article leverages the prominent approach of the managed evolution of application landscapes in order to exemplify the elaborated framework.

As part of the article, the term linguistic community was introduced as a group which continuously communicates their architecture understanding to collaboratively design and evolve the EA. When taking a closer look on the framework elements being related to the implicit model, it becomes obvious that distinct groups share a different understanding resulting in deviating interpretations of the concept's instances. For example, the same EA principle can be operationalized by different corresponding standards taking the context like terminology, knowledge, and experience of the according linguistic community into consideration. Further research could focus on the manner how the different implicit model elements, i.e. EA vision, principle, strategy, and conformance to the vision, are made explicit depending on the community they are exposed to.

In the course of our literature review, we noticed that only few sources suggest utility function(s) in order to evaluate the fulfillment of the desired results given a specific set of constraints while following a certain course of action. Not only this type of functions could help to assess the actual goal achievement, it would also allow for a profound statistical analysis in addition to an internal and external comparison. Some approaches, like the one of COBIT [12], already motivate and propose textual questions targeting at the actual goal fulfillment, but fall short when it comes to the incorporation of principles and standards restricting the admissible EA states.

Regarding the various relationships among the framework's concepts, further research is required in order to refine the different interrelations. Thereby, one has to consider both, relations between concepts residing in the same as well as concepts being part of different models. In doing so, generic methodologies could be elaborated transforming elements of the implicit model to their explicit pendants. Furthermore, the incorporation of the framework's concepts within an explicit model longs for further investigation. For instance, the question arises how to easily integrate different EA standards representing guidelines and constraints in an existing information model by simultaneously striving for a generic and reusable approach.

Lastly, one has to take into consideration that during the EA design process each enterprise pursues several goals by also being bound to a variety of constraints at the same time. In most cases, different courses of actions can be taken while applying a different set of appropriate utility functions. From this perspective, multi-objective optimization coping with the increased complexity of the problem may deserve further investigations including the identification of an appropriate Pareto optimum.

## References

1. V. R. Basili, G. Caldiera, and H. D. Rombach. *The Goal Question Metric Approach*. Wiley, New York, 1994.
2. 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.
3. S. Buckl, S. Krell, and C. M. Schweda. A formal approach to architectural descriptions – refining the iso standard 42010. In *6$^{th}$ International Workshop on Cooperation & Interoperability – Architecture & Ontology (CIAO2010)*, 2010.
4. B. Dinter and R. Winter, editors. *Integrierte Informationslogistik (Business Engineering) (German Edition)*. Springer, Berlin, Heidelberg, Germany, 1st edition, 2008.
5. Dirk Stelzer. Enterprise Architecture Principles: Literature Review and Research Directions. In *4th Workshop on Trends in Enterprise Architecture Research (TEAR)*, Stockholm, Sweden, 2009.
6. Dirk Stelzer. Prinzipien für Unternehmensarchitekturen - Grundlagen und Systematisierung. In M. Schumann, L. M. Kolbe, M. H. Breitner, and A. Frerichs, editors, *Multikonferenz Wirtschaftsinformatik (MKWI 2010)*, pages 55–66, Göttingen, Germany, 2010.
7. U. Frank. Multi-perspective enterprise modeling (memo) – conceptual framework and modeling languages. In *Proceedings of the 35$^{th}$ Annual Hawaii International Conference on System Sciences (HICSS 2002)*, pages 1258–1267, Washington, DC, USA, 2002.
8. 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.
9. H.-G. Gadamer. *Wahrheit und Methode – Grundzüge einer philosophischen Hermeneutik*. J.C.B. Mohr, Tübingen, Germany, 3$^{rd}$ edition, 1975.
10. J. A. P. Hoogervorst. *Enterprise Governance and Enterprise Engineering (The Enterprise Engineering Series)*. Springer, Heidelberg, Germany, 1 edition, February 2009.
11. International Organization for Standardization. ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems, 2007.
12. IT Governance Institute. Framework Control Objectives Management Guidelines Maturity Models. `http://www.isaca.org/Knowledge-Center/cobit` (cited 2010-06-18), 2009.
13. W. Kamlah and P. Lorenzen. *Logische Propädeutik: Vorschule des vernünftigen Redens*. Metzler, Stuttgart, Germany, 3$^{rd}$ edition, 1996.
14. M. Lankhorst. *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, Berlin, Heidelberg, Germany, 2005.
15. S. Murer, C. Worms, and F. J. Furrer. Managed evolution. *Informatik Spektrum*, 31(6):537–547, 2008.
16. Object Management Group. Business Motivation Model 1.1. `http://www.omg.org/spec/BMM/1.1/` (cited 2010-06-15), 2010.

17. H. Österle, R. Winter, F. Hoening, S. Kurpjuweit, and P. Osl. Business Engineering: Core-Business-Metamodell. *Wisu – Das Wirtschaftsstudium*, 36(2):191–194, 2007.
18. J. Schekkerman. *Enterprise Architecture Good Practices Guide – How to Manage the Enterprise Architecture Practice*. Trafford Publishing, Victoria, BC, Canada, 2008.
19. H. A. Simon. *The Sciences of the Artificial*. MIT Press, Cambridge, Massachusetts, USA, 3rd edition, 1996.
20. K. Steininger, R. Riedl, F. Roithmayr, and P. Mertens. Moden und Trends in Wirtschaftsinformatik und Information Systems. *Wirtschaftsinformatik*, 51(6):478–495, 2009.
21. The Open Group. TOGAF "Enterprise Edition" Version 9. `http://www.togaf.org` (cited 2010-02-25), 2009.
22. The Open Group. *TOGAF Version 9 - A Manual*. Van Haren Publishing, 9th edition, 2009.
23. P. van Bommel, P. G. Buitenhuis, J. Stijn, S. J. B. A. Hoppenbrouwers, and E. H. Proper. Architecture Principles - A Regulative Perspective on Enterprise Architecture. In M. Reichert, S. Strecker, and K. Turowski, editors, *1st International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA)*, pages 47–60, Bonn, Germany, 2007.
24. B. van der Raadt and H. van Vliet. Designing the enterprise architecture function. In S. Becker, F. Plasil, and R. Reussner, editors, *4th International Conference on the Quality of Software Architectures (QoSA2008)*, volume 5281 of *Lecture Notes in Computer Science*, pages 103–118, Karlsruhe, Germany, 2008. Springer.
25. R. Winter and R. Fischer. Essential layers, artifacts, and dependencies of enterprise architecture. *Journal of Enterprise Architecture*, 3(2):7–18, 2007.