On the State-of-the-Art in Enterprise Architecture Management Literature

Sabine Buckl and Christian M. Schweda

June 1, 2011

Abstract

The enterprise architecture (EA) and its management are topics receiving ongoing interest from academia, practitioners, standardization bodies, and tool vendors. Over the last decade and especially in the last five years, much has been said and written on these topics that nevertheless have a much longer history dating back to the nineties of the last century. In these days, John Zachman was one of the first to understand the 'bigger whole' in which IS architecting and IS development is embedded. Ever since these days, the canonic knowledge on this topic, which would later become known as "EA management", has been furthered by many contributors originating from different philosophical, educational, and theoretical backgrounds, leading to numerous presentations and publications in this area. But while each article, paper or book extends the body of knowledge, it also 'raises the stakes' for anyone willing to enter this field of engagement. Especially, young researchers novel to this area that is not covered that much in university education than it perhaps should, find themselves confronted with a vast amount of 'hits', when they enter "EA management" as keyword in their favorite (scientific) search engine.

This report aims at charting the landscape of EA management research and practice. Applying a generic framework for structuring the body of knowledge in the field into the two core areas of "method" and "language", the work provides an overview on the state-of-the-art in the field, delineates interesting questions for future research, and shows how different approaches taken may be worthwhile subjects for researching how they may complement each other. Notwithstanding, subsequent work does not claim to cover *all* what has ever been written on the subject, but covers 22 different approaches from EA management communities from academic and industry based on over 150 sources ranging from technical reports to journal articles, from workshop papers to monographs. Dealing with each of the approaches not only on an abstract level but classifying it with a generic framework of the field and giving a summary of the approach's key achievements, the work provides a viewpoint balancing depth and breadth of the investigation.

Contents

List of Figures

List of Tables

CHAPTER 1

Motivation

Once upon a time information systems (IS) development has been both a profession for practitioners and a topic for academic research. John Zachman was yet to write what would later turn out to be the first and perhaps most well-known publication [Za87] on this subject, describing among others that IS are constituted of an organization and an information technology (IT) part. In his work Zachman emphasized a critical aspect of IS development, namely the IS architecture, which goes beyond a sole software architecture but has also to account for the organizational environment into which the information system is to be embedded. Over the years the complexity of organizations as well as the regulatory, economic, and technical environment that the organizations participate in, as well as the relevant IT support has changed dramatically calling for mechanisms to document, plan, analyze, and evolve – in short to manage – business and IT aspects in an integrated manner. At this time, Henderson and Venkatraman [HV93] presented the "strategic alignment model" establishing a linkage between strategic transformations of business and IT, but also relating the strategic propositions to infrastructure level architectures on both business and IT side. Back in those early days of IS development little was known that the Zachman Framework [Za87] and the strategic alignment model were harbingers of a new management discipline, whose subject is the enterprise as a whole. At nearly the same time as Henderson and Venkatraman, Spewak and Hill coined in [SH93] the term "enterprise architecture" that would turn out to be the name of choice for the embracing management subject of the dawning discipline, which is nowadays known as "enterprise architecture (EA) management".

Since these early days in the last millennium much more has been published on the topic of EA management and a plurality of research institutes, consultancies, and standardization bodies have made their stage appearances, not all of them being recurring actors in this environment. Notwithstanding many of them have contributed to the development of a discipline, which has ever since received increasing interest from manifold directions. In [LW04a], Langenberg and

Wegmann were the first to analyze the topic of EA management from a researcher's perspective, outlining among others the topics that research was heading at in 2004. The top ranking topic referred to in about 50% of the analyzed papers was "usage". These papers describing why EA management should and how EA management can be applied in organizational environments spoke volumes of the practical importance of the developing management discipline. Complementing this perspective Schönherr's more recent analysis on EA management-related literature [Sc08b] describes that despite the ongoing publication activities in this field, a common understanding of the research subject was in 2008 still yet to emerge. A similar position is taken by Aier et al. in [ARW08b], where the authors give an overview on different prominent EA management approaches further emphasizing on similarities and differences in their perspective on the subject. A key result of this analysis is the quite stable understanding of EA management being concerned with three types of states of the EA, namely a "current", several "planned", and a "target" state. This emphasizes both the documentation and planning aspect of EA management. In the same year, Schelp and Winter conducted in [SW08] an analysis on the scientific literature on the topic of EA management, diagnosing the formation of seven research groups in this field. These groups are elicited from over 94 publications, from which 'one-hit-wonders', i.e. sole publications from a single author, as well as approaches from practioner communities were excluded, as they "cannot be related consistently to any other publication" or are "considered weak regarding the definition of terminology, [... language] and/or methodology". Contrasting the findings of Schönherrs broader analysis, Schelp and Winter further describe a situation of "terminological convergence" between the different researchers in this field, going so far to foretell the advent of some kind of overarching "language community" spanning the seven research groups. Complementing their terminology-related analysis, another idea of their work is to classify the approaches in respect to the provision of a) a *procedure model* and b) a *description language*, as well as in respect to their coverage of the architectural levels 1) *strategy*, 2) *organization*, 3) *integration*, 4) *software*, and 5) *infrastructure*. This classification schema well serves as input for the analysis undertaken in this work, in which we pick up the central *method-language* dichotomy, but go into more detail both regarding method-related aspects like main activities of EA managementand configuration aspects and language-related aspects like temporal aspects of EA management.

The remainder of this work is structured as follows. Section 2 describes the method used to collect the papers, articles, and books that serve as input for the subsequent analyses. In this section, we further detail our understanding of "state-of-the-art" as reflected in the analysis method taken. Further, we come back to the method-language dichotomy and devise an analysis and classification framework for EA management approaches based on an understanding of EA management being a design process for the artifact EA. In this sense, generic design-related literature is taken as input to frame relevant characteristics that an approach may take. Section 3 revisits 22 EA management and EA management-related approaches as reflected in a total number of over 150 publications on the topic. Each approach is discussed against the background of the analysis framework and classified along the elicited characteristics. The final Section 4 summarizes the core findings of the analysis and gives an outlook on future research directions that can be derived thereof.

CHAPTER 2

Analysis design

This chapter details on the analysis method used to investigate the current state-of-the-art in EA management literature. Thereby, the literature review uses a systematic approach, which is based on the method of hermeneutic text comprehension as proposed by Gadamer in [Ga75] and follows the guidelines for literature reviews promoted by Webster and Watson in [WW02] (see Section 2.1). In line with Webster and Watson in [WW02, page xiv] and Bem in [Be95, page 174], we believe that "a coherent review emerges only from a coherent conceptual structuring of the topic itself". Therefore, relevant kernel theories (cf. Gregor in [Gr06]) from related disciplines are discussed in Section 2.2. Based on these kernel theories an analysis framework is developed, which spans the conceptual and cognitive background for the review synthesis.

2.1 The analysis method

Taking into account the characteristics of the IS discipline, Webster and Watson propose guidelines and hints for writing literature reviews in [WW02]. Among them are requests for

1. making the objectives of the review synthesis explicit (see Chapter 1),
2. discussing the scope and limits of the literature included in the article, and
3. providing a clear structure for the analysis (see Section 2.2.1 and Section 2.2.2).

Because of the amount of literature published in the context of EA management, discussing the scope and limits of the literature included in this report deserves special attention and is discussed subsequently.

Due to the increasing importance of the topic of EA management in recent years, the amount of literature published in this area proliferates (cf. study of Langenberg and Wegman in [LW04a] and the state-of-the-art overview of Mykhashchuk et al. [My11]). At the same no common understanding on the topic has evolved, leading to distinct research communities each forming a so-called language community. As EA management is a new discipline, for which different terms, e.g. strategic alignment (cf. Henderson and Venkatraman in [HV93]), IS architecture (cf. Zachman in [Za87]), or business IT alignment (cf. Luftman in [Lu03]) have been used in the past, before the term *enterprise architecture* was coined, the identification of relevant literature accordingly can be regarded a complex task, as existing databases, e.g. the web of science[1], the ACM digital library[2], or IEEE Explore[3] cannot be searched using a dedicated search string. In addition, research results concerning the topic of EA management are until now typically published as books in case of practitioners' experiences or presented on workshops (cf. Trends in Enterprise Architecture Management Research – TEAR, Practice-driven Research on Enterprise Transformation – PRET, or Workshop on Enterprise Architecture Challenges and Responses – WEACR) and therefore not included in scientific databases, which typically focus on journal publications. Therefore, we identified literature relevant for our synthesis by

- identifying research group via existing state-of-the-art analysis on EA management of Aier et al. in [ARW08b] as well as Schelp and Winter in [SW09],
- searching the DBLP[4] and the websites of the author for further publications,
- going backwards by reviewing the citations of the publications identified in the first two steps, and
- removing research groups, which have been cited sporadic or are not available in English.

Utilizing the above introduced method, we identified 22 research groups and individual authors with a publication record in the area of EA management, which provide input for our state-of-the-art review in Section 3. While practitioner frameworks like *The Open Group Architecture Framework (TOGAF)* [Th09a] or the *Integrated Architecture Framework (IAF)* [Wo10] are thereby included in our analysis, frameworks developed by governmental agencies, e.g. the Department of Defense Architecture Framework (DoDAF) in [De09a, De09b, De09c] and the NATO Architecture Framework (NAF) in [NA07] are not included. As these frameworks typically focus on the public sector and the intended audience of our subsequent review synthesis are practitioners and researchers in the are of EA management, we abstain from discussing these frameworks in detail.

To ensure a complete overview and correct evaluation of the included approaches, we contacted the main authors and asked them to provide feedback on the evaluation results. From the overall number of 22 persons that have been contacted, we received feedback regarding 17 approaches (77%). The feedback resulted in the inclusion of further papers and thesis from these groups and a second iteration regarding the review.

[1] See `http://www.webofscience.com`
[2] See `http://portal.acm.org`
[3] See `http://ieeexplore.ieee.org`
[4] See `http://www.informatik.uni-trier.de/~ley/db/`

2.2 An analysis framework

In line with the understanding of EA management as a methodical and language-based means to develop and evolve the overall architecture of an enterprise, we subsequently analyze different EA management-related approaches as outlined in scientific literature. The analysis is of a twofold nature reflecting the assumed method-language dichotomy, such that the first part of each analysis is dedicated to analyzing methodical prescriptions and indications as found in the description of the approach. The second part of each analysis in contrast is concerned with prescriptions on the language to be used in order to model the EA. Drawing from prefabrics in the field of management method or (architecture) modeling languages, respectively, we devise a framework guiding the in-depth analyses of the different approaches. The method-centric part of the framework is described in Section 2.2.1, whereas the language-centric part is detailed in Section 2.2.2. Summarizing Section 2.3 synthesizes the two perspectives and outlines an embracing analysis framework for EA management, which is used to analyze the state-of-the-art.

Beside the analysis of the method- as well as language-prescriptions made by the different approaches, each approach is shortly summarized in a fact sheet. In this sheet general information on the approach is provided, such as *name*, *issuing organization*, dedicated *tool support*, *period of activity* and the corresponding list of *publications*. Two more characteristics are described in the fact sheet (for an example see Table 2.2), namely the *inner organization* and the *focus area* of the approach. The embracing nature of the management subject as well as the corresponding function further influences the way the approach is presented in. Regarding each approach in itself as one (composite) artifact, the approaches may strongly differ in respect to their inner organization. In detail, an approach may be presented

- as one comprehensive *monolith* without apparent inner structure
- with an *explicit organization*, in which the components establish explicit links to each other, or
- with an *implicit organization*, where the components are grounded in a unified and linking terminology.

While the design process of EA management in its nature exerts the aforementioned method-language dichotomy, not every approach is likely to put similar emphasis on both aspects. The focus area categorization reflects this fact.

EA management or EA management-related approach	
Name of approach:	
Issuing organization:	
Focus area:	
Tool support:	
Period of activity:	
Publications:	
Inner organization:	

Table 2.1: Exemplary fact sheet

2.2.1 A method-centric analysis framework for EA management

EA management represents itself as a typical management function, which according to Deming [De82a] and Shewhart [Sh86] is typically organized in a cycle (so-called *PDCA cycle*) containing the following phases: *Plan – do – check – act*. Put in the context of EA management, the plan phase covers the development of architectural descriptions, i.e. current, planned, and target states of the EA. The do phase is reflected in the context of EA management by means of communicating and enacting EA artifacts and plans. Various ways to perform such an enacting exists, ranging from fairly non-interfering ways of informing and communicating via enacting, e.g. rewarding, to enforcing, e.g. punishment. The *check* phase of Deming and Shewhart is reflected by methods and means to analyze and evaluate parts of the EA. Thereby, an analysis can be performed to evaluate one state of the EA, e.g. the current state to identify potential for improvement as well as the achievement of objectives. Similarly, planned states can be analyzed and evaluated regarding their strategic impact of the transformation planned. Furthermore, analyses and evaluations can be performed in order to compare two states of the EA, e.g. a delta analysis between the current state and an planned state or between an planned state and the target state can be performed. Complementing the management cycle of Deming in [De82a] and Shewhart [Sh86], an assessment and improvement of the EA management function itself has to be performed in the act phase. Therefore, the overall performance of the EA management function is assessed and the different methods and means used are adapted to better suit the EA management goals and the organizational context.

Besides the area of management in general as discussed above further related research topics exist, from which analysis criteria for existing EA management approaches can be derived. Cognition and comprehension of the subject are further evolved by discussing different perspectives on the design of an EA management function as proposed by different research communities, which opt for approaching the topic of EA management from a

- *knowledge management* perspective, e.g.. Buckl et al. in [BMS09a] and Struck et al. in [St10b],
- *systemic* perspective, e.g. Buckl et al. in [BMS09b], Wegmann in [We02], or Pulkinnen in [Pu06], and
- *situational engineering* perspective, e.g. Harmsen et al. in [HPK09], Riege and Aier in [RA09], or Leppännen in [LVP07].

Based on the perspectives on designing an organization-specific EA management function as introduced above, analysis dimensions focusing on the method part of EA management are motivated in the following. Furthermore, distinct characteristic types of EA management approaches are introduced.

Analyzing integration

EA management does not exist as an isolated management function in an organization but is embedded into the context of other enterprise-level management functions as project portfolio or strategy management (cf. *system two* in the systemic perspective of Buckl et al. [BMS09b]).

The successful management of the EA is in this sense inevitably connected with linking these management functions. More precisely an EA management function in order to be effective must exchange management-relevant information with other enterprise-level management functions on a common basis. From this perspective, we classify the existing approaches more precisely their contained methods according to their level of *integration* in:

- approaches that do not provide mechanisms (*no mechanisms*) for integration. An EA management function according to such an approach does not account for information exchange with other management functions, neither concerning the content of information exchanged nor concerning tasks or triggers for the exchange.
- approaches that provide *unidirectional* integration mechanisms. An EA management function according to such an approach describes the exchanged information, i.e. the content, as well as the corresponding information source (management function). Information on triggers and tasks may optionally be supplied. Contrariwise, the EA management function is limited to one direction of exchange either being a receiver (sink) for external information or a sender (source) of information.
- approaches that provide *bidirectional* integration mechanisms. An EA management function according to such an approach describes the exchanged information as well the originating or targeted management function, i.e. acts as receiver (sink) as well as sender (source) each in at least one case. Information on triggers and tasks may also be supplied.

Analyzing develop & describe

Central prerequisite to EA management are means useful for understanding the EA in an abstract and problem-specific manner, i.e. making the complexity of the EA manageable via models. Such means are EA descriptions and plans, i.e. EA models of various types. The activity *develop & describe*, which corresponds to the *plan* phase of the PDCA-cycle, comprises different tasks aiming to create EA models of different architectural states as well as representations of the principles that guide future evolution of the EA by constraining the solution space (cf. Simon in [Si96]). Finally, the activity targets the concretization and documentation of questions as utility functions that apply on different architectural states. Further detailed against the different objects of representation and modeling, we classify the methods provided by the different approaches as follows:

- approaches describing the *current* state. An EA management function building on such an approach describes tasks and steps for documenting the status quo of the EA and optionally denotes the actors involved in such activities.
- approaches developing a *planned* state. An EA Management function committing to such an approach describes tasks and steps to be taken for developing planned states for an EA from projects, more precisely the architectural changes performed by these projects. Optionally such EA management function may describe responsibilities in these tasks by denoting actors involved therein.

- approaches developing a *target state*. An EA management function based on such an approach describes tasks and steps for developing target states of an EA, i.e. to formulate architecture visions. Optionally these steps may describe how the target state can be derived from the strategies of the enterprise, namely the business and the IT strategy. Further, the approach may describe responsibilities for these task.
- approaches developing EA *principles*. Approaches of that kind describe tasks and steps that can be taken to devise organization-specific development guidelines and to document these guidelines, e.g. via standards. Optionally these steps delineate how the principles may be derived from strategic input, as e.g. the business or IT strategy.
- approaches developing EA-relevant *questions*. An EA management function building on such an approach describes tasks and steps to commit a set of EA-relevant questions, i.e. methods for agreeing on an understanding of 'better' and 'worse' in respect to EAs. These questions may be formulated on a fairly abstract level.

Analyzing communicate & enact

The importance of communication as activity in EA management is frequently discussed in literature (cf. Lankhorst in [La05, 67–82] or Schekkerman in [Sc06d, page 88]), leading to the conclusion that an approach for EA management must cover the topic of *communicate & enact*, especially as the communicate & enact activity in the context of EA management can be mapped to the *do* phase of the PDCA cycle. In this sense we analyze, whether the approach describes steps to be taken and tasks to be performed in order to communicate EA related information to the corresponding stakeholders. Complementing the communicative nature of EA management, we further analyze, if the approach delineates tasks that may be applied to govern projects as the implementors of organizational change and enterprise-level management functions according to EA plans, visions, and principles. Detailed onto the level of the different classification this means:

- approaches communicating the *current* state. Approaches of that kind describe steps and tasks for communicating the status quo of the EA, or resort to the provision of visualizations together with a statement on the corresponding stakeholders.
- approaches communicating and enacting *planned* states. An EA management function building on such an approach describes tasks and steps for communicating planned states, or delineates visualizations and their corresponding stakeholders. Further, tasks and steps for enforcing architecture plans in related management processes may be given.
- approaches communicating a *target* state. An EA management function committing to such an approach describes tasks and steps for communicating target states of the EA, or describes visualizations for doing so as well as the corresponding stakeholders.
- approaches communicating and enacting EA *principles*. For the communication of principles an approach should describe steps and tasks or provide structured templates for communicating principles together with information on the intended audience thereof. Enactment mechanisms for principles, i.e. via dedicated steps in planning functions like quality gates are further described.

- approaches communicating EA-relevant *questions*. Approaches of that kind describe uniform templates for communicating questions and link these to the relevant stakeholders. Instead of doing so the approaches may delineate steps and tasks for communicating putting special emphasis on the informed stakeholders.

Analyzing analyze & evaluate

In the course of creating future, i.e. planned and target, states of the EA different alternatives for implementation may be developed and have to be analyzed in order to make an informed decision. Mapping the *check* phase of the PDCA cycle to the EA management context, a comprehensive approach must in this respect cover methods and responsibilities concerned with analyzing architecture states and plans as well as for comparing different states of the EA. Regarding the corresponding state, we classify the approaches to:

- approaches analyzing the *current* state. An EA management function building on such approach describes steps and tasks to be taken in order to (collaboratively) analyze the status quo of the EA in respect to given goals and principles. In this context the stakeholders of the corresponding analyses may be denoted and responsibilities for performing the analyses may be specified.
- approaches analyzing *planned* states. An EA management function building on such an approach describes which steps and tasks are necessary for analyzing planned states and may optionally specify the addressees of the analyses as well as the responsible actors. Goal- as well as principle-based analyses are expected mechanisms here.
- approaches analyzing a *target* state. For analyzing a target state especially expert-based analysis techniques are to be described. Such techniques, more precisely the steps and tasks preformed therein, are necessary to evaluate a target state in respect to principles and goals.
- approaches performing comparative analyses (*delta analysis*) targeting two states. Approaches of that kind provide steps and tasks for comparing different EA states highlighting the corresponding differences and similarities. Comparisons between current and target states, planned and target states, as well as between different planned states are of interest here.

Analyzing configure

An EA management function is an organization-specific artifact, i.e. has to be *configured* to fit into the *organizational context* as well as the intended *scope and reach*, i.e. the goals pursued. As proposed by the situational engineering perspective, each method should be "tailored and tuned to a particular situation" [Ha97, page 25]. Speaking more precisely, not any kind of implementing a management activity is suited in every context and for every intension, such that an EA management approach supporting its configuration must supply mechanisms to specifically design an EA management function in respect to the goals pursued and the organizational context, which embeds the management function. With the distinction between context on the one hand and scope and reach, i.e. goals pursued, on the other hand, we classify each approach as follows:

- approaches providing *no mechanisms* for configuration. Approaches of that kind do not regard EA management as organization-specific or make prescriptions on an abstract level abstaining from organizational implementation.
- approaches providing mechanisms to configure the EA management function to the *organizational context*. Configurable approaches of this type delineate organizational contexts, e.g. management structures, that are beneficial or detrimental for some of the provided management methods. Put in other words, these approaches describe common organizational contexts and link them to tasks, steps, and responsibilities.
- approaches providing mechanisms to configure to *scope and reach*. Approaches that are configurable to an organization-specific scope and reach, i.e. the goals pursued by the EA management endeavor link tasks, steps, and responsibilities to the specific management goals that are considered helpful for pursuing.

Analyzing adapt

Complementing the PDCA-cycle, the *act* phase has to be mapped to the EA management context. With the ongoing change of the organization itself as well as its environment, the EA management function may need to be *adapted* as well. Further, the need for adaptation may arise from the successful implementation of such function in the enterprise, which calls for an increased reach of the function. In the latter sense the adaptation reflects an increased level of maturity in EA management[5]. According to the provided mechanisms, we classify EA management approaches as follows:

- approaches providing *no mechanisms* for adaptation. Approaches of that kind make no or only abstract level prescriptions on how to react to changes in the organizational context or on how to adapt to a changed scope an reach. Especially these approaches do not describe how to transform an already implemented EA management function to an adapted one.
- approaches providing mechanisms to adapt to the *organizational context*. Adaptable approaches of this type delineate organizational context changes and describe transformations for implemented management tasks, steps, or responsibilities.
- approaches providing mechanisms to adapt to *scope and reach*. Approaches of that kind describe transformations for increasing and reducing the reach of the EA management function, e.g. by delineating how certain tasks and steps can be extended to relate to other enterprise-level management functions. Concerning the scope, such approaches describe mechanisms to perform the one-time change of the scope, i.e. detail transformation methods encompassing documentation, communication, and analysis 'one-timers'.

[5]See Szyszka in [Sz09] for in depth discussion on maturity models in the context of EA management.

2.2.2 A language-centric analysis framework for 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 as a design activity (cf. van der Raadt and van Vliet in [RV08]) targeting the enterprise in a comprehensive manner. Therefore, a common language to describe the EA, i.e. the management subject, has to be developed. Enterprise architects (*designers*), with a planned state (*end*) in mind, search for the *means* by which the EA will achieve these ends. As part of this search the architects develop different plan scenarios of the EA, i.e. architectural descriptions, 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 organization also incorporating the according *means-end*-relationships [Bu10b]. In [Si96] 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'. In line with the above argumentation on the benefits of different perspectives on methodological aspects of EA management, we propose a multi-perspective approach to analyze existing approaches in respect to the proposed languages. Besides the engineering perspective as introduced above, we propose to take an

- *model* perspective (cf. Stachowiak in [St73]),
- *ontological* perspective (cf. Dietz in [Di06]),
- *managed evolution* perspective (cf. Murer et al. in [MWF08]),
- *quality (goal – question – metric)* perspective (cf. Basili et al. in [BCR94]), and
- *situational engineering* perspective (cf. Harmsen et al. in [HPK09], Riege and Aier in [RA09], or Leppännen in [LVP07]).

Subsequently, we motivate an analysis framework focusing on language-aspects consisting of different dimensions for classifying existing EA management approaches. Thereby, the above identified perspectives on designing an organization-specific EA management function are used to derive the dimensions and motivate the distinct characteristic types.

Analyzing blackbox coverage

The organization in itself is designed to support the corporate objective, i.e. to deliver relevant business capabilities to its customers. In order to manage the organization's structure, i.e. the EA, a model, more precisely an architectural description has to be developed. During the conceptualization, i.e. model creation, the coverage of elements is limited to those relevant in respect to the objective of the EA management initiative (cf. reduction characteristic of Stachowiak in [St73]). The notion of purposefulness recurs on the different level of the enterprise and its corresponding architecture (cf. Winter and Fischer in [WF06]). From a blackbox perspective each of these architectural levels is seen as entity providing dedicated capabilities and services for use by other architectural levels or customers. In this sense, the blackbox perspective may be regarded as "functional" decomposition of the enterprise (cf. Dietz in [Di06]) reflected on three layers as follows:

- *business & organization* layer. An approach targeting this layer from a blackbox perspective defines concepts for a functional view on the business, most notably business capabilities but also business contracts and elements to specify the corporate objective.
- *application & information* layer. The blackbox perspective on this layer encompasses concepts as business services describing the support functions provided by the information systems as well as corresponding operating-level agreements. These concepts may be identified with concepts of a service oriented architecture (SOA).
- *infrastructure & data* layer. An approach covering a blackbox perspective on this layer describes technical service provision as storage and network transmission, and corresponding operating-level agreements. The corresponding concepts may partially be identified with those prevalent in a virtualization perspective on hardware devices.

Analyzing whitebox coverage

The provision of support for the corporate objective is achieved in the complex and delicate interplay of multiple components that together with their interconnections shape the overall structure of the EA. This structure forms the focal point of the whitebox perspective, which can be applied on different architectural levels. The whitebox perspective may be regarded as "structural" decomposition of the enterprise (cf. Dietz in [Di06]) in mirrored on three EA layers as follows:

- *business & organization* layer. An approach targeting this layer from a whitebox perspective covers concepts for describing the structure of the business via business processes and business functions, but also via organizational units.
- *application & information* layer. The whitebox perspective on this layer encompasses concepts as business applications and interfaces but may also detail to application components.
- *infrastructure & data* layer. An approach covering a whitebox perspective on this layer describes technical devices that make up the IT infrastructure of a using organization, as hardware and network devices.

Analyzing projects coverage

Strategies describe a coarse evolution path for the organization targeting architectural elements on all different layers. According to the idea of managed evolution (cf. Murer in [MWF08]), this path is subsequently concretized via *projects*, i.e. the means in terms of Simon [Si96] that act as implementors of organizational change. This reflects a broad understanding of the term "project" not bound to activities with a short-term project nature but also to ongoing maintenance activities. In this sense, the project coverage reflects a change perspective on the enterprise covering both transformation and evolution aspects on three EA layers as follows:

- *business & organization* layer. An approach targeting projects on business and organization layer supplies means to describe e.g. organizational restructuring and business process reengineering.
- *application & information* layer. Projects to be accounted for on this architectural layer typically are software development projects but also standard software introduction and customization projects.
- *infrastructure & data* layer. An approach covering projects on infrastructure layer encompasses concepts that describe infrastructure transformations, e.g. hardware replacements, virtualization projects, or data migrations.

Analyzing goals coverage

Complementing the projects as "means" of organizational change, a comprehensive language for modeling EAs must also account for describing EA-relevant *goals*, i.e. desired "ends" (cf. Simon in [Si96] that should be achieved. Following the GQM approach of Basili et al. [BCR94], the language must supply concepts to specify an intended state of architectural elements on any layer of the architecture:

- *business & organization* layer. An approach targeting goals on business and organization layer allows to describe business goals, stating intended qualities, e.g. throughput, of the business processes and capabilities, as well as organizational goals making prescriptions on organizational qualities, e.g. number of employees.
- *application & information* layer. Goals to be accounted for on this architectural layer may target the complexity of the structures as well as availability, latency or throughput of the provided services.
- *infrastructure & data* layer. An approach covering goals on infrastructure layer allows to make prescriptions on infrastructural complexity, network latency, hardware or infrastructure service availability.

Analyzing principles coverage

Principles delineate design constraints that are imposed on the projects and limit the admissible development paths for the EA or parts thereof (cf. constraints as proposed by Simon in [Si96]). In order to reflect this in EA models a comprehensive EA description language must allow to specify restrictions and impose standards on all different architectural layers:

- *business & organization* layer. An approach covering principles on business layer allows to prescribe standard business processes and capabilities that the organization seeks to provide.
- *application & information* layer. Principles to be accounted for on this layer make prescriptions on application architectures and interface standards that should be used.
- *infrastructure & data* layer. An approach targeting principles on infrastructure and data layer allows to specify e.g. which infrastructure technologies should be used in realizing the organization's EA infrastructure.

Analyzing questions coverage

EA management as a design function targeting the enterprise in an embracing manner has to supply techniques for measuring success and goal achievement, i.e. supporting a utility function in terms of Simon in [Si96]. With the description language for the EA providing the basis for any kind of comprehensive description, the language is required to cover measurement techniques (*questions* cf. [BCR94]) as well and allow them to be linked to the corresponding measurement objects, i.e. the architectural elements, on all different architectural layers:

- *business & organization* layer. Questions on this architectural layer may target throughput, availability, and latency of business processes and provided capabilities.
- *application & information* layer. Questions to be accounted for on this architectural layer may target the complexity of the structures as well as availability, latency, or throughput of the provided services.
- *infrastructure & data* layer. Relevant questions on infrastructure layer target infrastructural complexity, hardware, or infrastructure service availability.

Analyzing configure & adapt

The EA description language has to mirror the organization-specificity of the EA management function, i.e. has to be *configured* and *adapted* to fit the intended scope and reach (cf. appropriateness for a specific situation in situational engineering). This specifically means that a using organization must be allowed to select the concepts appropriate for the specific utilization context. Further, mechanisms for adapting the description language with changing scope and reach of the EA management function should be supplied. Against that background, we classify the approaches as follows:

- approaches providing *no mechanisms* for configuration. Approaches of that kind do not regard EA management as organization-specific or make prescriptions on an abstract level abstaining from language details.
- approaches providing mechanisms to *configure* the language. Approaches that may configure their scope and reach link language concepts to the specific management goals that they may be helpful for pursuing.
- approaches providing mechanisms to *evolve* the language. Approaches that may adapt their scope and reach describe mechanisms to develop an existing language into the direction of a changed scope.

2.3 Integrating method and language perspective

At first sight it may seem that method and language classification frameworks are largely unconnected, but having a closer look on the categories and characteristics this turns out to be a misconception. In the following, we shall explore the linkages between the two perspectives showing that and how they are tightly interconnected. A central connex between the two frameworks grounds in the understanding of EA management as design activity. In detail this means that EA management is concerned with planning the transformation from a current state EA via projects and associated planned states towards the long-term target state EA.

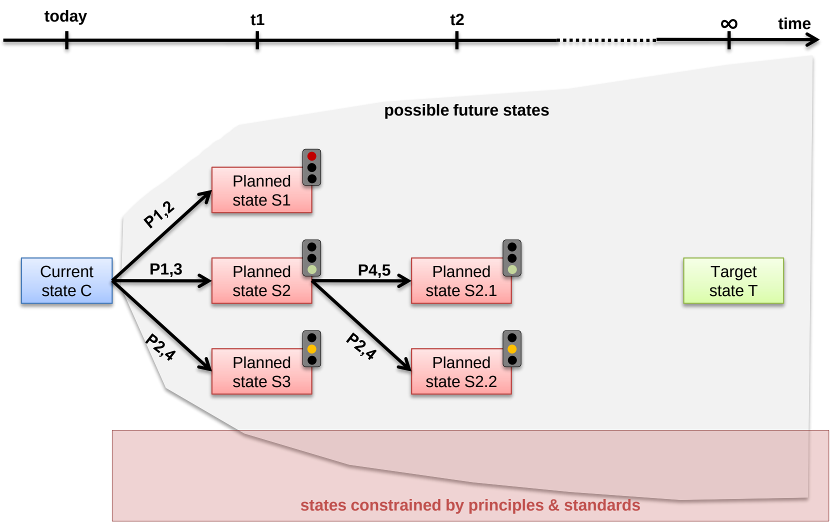

Figure 2.1: Developing the EA over different states according to Buckl et al. [Bu09e]

Figure 2.1 illustrates this fact displaying the relationships between the three types of architecture states (current, planned and target), the projects that implement the architecture changes, the principles and standards guiding the change and constraining the admissible states, and finally the questions that reflect organization-specific utility functions for evaluating EA states (via the traffic lights). In the sense of the method framework (cf. Section 2.2.1) each of the three types of states has to be documented (current) or developed (planned and target), respectively. The same holds for the principles that guide the architectural change, constraining the admissible future states, and the questions used to evaluate the "utility" of an intended EA. Thereby, a broad understanding of "utility" is applied, where an EA is considered 'util', when it satisfies the requirements of its stakeholders. Having documented and developed these concepts, the method framework further discusses how the architecture states, the principles and questions can be communicated or enacted (planned state, principles). Complementing this, the states are further analyzed and difference between the states (delta analysis) are elicited.

Switching to the terminology of the language framework (cf. Section 2.2.2), current states and target states become white-box, black-box and goal category descriptions. Thereby, the distinction between black-box and white-box allows to specify architectural states on different levels of detail, whereas goals are used to define intended target states on both black-box and white-box perspective. The planned states, being closely linked to their implementing projects, are reflected by the project characteristic, whereas the constraining principles have their own language framework characteristic. Finally, the utility function of design is reified via questions that may affect arbitrary architectural layers ranging from business to infrastructure. In this sense, the different types of artifacts to be developed, communicated and analyzed according to the prescriptions of the method framework are reflected in language framework artifacts, which themselves may span the EA in differing depth. Figure 2.2 illustrates the linkage between the two frameworks as established by the common artifacts.

Analyze & evaluate	current	planned	target	delta analysis	
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
	White- & Black-box	Projects	Goals	Principles	Questions
	business & organization	business & organization	business & organization	business & organization	business & organization
	application & information	application & information	application & information	application & information	application & information
	infrastructure & data	infrastructure & data	infrastructure & data	infrastructure & data	infrastructure & data

Figure 2.2: Linkage between the method and language framework

2.4 Classifying the state-of-the-art analysis

In 2006, Fettke conducted a state-of-the-art analysis of the state-of-the-art in the German-speaking IS community (cf. Fettke in [Fe06]). Thereby, he identified a characterization framework for literature reviews [Fe06, page 259]. This framework is used in the following to summarize the scientific methodology the review results are based on. According to Fettke, two different types of reviews can be distinguished – natural language and statistical reviews. Our research synthesis emphasizes on the natural language characteristic. While each review has a distinct focal points, e.g. results, research method, theory, and experience, we focus on the theories presented by the different approaches. In line with Fettke in [Fe06, page 265], we opt for making the objective of the review explicit.

Type		natural language		mathematic-statistical	
Focus		research results	research method	theory	experience
Target	Formulation	not explicit		explicit	
	Content	integration	criticism	central topics	
Perspective		neutral		position	
Literature	Selection	not explicit		explicit	
	Extensiveness	foundations	representative	selective	complete
Structure		historical	thematically	methodical	
Target group		common public	practitioners	common researcher	specialized researcher
Future research		not explicit		explicit	

Table 2.2: Characterization of the review synthesis presented in this chapter according to Fettke in [Fe06, page 259]

The aim of the state-of-the-art analysis is detailed in the motivating Chapter 1 and can be interpreted as a combination of integrating existing approaches via the framework derived from the kernel theories and at the same time providing a critical reflection of the thereby made contributions. In order to do so, a neutral perspective is chosen. In line with the guidelines of Webster and Watson in [WW02, page xv] and the argumentation of Fettke in [Fe06, page 265], we made the boundaries of our work explicit and made the criteria how the literature was selected transparent. In this vein, we provide a comprehensive overview of existing EA management approaches, although proof for complete coverage is unfeasible to give. According to Fettke, a review can be structured historically, thematically, or methodically. In our review we used the historic structuring, resulting as a side effect in an overview how long the different research groups have been active in this area. Although we focus on the scientific community as prospective readers, we believe that the topic has a strong relation to industry, therefore, the results address researchers in general as well as practitioners. Table 2.2 summarizes the classification of this review synthesis according to the framework developed by Fettke in [Fe06].

CHAPTER 3

Revisiting the state of the art

3.1 The Zachman Framework

EA management approach	
Name of approach:	Zachman Framework
Issuing organization:	Zachman Institute
Focus area:	Modeling
Tool support:	-
Period of activity:	since 1987
Publications:	[Za87], [SZ92]
Inner organization:	monolith

Zachman developed what was initially (cf. [Za87]) called a "framework for information systems architecture" and has ever since broadened its scope to the perhaps most well-known framework for EA – the Zachman Framework. In its most recent version[1] the framework presents five modeling layers and six dimensions mirrored by corresponding interrogative pronouns. The modeling layers are "scope", "business", "logical systems", "technical systems", and "detailed representations", whereas the latter are according to Zachman not in the context of EA management. On these different layers, the central questions of "what", "how", "where", "who", "when", and "why" apply. Figure 3.1[2] outlines the structure of the Zachman Framework. Putting the interrogative pronouns together, the core question that the Zachman framework associates with the EA is:

> Who does what in which way (how), when and where? Why does he do it?

[1] An overview on the framework is available online at `http://www.zifa.com/framework.pdf`. The recent version was accessed on February, 16^{th}, 2010.
[2] See `http://zachmanframeworkassociates.com/index.php/`

This core question is answered on each of the layers with increasing level of detail, i.e. the conceptualizations used in answering the questions become increasingly fine-grained. Illustrating this fact along the question of "who", major business divisions (on the scope level) are decomposed and operationalized to organizational units (on the business level), to roles (on the logical system level), and finally to users (on the technical system level).

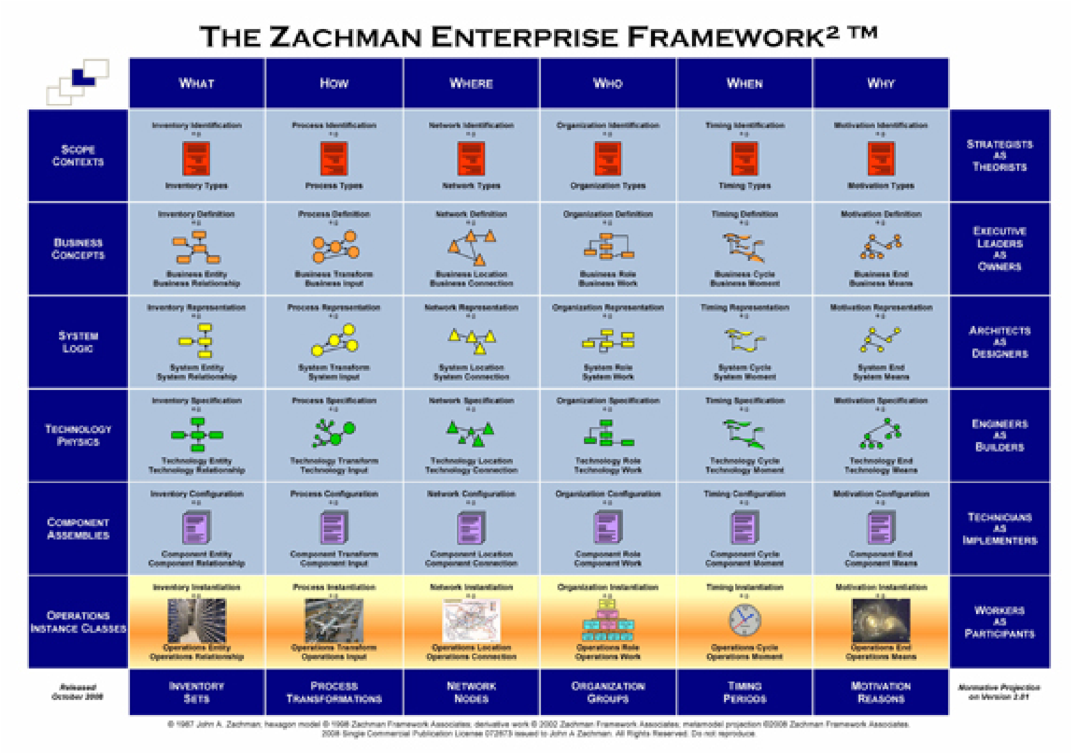

Figure 3.1: Two dimensional schema behind the Zachman Framework

From a methodic point of view the Zachman framework does not make detailed descriptions, i.e. no management activities or tasks are described. This aligns with the understanding of Zachman as an EA framework and not as EA management framework. Consequently, only some minor method-related information is put forward by the framework, outlining that the framework may both be applied in describing the state of the EA as well as in describing requirements for a future EA, i.e. in developing a target state [Za87]. Additionally, the framework gives several remarks on the importance of transformation activities in order to get from the current to a future state, thus highlighting the importance of planning processes, although more detailed information on how to plan EAs are not directly given. With respect to the communication of information corresponding to the framework's prescriptions, Zachman delineates in [Za87, pages 282–284] the variety of purposes that such architecture descriptions may serve as well as the plurality of addressed stakeholders, e.g. business owners or information system designers. All this aligns with the basic notion of the framework understanding itself as structuring principle to be used in information system architecture development activities in order to get an embracing perspective, whereas no methodical integration points are discussed but a flexible utilization of the framework is advocated for. In line with the requirements put forward in Section 2.2.1, we classify the Zachman Framework as shown in Table 3.1.

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.1: Method classification for the Zachman Framework

Central conceptualization of the Zachman framework with respect to the language used for describing architectures is the "thing-relationship-thing" paradigm. This conceptualization is adapted to the specific perspective taken in respect to the selected dimension to "input-output-input" in the "how"-perspective, "entity-relationship-entity" in the "what"-perspective, and "node-line-node" in the "where"-perspective (cf. Zachman [Za87, page 283]). In the initial work, the application of the paradigm on the remaining dimensions is only briefly sketched in an appendix providing concretizations as "organization-reporting-organization" for the "who"-perspective, "event-cycle-event" for the "when"-perspective, and "objective-precedent-objective" for the "why"-perspective (see [Za87, page 292]). Zachman further emphasizes that each of these perspectives is unique, i.e. that the corresponding descriptions are different even though they "may pertain to the same object and therefore are inextricably related to one another". Another type of relationship between different perspectives exists regarding the modeling layer (scope) of the description. These relationships are further detailed by Sowa and Zachman in [SZ92, pages 592,603–605], explaining that a higher-layer model should not be derivable from the information contained in the according lower-layer models. Sowa and Zachman further discuss the recursive application of the framework in an enterprise context, delineating that beyond an enterprise-wide level, the framework also applies for the level of individual "products" (information systems). These abstract prescriptions are complemented with a critical distinction refraining the design nature of the Zachman framework as reflected by two "versions" that a framework user should create, namely an "as-is" and a "to-be" version of the architecture. Complementing the abstract paradigm "thing-relationship-thing", Sowa and Zachman introduce a graphical notation in [SZ92, page 607], the so-called "conceptual graph" that in conjunction with quantors from predicate calculus is used to describe concrete situations from the paradigm's perspective. In this vein, the conceptual graphs complement the "detailed cell metamodel" given in the same work [SZ92, pages 594–595], which concretizes the paradigm on business, information systems and technology layer taking the how, what and where perspective simultaneously. For each layer this leads to an integrated white-box meta-model introducing concepts as "business entity", "business process", and "business location" as well as their corresponding counterparts on the other layers. In line with the uniqueness requirement regarding the models on the different layers, the integrated meta-models as represented in their types and relationships are structurally equivalent, whereas no one-to-one mapping between the corresponding instantiations can be expected to exist. According to the explanations complementing the meta-model, all concepts contained therein may be subject to transformations during architecture development, although concrete language facilities supporting the description of transformational activities are not supplied with the framework. This leads to a limited coverage of the requirements (cf. Section 2.2.2) by the Zachman Framework as shown in Table 3.2, whereas many of the not fulfilled requirements are discussed there in some detail but not complemented with language elements.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.2: Language classification for the Zachman Framework

3.2 Architecture of Integrated Information Systems (ARIS)

EA management approach	
Name of approach:	Architecture of Integrated Information Systems (ARIS)
Issuing organization:	University of Saarbrücken, IDS Scheer AG
Focus area:	Modeling
Tool support:	ARIS Toolset
Period of activity:	since 1992
Publications:	[KNS92], [Ki99], [Sc01], [Sc02]
Inner organization:	monolith

The ARIS approach is a framework for holistic modeling of business information systems, targeting the development of such systems from a process-based perspective (cf. Scheer [Sc01, Sc02]). This in particular is mirrored by the overall method of the ARIS approach being very close to a 'classic' software development consisting of the sequence "requirements elicitation", "design specification" and "implementation description". The waterfall-like method is nevertheless not executed once, but applied on the different views[3] that pertain to a business information system.

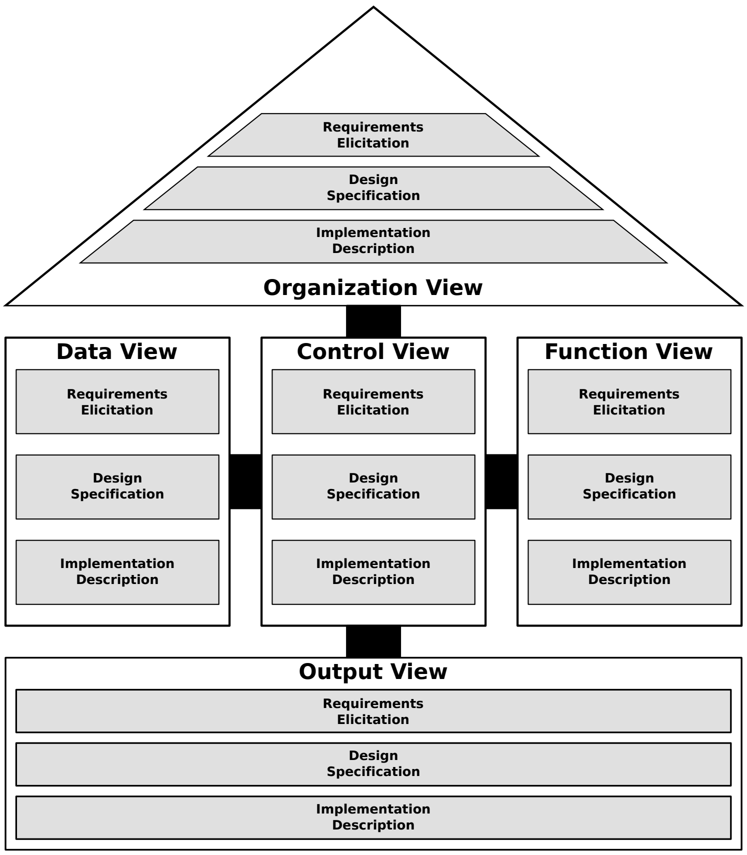

Figure 3.2: ARIS house

[3]In line with the terminology used in this report, the "views" would correctly be alluded to as "viewpoints".

The so-called “ARIS house” (see Figure 3.2) introduces these views as follows:

- *organization view* describing the structure of the organization together with the lines of authority and the communication channels in the organization,
- *data view* describing the business data objects created, manipulated, and exchanged between the business functions
- *function view* describing the business functions that are to be executed by the organization as part of its value proposition b
- *output view* describing the values, goods, and services delivered by the organization in executing its business functions
- *control view* interlinking the other views from a process-oriented point-of-view describing the business processes that are executed by organizations, work on business data objects, support the business functions, and are involved in delivering the goods and services

Beyond the waterfall-like software development method the ARIS approach does not directly provide methodical guidance for working with the modeling language. Extending the basic ARIS approach, Kirsch [Ki99] develops a method for “process-oriented management of client-server-systems”. Central to this approach is a iterative development method consisting of the phases “plan”, “realize”, and “apply and control”. During the different phases of the method, process-oriented analysis models are created and refined towards implementation models, which are fed to implementation. Kirsch further emphasizes related processes, as e.g. the one of release management that are to be supplied with information from the ARIS models. Nevertheless, the method-related guidelines of the ARIS approach are strongly focused on developing models, such that the overall approach may be classified as shown in Table 3.3.

<table>
<tr><td>INTEGRATION</td><td colspan="2">unidirectional</td><td colspan="3">bidirectional</td></tr>
<tr><td>DEVELOP & DESCRIBE</td><td>current</td><td>planned</td><td>target</td><td>principle</td><td>question</td></tr>
<tr><td>COMMUNICATE & ENACT</td><td>current</td><td>planned</td><td>target</td><td>principle</td><td>question</td></tr>
<tr><td>ANALYZE & EVALUATE</td><td>current</td><td>planned</td><td>target</td><td colspan="2">delta analysis</td></tr>
<tr><td>CONFIGURE TO</td><td colspan="2">organizational context</td><td colspan="3">scope and reach</td></tr>
<tr><td>ADAPT TO</td><td colspan="2">organizational context</td><td colspan="3">scope and reach</td></tr>
</table>

Table 3.3: Method classification for ARIS approach

With the ARIS approach being centered around the question of modeling business information system especially from a process-oriented point-of-view, the meta-language of ARIS introduces manifold process-related concepts. From a strategic perspective, business processes are further distinguished into support processes and core processes, for which the model further supplies techniques to link to corresponding reference process models. The business functions of the organization are linked to the organization’s goals and are distinguished with respect to the level of IT support provided. In a similar sense, business data objects are refined with respect to their IT involvement and the services provided by the organization are distinguished along their information-relatedness. Linking together these different perspectives on the organization, the concepts in a control view decompose the business processes into function and events, of which the former are executed by organizational units and create business relevant output. Figure 3.3 gives an overview on the meta-model of the ARIS approach against the background of the viewpoints provided by the ARIS house.

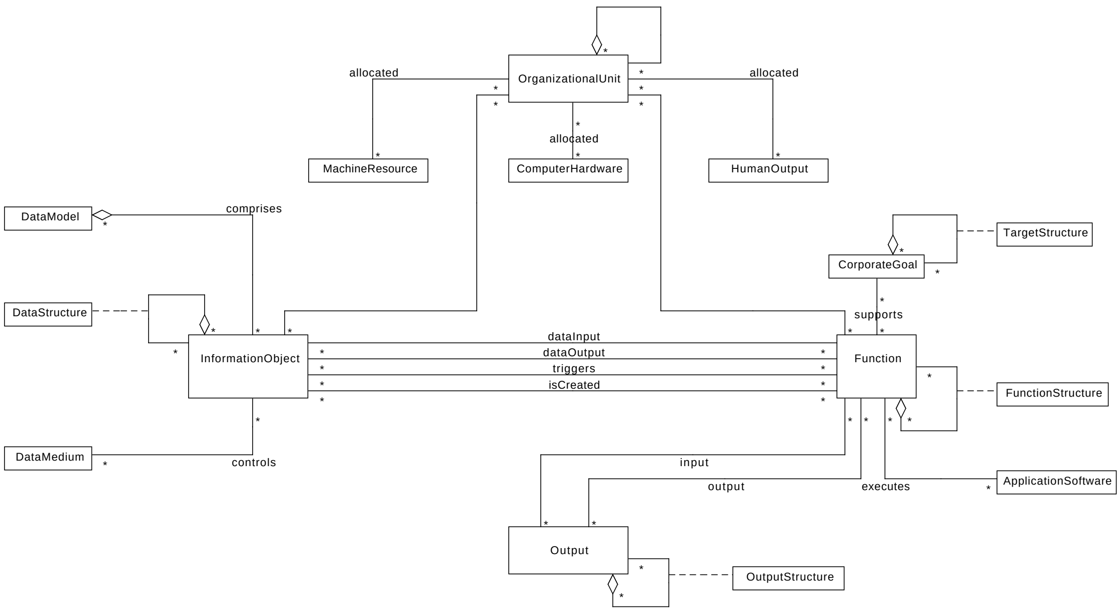

Figure 3.3: Overview of the meta-model of the ARIS approach

The event-function dichotomy of a control view is reflected in a corresponding modeling method, namely the one of the "event-driven process chain" introduced by Keller et al. in [KNS92]. An event-driven process chain details the structure of events and functions with additional "operators" that may be used to denote splits, joins, and decisions in the process execution. For each concept introduced in the ARIS meta-model, the ARIS approach further supplies a unique symbolic representation, making up the well-known and colorful appearance of the event-driven process chains and their extended versions. In the light of the aforementioned modeling capabilities, we classify the language aspects of the ARIS approach as shown in Table 3.4.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.4: Language classification for ARIS approach

3.3 The Integrated Architecture Framework (IAF)

EA management approach	
Name of approach:	Integrated Architecture Framework
Issuing organization:	Capgemini
Focus area:	Modeling
Tool support:	-
Period of activity:	since 1993
Publications:	[Wo10]
Inner organization:	monolith

The integrated architecture framework (IAF) has a long history with its early days dating back to a program named “Snowball” (cf. [Wo10]). This program was initiated by the consultancy Capgemini in response to the need for new IS delivery methods that should replace the outdated waterfall model. Over several iterations, the new method eventually become the core of what was then named the IAF centering around an architecture development method[4]. Ever since that time the focus of the architecture framework was broadened to account for distributed IS and secure IS as well as their underlying technology infrastructure. In 1997 another aspect, namely development governance was added to the framework, which has been further detailed with a content framework in 2000. This content framework specifies the architectural concepts and deliverables that are to be accounted for in architecture development. Finally, in the year 2008 IAF has qualified as framework for The Open Group's architect certification program (ITAC)[5].

IAF sees itself as part of a larger context of EA-related activities that form an iterative process as shown in Figure 3.4. Receiving input from both business and IT strategy management, IAF is concerned with “translating” this input to a target architecture. According to van't Wout et al. [Wo10] this translation is “the architect's profession”. In line with this argumentation IAF does not concern itself with migration planning, whose task would be to derive a roadmap for implementing the target architecture. Moreover, as van't Wout et al. put it in [Wo10, page 9]. IAF “tries to avoid overlap with other professions like business analysis”. This ‘narrow and pragmatic focus’ of the framework can be regarded as attempt to keep responsibilities clear, while techniques subsumed under “using IAF outcomes by non-architects” (cf. [Wo10, pages 218–223]) give clear indications how IAF deliverables are handed over to related management functions. In particular, IAF sketches that architecture assessments may be used to make business cases in program management, or delineates that both functional and non-functional requirements crucial for IS development are related to a contextualizing architecture understanding. Nevertheless, in-depth recommendations and guidelines on how to establish and maintain these links to related management functions are not provided.

For the architecture management process, called “architecting”, in terms of van't Wout et al. [Wo10], IAF does not provide actual prescriptions or best-practices as the process is expected to be different in every “engagement”, i.e. project and organization. Nevertheless, the

[4] This method is not to be mixed up with the method of the same name provided by The Open Group Architecture Framework. IAF's method is focused in developing IS architectures.

[5] For additional information on the program see `http://www.opengroup.org/itac/`.

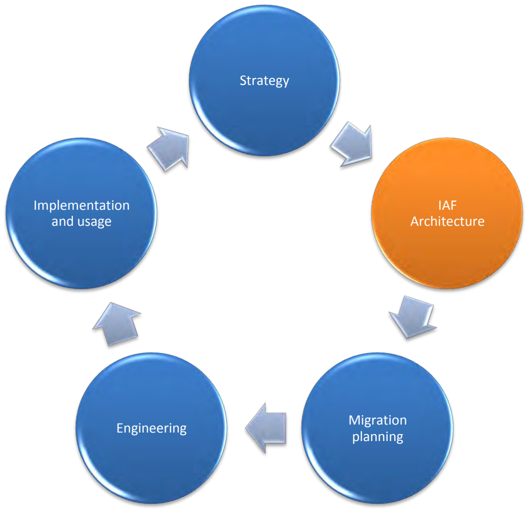

Figure 3.4: Context of the IAF architecture function according to van't Wout et al. [Wo10]

framework defines a large number of contextual factors that influence the process of architecting. Aside from strategic factors, i.e. input from upstream strategy development functions, knowledge about the organization's competitors, its operating model and its culture as well as of the key stakeholders and key objectives of architecting is deemed indispensable in order to devise the most appropriate architecture management process. Complementing former list, van't Wout et al. [Wo10, pages 41–50] further name the "architecture scope" as crucial factor determining how to architect, although the remarks on how to account for a specific contextual setting are fairly general and abstract. Concerning the documentation of the architecture on the three layers "business", "information system" and "technology", IAF provides strongly artifact-centric guidelines and methods. For example, van't Wout et al. give in [Wo10, pages 55–59] some indications on how the business layer may be documented starting with any of the relevant concepts, as business object, business role, business goal, business event, or business process. Similar indications may also be found for the two more architectural layers, although the methods remain highly abstract. This nevertheless well aligns with the self-image of the framework, as being an "artifact framework", i.e. an EA framework oriented towards architecture modeling. According to van't Wout et al. [Wo10, pages 152–169] IAF may well be complemented with other EA management frameworks, most notably ones delineating governance and architecting methods. Complementing the abstract understanding of architecting activities, IAF elaborates on the crucial topic of the "design authority", i.e. of the organizational responsibility to align developments with the strategy. With the linking role of architecting, different formats of authority may be handed to the enterprise architects. For example, architects may be have the opportunity to give advice to project teams, may participate in a formal project review process in decision making, or may be empowered to veto projects (cf. [Wo10, pages 207–209]). In this sense, different models of communicating and enacting architectures via differing design authorities are discussed in the framework, contributing to an overall evaluation of its method-related prescriptions as shown in Table 3.5.

Van't Wout et al. describe in [Wo10, pages 19–21] the general structure of what they call the "IAF content". The content is structured along three dimensions, of which the first is con-

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.5: Method classification for the Integrated Architecture Framework

cerned with the level of abstraction ranging from "why" over "what" and "how" to "with what". The second dimension partitions the architecture along different "aspect areas": "business", "information", "information systems" and "technology infrastructure". Finally, a third dimension complements the aforementioned structural EA aspects with two relevant cross-cutting topics "security" and "governance". Figure 3.5 depicts the three dimensions outlining the IAF content.

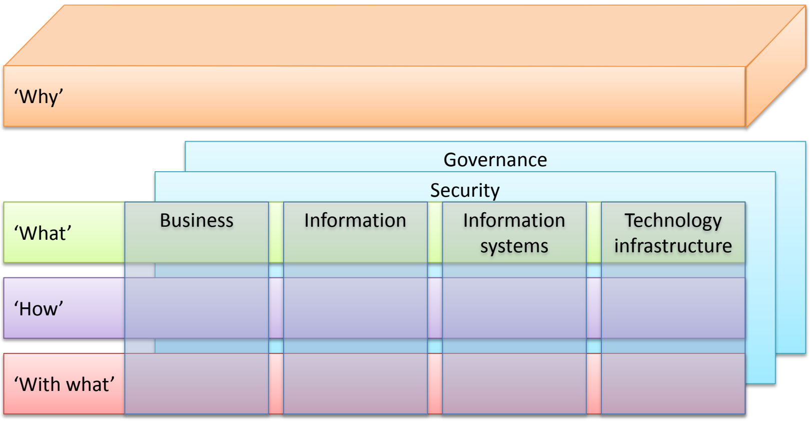

Figure 3.5: Three dimensions of the IAF content according to van't Wout et al. [Wo10]

Going into the details of the IAF content, van't Wout et al. describe the relevant concepts for each of the aspect areas. For example, the concepts "business goal", "business role", "business activity", "business event", "business object", "business object contract", "business service" and "business domain". Each of these concepts is introduced an complemented with a textual definition of its semantics and indications on the relevant properties as well as relationships. In addition to the definition of concepts, exemplary viewpoints (named "views" in the context of IAF) are described and possible utilization scenarios therefore are outlined. Having elaborated on the different architectural aspects of relevance, a cross-cutting topic concludes the exposition of the IAF content. Van't Wout et al. discuss in [Wo10, pages 138–150] different perspectives on the quality of an architecture. In particular, they propose quality characteristics that may be applied on different elements of the overall architecture, differing the implementation of measurement procedures for these characteristics to the using organization. With its strong account for the organization-specific substance of architecting, the IAF introduces the idea of the "IAF roadmap" describing possible ways for evolving the IAF content in an organization. Put in other words, any such roadmap describes a specific way of introducing IAF into an organization. For each of these roadmaps a rationale as well as the needed organizational context is delineated, and expected benefits as well as possible liabilities are described. Exemplifying the idea of roadmaps, we subsequently summarize the "infrastructure focused roadmap" as described by van't Wout et al. in [Wo10, pages 215–216]. After starting with documenting the "why" layer, both business and information system aspects are only described from a blackbox point of view ("what"), delving into the very details of the technology aspect. According to IAF such roadmap is especially appropriate, when technical topics as virtualization or cloud computing should be incorporated into a stable business environment. A possible liability of the roadmap on the other hand is that both business and information systems are only briefly considered, leaving room and risks for errors. While these roadmaps are intensionally designed to support the configuration of the content, i.e. description language, for architecting, they may nevertheless also be used to evolve an existing EA management function. Summarizing, due to the scope of IAF centering around the defining architectures, its language-related prescriptions may be classified as shown in Table 3.6.

BLACK-BOX PERSPECTIVE	business & organization	application & information	infrastructure & data
WHITE-BOX PERSPECTIVE	business & organization	application & information	infrastructure & data
STRATEGIES & PROJECTS	business & organization	application & information	infrastructure & data
VISIONS & GOALS	business & organization	application & information	infrastructure & data
PRINCIPLES & STANDARDS	business & organization	application & information	infrastructure & data
QUESTIONS & KPIs	business & organization	application & information	infrastructure & data
CONFIGURE & ADAPT	initially	evolutionary	

Table 3.6: Language classification for the Integrated Architecture Framework

3.4 Enterprise Architecture Planning (EAP)

EA management approach	
Name of approach:	Enterprise Architecture Planning (EAP)
Issuing organization:	-
Focus area:	method
Tool support:	-
Period of activity:	since 1993
Publications:	[SH93]
Inner organization:	monolith

In the book *Enterprise Architecture Planning – Developing a Blueprint for Data, Application, and Technology* [SH93], Spewak and Hill present a pragmatic approach to EA management focusing on the top two layers of the Zachman Framework (see Section 3.1) [SH93, page 13]. According to Spewak and Hill in [SH93, page 1] Enterprise Architecture Planning (EAP) is "the process of *defining architectures* for the use of information in support of the business and the *plan* for implementing those architectures". In this sense, Spewak and Hill propose a method for EAP emphasizing on planning aspects. The main objective of EAP according to Spewak and Hill is to "provide quality data to those who need it" [SH93, page 15]. Based on the 14 points for quality of Deming in [De82b] they derive the 14 points of data quality, e.g. management commitment, manage data as asset, develop measures, which represent the foundation that should be agreed upon prior to each EAP-related endeavor (cf [SH93, pages 5–6]).

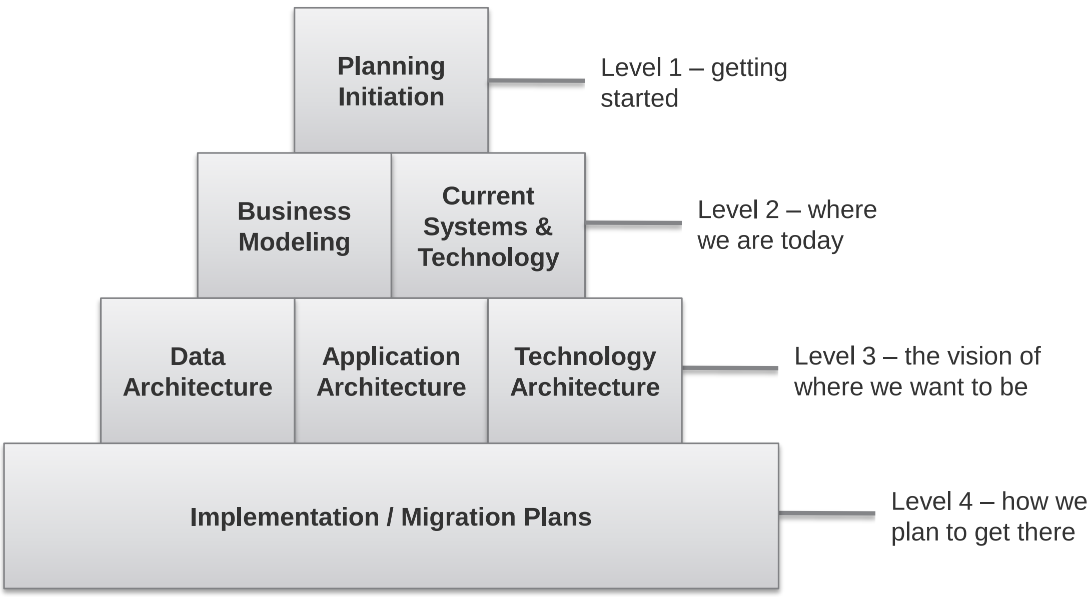

Figure 3.6: Phases and levels of EAP according to Spewak and Hill in [SH93, page 16]

The EAP method consists of seven phases, which are organized in four layers with each layer representing a different focus or question – where we start, where we are today, where we want to be in the future, and how we get there – of task as illustrated in Figure 3.6 (cf. Spewak and Hill in [SH93, pages 14–15]). For each phase, Spewak and Hill detail on the steps to be performed, deliverables to be created, utilized procedures, and supporting guidelines as subsequently sketched.

- *Planning initiation:* In this first phase, the question 'where we start' is answered. Therefore, the scope and reach as well as the goals of the EAP initiative are defined [SH93, pages 38–39]. Further, the organizational setting is investigated in respect to favorable and unfavorable organizational characteristics and the corporate culture [SH93, pages 39–44]. A vision, which has to be confirmed by the management. Finally, the set up of the EA program has to be performed, e.g. defining and adapting the method to be used, setting up the EAP team, and defining the workplan and deliverables to be created.
- *Business modeling:* In this phase, the structure of the organization is documented, the main business functions are identified and the business model is documented as well. The conduction of an enterprise survey is proposed to validate the developed business model. Therefore, steps, tasks, and guidelines are detailed facilitating the conduction. The aspect of communicating the achieved results is referred to by identifying potential addressees and means for distributing and discussing the achieved results.
- *Current systems & technology:* Based on the developed vision and workplan, the information to be gathered on the constituents of the EA, e.g. applications, technology components, or business functions is determined in this phase. Methods and guidelines how to perform data collection, maintenance, and presentation are discussed.
- *Data architecture:* In this phase, the candidate data entities are listed and related to the business functions. Therefore, the attributes and relationships to other architecture elements has to be defined. Complementingly, methods and means, i.e. deliverables, how to distribute the architecture description are presented.
- *Application architecture:* Derived from the data architecture, this phase is concerned with listing candidate applications, analyzing the impact to the current application portfolio, and distributing the plans developed. Therefore, the relevant attributes and relations for applications have to be define prior to the collection process.
- *Technology architecture:* The purpose of this phase is to identify the underlying principles for technology platforms and potential future platforms. Thus, in a first step, the technology principles and platforms are identified and their future distribution and relations to applications and business functions are specified. Again, the aspect of communicating the achieved results is emphasized by providing methods and meant how to accomplish this task.
- *Implementation/migration plans:* The architectures developed in the above phases have to be implemented in an integrated fashion. This phase is concerned with preparing the respective plan. Thus, in a first step a prioritization of the application to be implemented is defined, the efforts and resources are estimated and used to prepare a plan. Subsequently, the plan is evaluated in respect of costs and benefits and success factors and recommendations have to be determined.

Each of the above described phases is subdivided in steps and accompanied by tasks and guidelines enabling the implementation. While these tasks and guidelines present in some cases detailed method descriptions, e.g. for data gathering (cf. [SH93, pages 152–155]), in other phases, e.g. impact analysis (cf. [SH93, pages 217–221]), the method descriptions stay on a rather abstract level, leaving implementation detail to the reader. The above introduced phases are further detailed by Spewak and Hill by describing the steps to be performed, deliverables to be created, utilized procedures, and supporting guidelines. If necessary, the relation between the EAP process and related management functions are sketched and differentiated from a theoretic perspective (cf. strategic business planning in [SH93, pages 86–89]).

INTEGRATION	unidirectional		bidirectional		
DEVELOP & DESCRIBE	current	planned	target	principle	question
COMMUNICATE & ENACT	current	planned	target	principle	question
ANALYZE & EVALUATE	current	planned	target	delta analysis	
CONFIGURE TO	organizational context		scope and reach		
ADAPT TO	organizational context		scope and reach		

Table 3.7: Method classification for EAP

To enable organizations in successfully conducting EAP endeavors, Spewak and Hill discuss obstacles that need to be overcome, e.g. awareness/recognition/acceptance by top management, inexperience with EAP/lack of training, or (cf. [SH93, pages 20–32]). Thereby, the description of an obstacle is complemented by a description of symptoms and possible solutions. These obstacles mainly regard communication and enactment issues of EA management-related aspects. Some of the thereby announced obstacles can be related to organizational settings of the enterprise, e.g. unfavorable corporate culture, fear of loss of data control, or inaccessible or uncooperative users. In this sense, the proposed solutions can be regarded as proposed methods for context specific settings in terms of our analysis framework. In line with the understanding of EA management as a project-related initiative, the aspect of adaptation is neglected, thus contributing to an overall evaluation of its method-related prescriptions as shown in Table 3.7.

While abstaining from presenting an overall model for the EA, Spewak and Hill discuss the relevant elements that make up an EA based on the layers as presented by the Zachman framework (see Section 3.1). In line with the there presented structuring, the elements are distinguished according to their affiliation to *business model*, the *data, application*, and *technology architecture* [SH93, pages 12–13]. The identification of relevant elements, attributes, and relationships is left to the reader as part of the EAP process (see above). Recommendations on elements are however provided by the examples given, e.g. in [SH93, pages 146–148], and in the appendix [SH93, pages 323–330]. Although the EAP method focuses on planning aspects of EA management, cross-cutting aspects like projects, strategies, or principles are not linked to the elements constituting the EA. The importance of balancing time resources and the level of detail to ensure the right scope and reac of the EA management initiative is discussed in [SH93, pages 38–39], leading to an evaluation of EAP in respect to our language framework as shown in Table 3.8.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.8: Language classification for EAP

3.5 The Generalised Enterprise Reference Architecture and Methodology (GERAM)

EA management approach	
Name of approach:	The Generalised Enterprise Reference Architecture and Methodology (GERAM)
Issuing organization:	IFIP-IFAC Task Force on Architectures for Enterprise Integration
Focus area:	-
Tool support:	-
Period of activity:	since 1994
Publications:	[BN94], [BN96], [IF99], [In99], [BNS03], [IF03], [No03], [In06]
Inner organization:	monolith

In the 1970s and the 1980s several EA-related frameworks have been developed (cf. CIMOSA in [CI04], the Zachman framework as discussed in Section 3.1, DoDAF in [De09a, De09b], or ARIS in 3.2). In response to the emerging number of frameworks in this area, the International Federation of Information Processing (IFIP) and the International Federation of Automatic Control (IFAC) established the *International Task Force on Enterprise Integration* aiming at the development of a reference framework that supports comparison and evaluation of existing approaches (cf. Bernus and Nemes in [BNS03, page 13]). As a result of the investigation, the Task Force developed the *Generalised Enterprise Reference Architecture and Methodology (GERAM)*, which in 2000 became part of the international standard ISO 15740:2000 [In99]. The intention of GERAM is to provide a framework to compare, evaluate, and combine existing methodologies and modeling techniques. Furthermore, GERAM can be utilized to identify missing elements in existing approaches for EA management but can also be used as an EA framework itself. Therefore, mappings of established approaches to GERAM exist (cf. Noran in [No03]). According to [IF03, pages 23–24] a central aspect of GERAM is the recognition and identification of feedback loops reflecting changes and impacts in the internal and external environment.

GERAM consists of the subsequently described nine components as illustrated in Figure 3.7. Thereby, the components do not impose particular methods or models but define the criteria, which must be satisfied by an EA management approach (cf. [IF03, page 25]):

- *Generalised Enterprise Reference Architecture (GERA):* GERA describes the basic concepts to be used in enterprise engineering and integration projects. According to GERAM these concepts can be categorized as *human oriented concepts*, e.g. capabilities, skills, know-how, and roles of humans in the enterprise organization (responsibilities, authorities, etc.) and operation (the qualities of humans as resource elements), *process-oriented concepts*, e.g. functionality, behavior, entity life-cycles, and activities, and *technology oriented concepts* describing the supporting technology involved in enterprise transformation and operation.
- *Enterprise Engineering Methodology (EEM):* EEMs provide process models or structured procedures with detailed instructions for enterprise engineering and integration.

- *Enterperise Modeling Languages (EMLs):* EMLs define the generic modeling constructs for enterprise modeling. In particular, the EMLs provide constructs to describe and model human roles, operational processes, supporting information, and technologies.
- *Generic Enterprise Modeling Concepts (GEMCs):* GEMCs define and formalize the generic concepts of enterprise modeling. The following ways of formalization exists (in increasing order of formality): natural language explanations (*glossaries*), meta models describing the elements and their relationships (*information models*), and theories defining the meaning, i.e. semantics of enterprise modeling languages (*ontologies*).
- *Partial Enterprise Models (PEMs):* PEMs represent reusable, paradigmatic models capturing characteristics of enterprises. They capitalize on previous knowledge by supporting the development of model libraries in a 'plug-and-play' manner rather than developing models from scratch. PEMs may cover the whole or a part of the organization under consideration and typically concern a variety of enterprise entities such as products, projects, or companies.
- *Enterprise Engineering Tools (EETs):* EETs provide implementation support for the methodology and modeling language used for enterprise transformation, e.g. a shared repository that enables creation and maintenance of PEMs.
- *(Particular) Enterprise Models (EMs):* EMs capture concepts common to many enterprises and are expressed using a certain enterprise modeling language. They are used for analysis or represent executable models to support the operation of the enterprise. EMs may consist of several models, which describe certain aspects, i.e. views, on the enterprise.
- *Enterprise Modules (EMOs):* are components that can be utilized in implementing the organization. Exemplary EMs are human resources with given skill profiles, common business procedures (e.g. banking and tax rules), or IT infrastructure. The EMOs can be used to model portability and interoperability facilitating homogenization of heterogeneous environments as well as provide decision support and operation monitoring and control via real-time access to the enterprise environment.
- *Enterprise Operational Systems (EOS):* EOS represents the operation of a particular enterprise, which is typically guided by a particular enterprise model. This model provides the specification of the system and defines the EMOs used in the implementation of the particular enterprise, i.e. the EOS consists of all hardware, software, and socio-technical elements needed to fulfill the enterprise objectives and goals.

Reflecting the method-language-dichotomy as introduced by our analysis framework, GERAM distinguishes between these methodologies for enterprise engineering (EEMs) and the modeling languages (EMLSs) used by the methodologies. These two components and the GERA, which defines the foundations, are discussed subsequently alongside the analysis frameworks provided in Section 2.2.

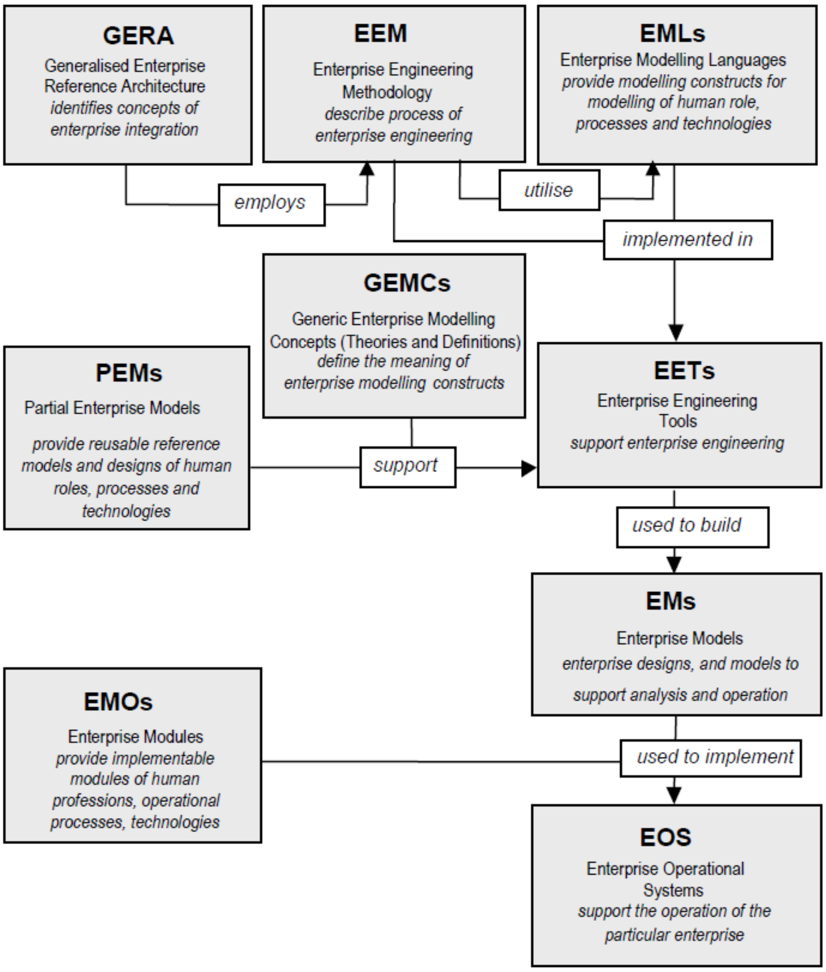

Figure 3.7: The components of the GERAM framework [IF99, page 5]

GERA provides three dimensions for defining the scope and content of enterprise modeling as shown in Figure 3.8 (cf. [IF03, pages 42–44][6]), namely

- *life-cycle dimension* providing means for modeling entities according to the life-cycle activities,
- *genericity dimension* supporting the controlled particularization, i.e. instantiation, from generic and partial to particular, and
- *view dimension* enabling visualization of specific views of the enterprise entities.

Besides these dimensions, GERA opts to define the pragmatic purpose for each view and therefore concepts to be considered by the EA endeavor. Possible pragmatic purposes, e.g. support of design choices, simulation of processes to identify characteristics as cost or duration are given in [IF03, page 45].

[6] The *GERA Modelling Framework* represents the basis for the International Standard ISO 19439:2006. Framework for Enterprise Modelling [In06].

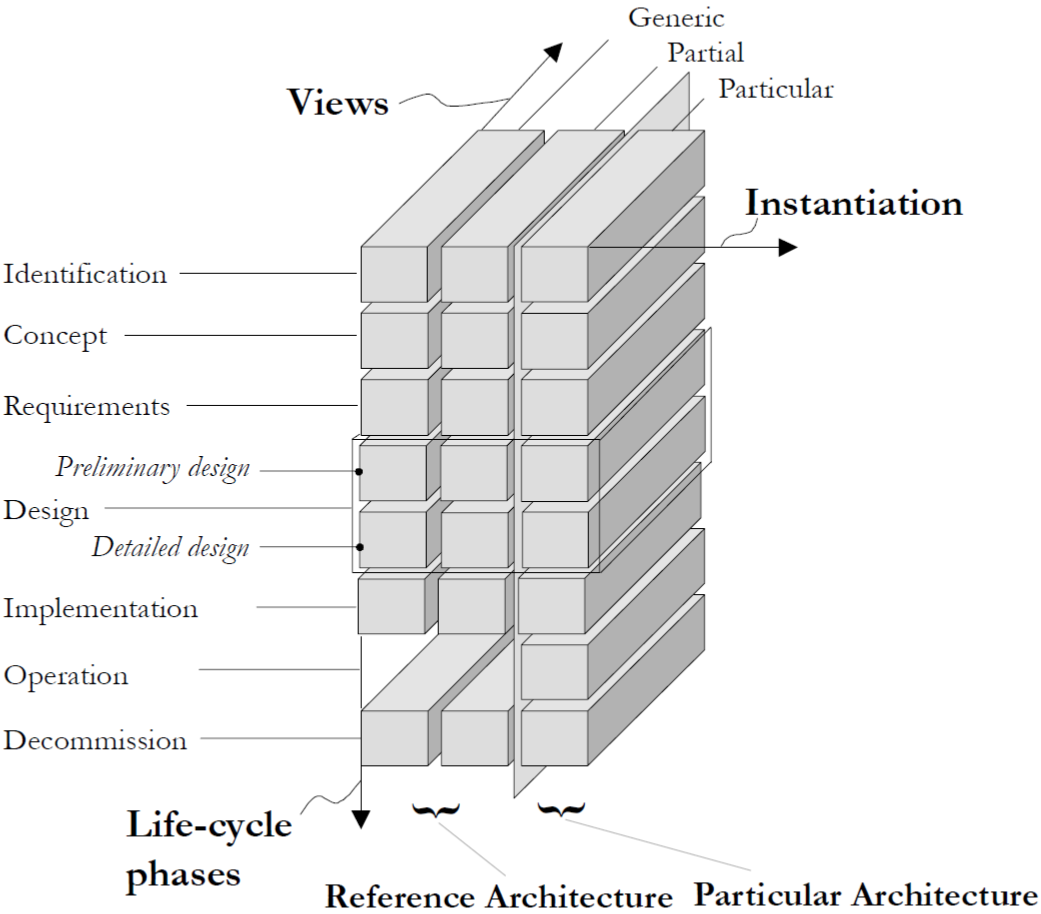

Figure 3.8: The components of the GERA modeling framework [IF99, page 18]

GERA defines a life-cycle for each constituting concept of the enterprise, which consists of the phases *identification, concept, requirements, (preliminary and detailed) design, implementation, operation*, and *decommission*. While most of the aforementioned phases are self-explanatory, the *concept* phase deserves a more in depth analysis with respect to our analysis framework. The phase is concerned with the definition of the entity's mission, vision, strategies, objectives, etc. [IF03, pages 32–34]. Thus, linking the cross-cutting aspects of strategies, projects, visions, and goals to any concept considered during enterprise transformation. In line with the objective of GERAM to define requirements for EA (management) frameworks, no description how this relation should be conceptualized is given. Similarly, the concept of *life history* is discussed as main aspect of EA management approaches by GERA and the link to different kind of projects, e.g. engineering, redesign, or improvement projects, is discussed and related to the phases of the EA concepts.

Following a systemic perspective, GERA proposes a *recursive enterprise entity type* concept, which consists of five different types (cf. [IF03, page 39]).

- *Entity type 1 (strategic management entity)* defines the necessity and the starting of any EA-related effort.

- *Entity type 2 (engineering implementation entity)* provides the means to carry out the EA-related effort, i.e. uses a methodology (entity type 5) to define, design, implement, and build the operation of the enterprise entity (entity type 3).
- *Entity type 3 (enterprise entity)* uses the methodology (entity type 5) and the operational system provided by entity type 2 to define, design, implement, and build the products of the enterprise (entity type 4).
- *Entity type 4 (product entity)* represents all products (or services) of the enterprise.
- *Entity type 5 (methodology entity)* represents the used methodology in the course of operation, which in general leads to the creation of another entity type.

In line with the objective of GERAM to represent an evaluation framework for EA (management) approaches as well as an EA management framework itself, the life-cycle concept introduced above does not only apply for the constituting concepts of the enterprise but applies to the enterprise itself, thus representing a methodology for EA management. Besides this basic description, the EEMs define further requirements for an EA management function. According to [IF03, pages 49–50], the methodologies should be described in terms of process models or descriptions with detailed instructions for each activity, the information used and produced, resources needed, and relevant responsibilities assigned to the tasks. Again, special emphasis is put on the 'human aspect' as the EEMs opt for an explicit modeling of humans and their relations to tasks, responsibilities, and influences. The role of humans as supporter or opponent of an EA management initiative and therefore the human role as success factor is alluded to by the EEMs (cf. [IF03, pages 50–52]). Dedicated methods and means how to overcome this challenge are however not provided. The aspect of human knowledge and tacit knowledge in particular is accentuated and specialized models are proposed to address this challenge. In addition, the methodologies may employ the relevant EMLs. Emphasizing on the requirements defined by GERAM and the EEMs in particular, the overall evaluation of the method-related prescriptions of GERAM is shown in Table 3.9.

INTEGRATION	unidirectional		bidirectional		
DEVELOP & DESCRIBE	current	planned	target	principle	question
COMMUNICATE & ENACT	current	planned	target	principle	question
ANALYZE & EVALUATE	current	planned	target	delta analysis	
CONFIGURE TO	organizational context		scope and reach		
ADAPT TO	organizational context		scope and reach		

Table 3.9: Method classification for GERAM

In addition, to the generalized propositions for a language for EA descriptions as discussed above, the EMLs define two requirements to enable integration of special purpose modeling languages (cf. [IF03, page 54]). First, every area as represented in the modeling framework must be covered for every enterprise entity type, and second, any model developed must be able to be integrated with models of other subject areas, if the information content of the model requires integration. The need to integrate different languages results from the distinct 'expressive powers' related to the intended purpose, e.g. description vs. analysis, of the languages. Thereby, a link between the methodologies used and the supporting languages can be established.

Keeping in mind the aim of generalization and therefore abstaining from presenting an explicit language for EA description, the requirements elicited by GERAM result in the evaluation given in Table 3.10.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.10: Language classification for GERAM

3.6 Semantic Object Model Approach (SOM)

EA management approach	
Name of approach:	Semantic Object Model (SOM) Approach
Issuing organization:	University of Bamberg
Focus area:	Modeling
Tool support:	SOM Modeling Environment (in development – see `http://www.openmodels.at/web/som`, last-cited 09-09-10)
Period of activity:	since 1994
Publications:	[Fe94], [FS95], [FS97]
Inner organization:	monolith

In [FS95] Ferstl and Sinz diagnose a fundamental change in the way business information systems are understood via models. Whereas up to this point information systems modeling centered around structural aspects of the systems, more recent approaches in those days started to identify IS with the set of interlinked business processes that the systems support. Especially with this perspective, the focus of IS modeling is broadened to not only incorporate the single system but also its enterprise environment, called the "context of the EA" by Ferstl and Sinz in [FS95]. Reflecting this understanding of the enterprise, the Semantic Object Model (SOM) approach introduces two key abstractions: the "business transactions" reflecting the exchange of services between "business objects" that conversely provide or consume such services (cf [Fe94]). Key principle of the approach is the "decomposition" of both objects and transactions into smaller parts thereof, getting from an abstract perspective on an enterprise to more concrete descriptions. This is further mirrored in the approach's framework for the EA as shown in Figure 3.9. The three layers of the EA cover the strategic goals as well as the value proposition of the enterprise, the supporting business processes, and the necessary organizational, technical and physical implementation thereof.

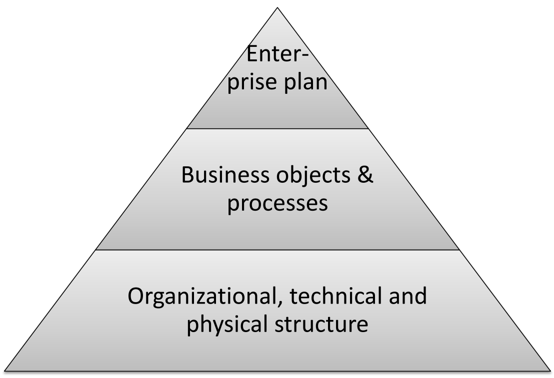

Figure 3.9: Enterprise architecture framework of the SOM approach [FS95, page 8]

As Ferstl and Sinz describe in [FS95] each layer of the EA has both a structural and a behavioral aspect, which have in turn to be covered by an appropriate description method. Centrally, this method is described via the so called "V-model" of the SOM approach as shown in Figure 3.10. On the top-most level the method calls for informal (textual) descriptions of the enterprise plan both from a structural and a behavioral perspective. The latter perspective covers the enterprise's value proposition and strategic goals, whereas the former perspective is used to distinguish between the enterprise systems and its environment. In a subsequent step the structural plan of the enterprise is refined into an "interaction model" (cf. [FS97]) describing the interacting internal and external business objects, as e.g. organizational units or clients. Building on the interaction model's description of the participants, the "value model" of the enterprise is described by connecting the participants via transactions. In subsequent steps the preceding models are further detailed, finally concluding in the "conceptual object design" which in turn forms the basis for implementing business information systems (cf. [FS97]).

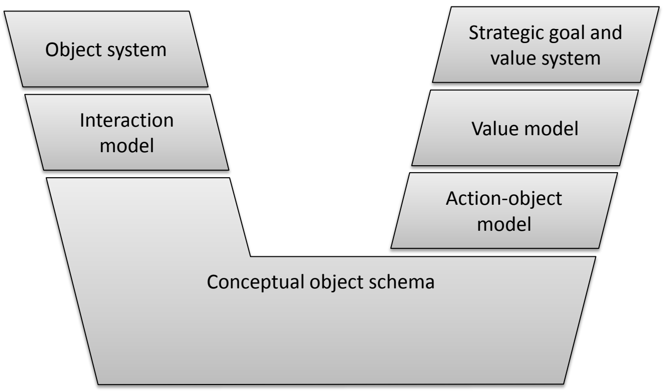

Figure 3.10: V-Model (method model) of the SOM approach [FS95, page 9]

For the different modeling levels below the enterprise plan, Ferstl and Sinz [FS95, page 16] call for formal modeling techniques and languages based on the principle of decomposition. Using a BNF-like syntax, they describe decomposition rules which are to be applied in the V-model method. An exemplary rule reads as follows

$$O ::= \{O_1, O_2, [T(O_1, O_2)]\}$$

and describes that a modeler may decompose an object into a set of two sub-objects that are optionally linked with a transaction. Similar rules for decomposing transactions also exist. While not explicitly stated the aforementioned method can be applied to engineer both a target state for an organization as well as to re-engineer the current state available. Reflecting these characteristics, we classify the method-related prescriptions of the SOM approach as shown in Table 3.11.

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.11: Method classification for the SOM appproach

Central to the SOM approach is the dichotomy of the object system (structure) and the process system (behavior) that reverberates over all layers of the EA framework shown in Figure 3.9.

Ferstl and Sinz reify this core distinction in the integrated meta-model of the approach as described in [FS95, page 16].

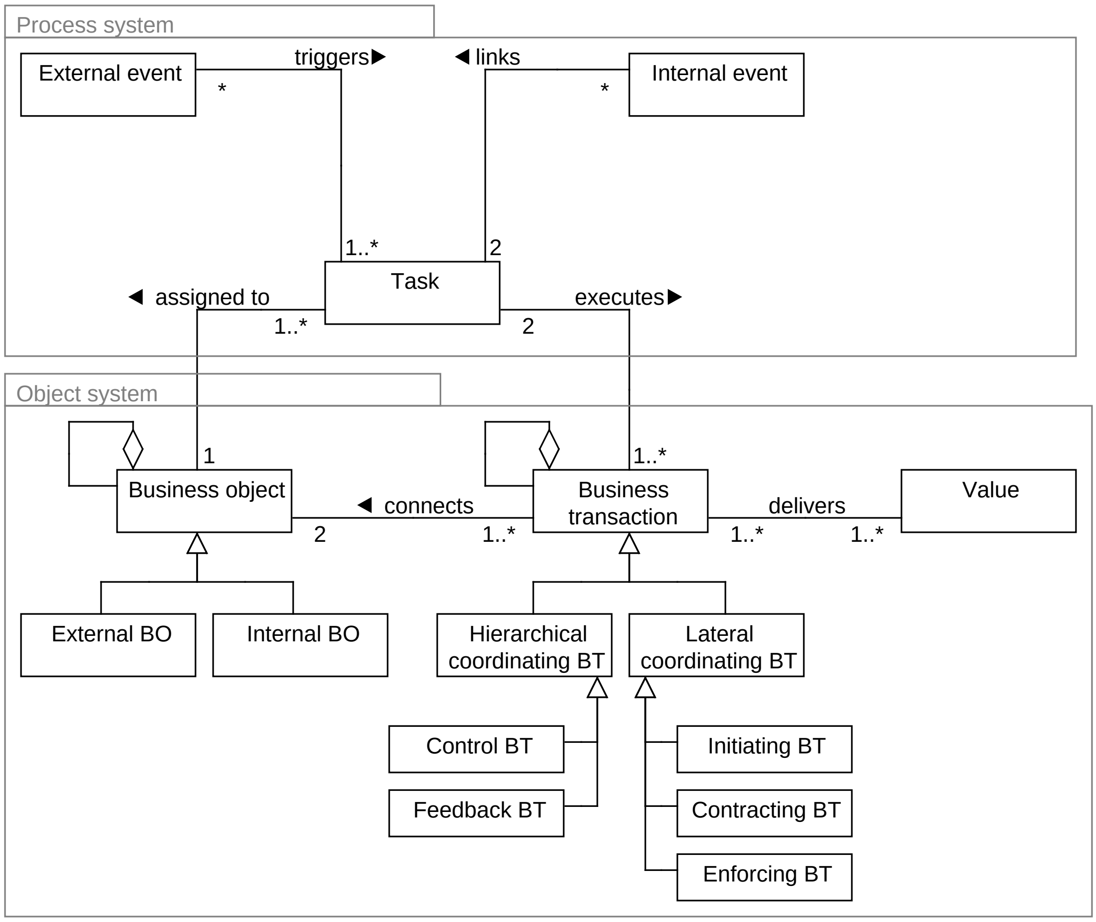

Figure 3.11: Meta-model of the SOM approach [FS95, page 16]

This model (cf. Figure 3.11) concretizes the two systems via corresponding concepts. Processes are decomposed into tasks, which in turn may be triggered by external events or form a sequence of tasks linked via internal events. For the object system, Ferstl and Sinz introduce a distinction between external and internal participating business objects. Concerning the kinds of transactions in which such objects are involved, two principles outlined in [Fe94] are incorporated into the model:

- *Negotiation principle* This principle describes that every transaction may be seen as composed of initiating transactions (request), contracting transactions (acknowledge) and enforcing transactions (deliver). These kinds of transactions are used to describe interactions between actors that have no hierarchic control relationship inbetween ("lateral coordination").
- *Hierarchic coordination principle* This principle describes that hierarchic relationships can form the basis for coordination in a sense that one participating partner (super) controls the other partner (sub), which in turn provides feedback on the transaction.

Building on the principles, the meta-model and a set of transformation rules, as exemplified above, the SOM approach has a strong focus on the business perspective on both the enterprise as well as its information systems. Not directly reflected in the meta-model is the linkage between the tasks and the strategic goals as well as the value proposition of the enterprise. On the downturn, the conceptual object model that may well be identified with a 'classic' object-oriented model for IS is not directly linked to the meta-model but discussions on potential linkages are undertaken by Ferstl and Sinz in [FS95, FS97]. In terms of the analysis framework for the language aspect as described in Section 2.2, the SOM approach classifies as depicted in Table 3.12.

BLACK-BOX PERSPECTIVE	business & organization	application & information	infrastructure & data
WHITE-BOX PERSPECTIVE	business & organization	application & information	infrastructure & data
STRATEGIES & PROJECTS	business & organization	application & information	infrastructure & data
VISIONS & GOALS	business & organization	application & information	infrastructure & data
PRINCIPLES & STANDARDS	business & organization	application & information	infrastructure & data
QUESTIONS & KPIs	business & organization	application & information	infrastructure & data
CONFIGURE & ADAPT	initially	evolutionary	

Table 3.12: Language classification for the SOM approach

3.7 Multi-perspective Enterprise Modeling (MEMO)

EA management approach	
Name of approach:	Multi-perspective Enterprise Modeling (MEMO)
Issuing organization:	University of Duisturg-Essen
Focus area:	Modeling
Tool support:	-
Period of activity:	since 1994
Publications:	[Fr94], [Fr98a], [Fr98b], [Fr99], [Ju07], [Fr08], [He08], [Ki08], [Fr09]
Inner organization:	explicit organization

In [Fr94] Frank motivates the need for models "of the whole enterprise or of parts of it" as means to address IS-related challenges as well as to leverage IS-enabled innovation, such as business process re-engineering. These models are required to span the different levels of the organization, ranging from a strategic perspective over an organization perspective to an IS perspective. The different perspectives should not be treated independently in order to reduce the risk of friction. Reflecting this requirement, Frank devises the notion of the "multi-purpose enterprise modeling" bringing together different viewpoints and foci on the enterprise (cf. Figure 3.12).

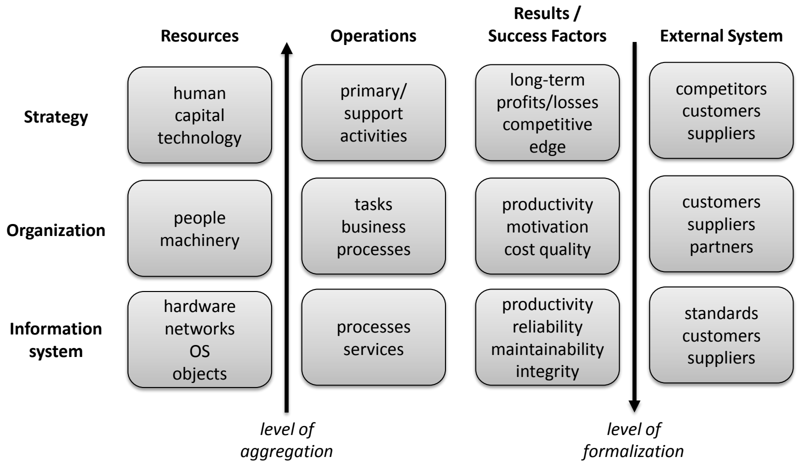

Figure 3.12: Viewpoints and foci of MEMO

In subsequent publications the approach was refined and renamed to "multi-perspective enterprise modeling", MEMO for short. Initially MEMO consists, as Frank describes in [Fr99] of three distinct modeling languages, namely the "strategy modeling language", the "organization modeling language" and the "object modeling language". Latter language is thereby specifically designed to support object-oriented development of information systems. Subsequent work introduced additional languages, namely the "resource modeling language" [Ju07] and the "IT modeling language" [Ki08]. Another language specifically targeting the development of indicator systems for IT controlling, the so-called "score modeling language" is yet in development (cf. Frank et al. in [Fr08]).

The MEMO approach presents itself a strongly focus on languages and modeling, i.e. on the creation of models using the corresponding languages. To optimally support language users in model creation, the languages are specifically designed for the corresponding application domain. Complementing the language only little methodical guidance is provided. A notable exception among the early languages is the object modeling language (OML). In [Fr98b] Frank not only describes the language features but relates them to the different steps of object-oriented development, such as "information analysis", "object design" and "system design". Using many examples, the prospective user of the language is guided on how to develop an object model in a stepwise fashion. With the specific focus on modeling the complex subject of the enterprise IT landscape, the IT modeling language of Kirchner [Ki08] identifies the different steps of enterprise IT modeling as follows:

1. *planning and design*,
2. *implementation*,
3. *operation*, and
4. *evaluation*.

For each of these steps potentially involved actor roles, such as "architect" or "SLA manager", are denoted and their specific competencies as well as responsibilities are described. For each step a list of tasks is given, providing more in-detail information on the execution. Each single task is thereby also linked to the relevant language elements, which are affected. Complementing prescriptions on how to perform modeling and planning-related activities, Kirchner further outlines how typical IT-relevant questions, as "which processes are supported by the IT landscape?" may be answered based on the specific models. For each of these questions, a short description of a possible analysis technique is provided and useful visualizations are exemplarily described. With respect to the design of a "business indicator system" based on enterprise models, Frank et al. outline in [Fr08] a method consisting of three phases "design", "use" and "refinement", of which the design phase is further detailled into three steps. The first of these steps sketches how business indicators may be derived top-down from relevant strategies and goals or may be elicited bottom-up from "measurable aspects". In the second step, the identified indicators are interlinked in an indicator system, but also connected to the "reference objects", i.e. enterprise elements that they apply upon. The third and final step centers around applying the indicator system, i.e. around evaluating the indicator values. Summarizingly, we can state that method-related guidelines are not in the focus of the MEMO approach, leading to a classification as shown in Table 3.13.

Integration	unidirectional	bidirectional			
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context	scope and reach			
Adapt to	organizational context	scope and reach			

Table 3.13: Method classification for MEMO

By design MEMO is able to mirror the plurality of perspectives on the enterprise by supplying different languages. Each language brings along its unique set of concepts together with a dedicated notation (cf. Frank [Fr99]). The object modeling language, for example, [Fr98b] supplies typical object-oriented concepts, as classes, associations and attributes, which may in turn be used to describe the object model underlying an information system. The resource modeling language of Jung [Ju07] supplies concepts for describing physical resources, as "computing devices", intangible resources, as "patents" and "software", as well as human resources together with their hard and soft skills. Each of these concepts is further detailed via corresponding attributes covering certain characteristics, such as "performance". Switching from the resource perspective on hardware devices, the IT modeling language of Kirchner [Ki08] provides concepts that detail the resources and allow describing IT-specific relationships among them. In particular, specific hardware devices are concretized, i.e. "scanners" or "printers" are distinguished via specialized classes bearing specific attributes. In a similar sense, the general concept software is further detailed via concepts e.g. to model "layered IT architectures" as well as "communication formats" for interfaces. The score modeling language of Frank et al. [Fr08] and Heise et al. [He08] reflects a cross-cutting perspective on the enterprise, dedicated to modeling arbitrary indicators that may be assigned to reference objects in the enterprise model. The mechanism of the reference object thereby concretizes a recurring principle of the MEMO approach. Different languages committed to a specific perspective on the enterprise are grounded in a single meta-language and are interlinked via shared concepts. Figure 3.13 displays the basic make-up of the language stack of MEMO.

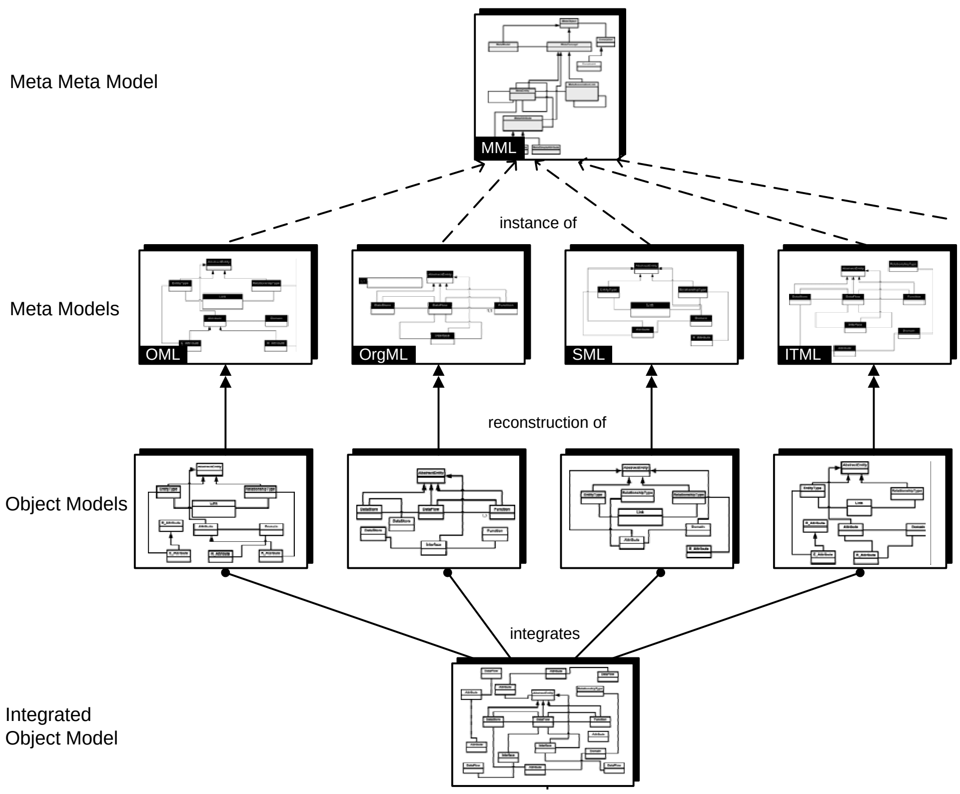

Figure 3.13: Make-up of MEMO's language stack [Fr09]

The MEMO meta-language (MML) has evolved over time since its initial publication in [Fr98a], but has retained its object-oriented nature to its most recent version, presented by Frank in [Fr09]. In all versions the meta-language supplies specific mechanisms to indicate that a language concept is a “border object”, i.e. is located at the intersection of two language perspectives. Summarizing above findings about the MEMO approach, we classify it according to our framework as shown in Table 3.14.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.14: Language classification for MEMO

3.8 The Open Group Architecture Framework (TOGAF)

EA management approach	
Name of approach:	TOGAF
Issuing organization:	The Open Group
Focus area:	Method
Tool support:	TOGAF 9 Method Plugin for the Eclipse Process Framework Composer tool[7]
Period of activity:	since 1995 (TOGAF version 1.0)
Publications:	[Th09a], [Jo09], [Th09b]
Inner organization:	explicit organization

The Open Group is a vendor and technology-neutral consortium with the objective to foster information flow via open standards for enterprises[8]. In 1995, The Open Group published the first version of the *The Open Group Architecture Framework (TOGAF)* which was based on the *Technical Architecture Framework for Information Management (TAFIM)* published by the Department of Defense. The current version 9.0 of TOGAF has been released in October 2009 [Th09a]. TOGAF is based on the terminology introduced in the ISO Standard 42010 [In07] and provides a method and supporting models and techniques for developing an EA management function. As a widely-used and known framework, the major players in the market of EA management tools have incorporated TOGAF in their tools (cf. the analysis of sebis in [se05] and Matthes et al. in [Ma08]). In addition, a method plugin for the open source eclipse process framework composer exists[9]. TOGAF 9 consists of six main parts, namely

- the *architecture development method (ADM)* describes an iterative process consisting of eight interconnected phases of EA development and a complementary preliminary phase (see Figure 3.14),
- the *ADM guidelines and techniques* cover aspects of adaptability and configuration of the ADM to different process styles or specific architectures, i.e. security
- the *content framework* provides a conceptual metamodel for describing architectural artifacts,
- the *enterprise continuum and tools* represents a view on the architecture repository providing methods to structure and classify architecture and solution artifacts to enable communication and reuse of EA-related descriptions, and
- the *TOGAF reference models* are divided into the *TOGAF foundation architecture* and the *integrated information infrastructure reference model* (III-RM). The foundation architecture is embodied in the *technical reference model* (TRM), which is universally applicable and and can be used to build any system architecture. The III-RM helps to address the need to design an integrated information infrastructure with reference designs.

[8]See `http://www.opengroup.org/overview`, last accessed 2010-09-03
[9]See `http://www.opengroup.org/architecture/togaf/epf_intro.html`

In line with other EA management approaches, TOGAF proposes to structure the EA in different architecture domains representing subsets of the overall EA [Th09a, page 10]. Thus, TOGAF distinguishes between

- *business architecture* is concerned with strategic, governmental, organizational, and process-related aspects,
- *data architecture* describing the structure of an organization's data assets and data management resources,
- *application architecture* considers the application systems, their interactions, and their relationships to the business processes, and
- *technology architecture* describing the logical software and hardware capabilities required to support the deployment of business, data, and application services.

Focusing on methodical aspects, the most-known part of TOGAF is the ADM, which describes an iterative process consisting of eight phases, which are complemented by a preliminary preparation phase and the central activity of requirements management (see Figure 3.14).

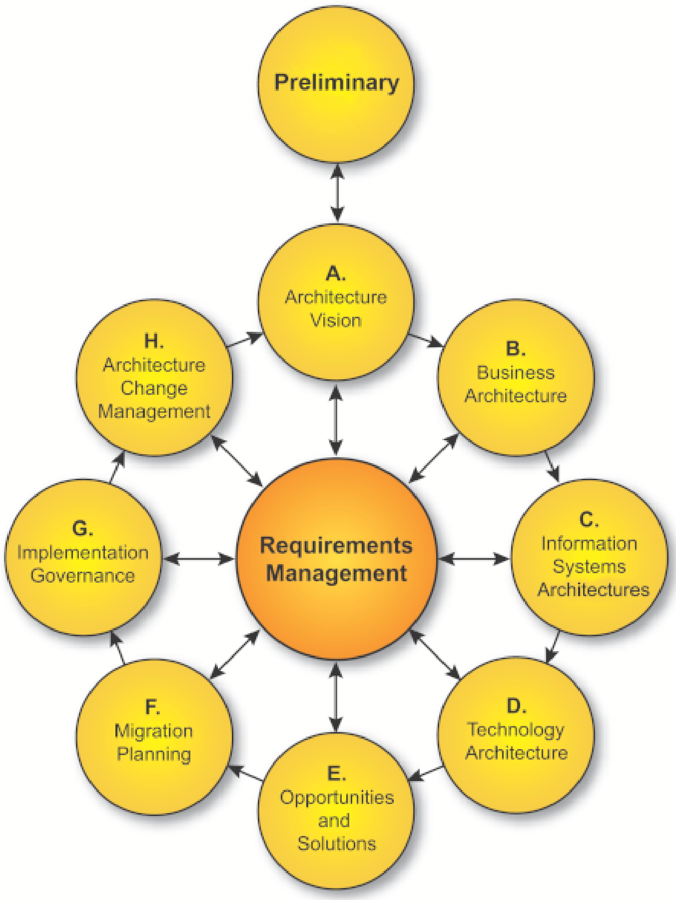

Figure 3.14: The architecture development method of TOGAF [Th09a, page 54]

The TOGAF ADM cycle starts with the *preliminary* phase, which prepares and initializes the EA management project. Typical tasks executed in this phase include the definition and establishment of the EA team, the selection and implementation of supporting tools, as well as the definition of architecture guidelines and principles. After the preparation and initialization activities are performed, the scope of the EA management endeavor is defined within the *architecture vision* phase (A). A core objective of this phase is to identify the relevant stakeholders and their concerns. Based on the identified stakeholders and concerns a high-level architecture vision of the enterprise is derived in this phase. Preceding phase A, the business, information systems, and technology architecture are developed in the phases *business architecture* (B), *information systems architectures* (C), and *technology architecture* (D) phase respectively. The fundamental make up of these three phases is very similar: Initially, the baseline architecture (current state of the EA) is described. Based on this architecture, a target architecture is developed taking the architecture vision into account. This vision was formulated as part of the preceding phase A. A delta analysis is performed to evaluate the differences between the current and the target architecture and roadmap components enabling the transition form baseline architecture to target architectures. The phase *opportunities and solutions* (E) is concerned with linking the separated business, information system, and technology architecture and deriving projects and programs, which describe the transformation from the current to the target architecture via intermediate transition architectures (planned states). The major steps to be performed in this phase are the consolidation of the delta analyses from phases B to D, the identification, refinement, and validation of dependencies between the different architectural layers, and the establishment on an integrated project and program portfolio. The transition architectures form the input of the *migration planning* phase (F), which is concerned with the formulation of an implementation and migration plan that schedules and realizes some or all of the planned architectures. The major steps within this phase are the assignment of a business value to each project, the prioritization of projects, and the generation of a roadmap and migration plan. In the phase *implementation governance* (G) the projects selected for realization in the preceding phase are executed. Major tasks of this phase are the identification of deployment resources and skills, monitoring of the execution, and the conduction of reviews, e.g. regarding architecture compliance. The final phase (H) *architecture change management* concludes an ADM cycle and prepares the initiation of the next iteration. As part of the phase, the changes of the architecture are assessed. Key tasks of this phase are the deployment of monitoring techniques for the architecture process, the development of change requirements to meet performance targets, and the management of the governance process.

The ADM of TOGAF thereby focuses on EA management-projects instead of a continuous EA management function. While this approach ensures that a sponsor for the EA management endeavor is available (see preliminary phase), it entails the disadvantage that each project has to start with information gathering as no up-to-date information and description of the EA is available. Complementing the high-level description of the phases, TOGAF provides exemplary guidelines and techniques for adapting or complementing the ADM (cf. The Open Group in [Th09a, pages 213–358]):

- *Capability-based planning:* A method for capability-based planning is presented by TOGAF, which enables a black-box view on the business level, i.e. the business capabilities of TOGAF include people, process, and material dimensions (cf. [Th09a, pages 353–358]).

- *Organizational contexts:* TOGAF for instance proposes to use hierarchies of ADM processes if the EA management-related project is too complex. As organizations typically differ strongly regarding the situation in which EA management is intended to be established, TOGAF discusses different situations in which an adaptation of the ADM might be required (cf. The Open Group in [Th09a, pages 56–57]). The ordering of the single phases, for instance, may be subject to adaptation (cf. discussion in [Th09a, page 217]).
- *Architectural principles:* Different best practices how to develop, document, and apply principles are alluded to in TOGAF (cf. [Th09a, pages 167–280]). While exemplary principles are presented, methods to communicate and enact them are not discussed.
- *Delta analysis:* A matrix-based approach to perform delta analysis is presented in [Th09a, pages 321–323].

To configure the management body of the ADM, TOGAF proposes three different dimensions for segmentation (cf. The Open Group in [Th09a, pages 58–63]). First, the EA can be segmented in respect to the *scope*, i.e. which specific business sectors, functions, organizations, geographical areas are to be included. Second, a segmentation in respect to *architecture depth*, i.e. are all types of architecture covered or only a subset thereof, e.g. the data and application architecture. Third, the management body of the ADM can be tailored in respect to *time*, i.e. only the baseline architecture (current state) is included. While the importance of tailoring the management body is alluded to by TOGAF, no mechanisms how to perform this configuration are given. Table 3.15 summarizes the key characteristics of TOGAF and especially the ADM classified against the background of the method framework from Section 2.2.1.

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.15: Method classification for TOGAF

Answering the question which elements should be considered in an EA management endeavor, TOGAF presents the *core content metamodel*, which is illustrated in Figure 3.15. Therein, the core entities and relationships that make up an EA are described (cf. The Open Group in [Th09a, pages 376–409]). The core content metamodel provides entities for all architectural layers from business architecture, e.g. organizational unit or process, via data and application architecture, e.g. application component or data entity, to technology architecture, e.g. technology component or platform service. Besides the entities, which can be grouped to one of the architectural layers, TOGAF also introduces crosscutting entities associated with all objects among others principle, requirement, work package. Thereby, the kind of relationship, e.g. affects, introduces, retires is not discussed. TOGAF further provides six metamodel extensions (cf. The Open Group in [Th09a, pages 380–392]), namely

- *governance extension* to support operational governance by introducing concepts as goal, objective, measure, and contract,

- *services extension* to enable description and management of IS services in addition to business services,
- *process modeling extension* to allow detailed modeling of process flows by adding product, event, and controls,
- *data extension* to enable more sophisticated modeling and management of data by introducing entities like physical or logical data component,
- *infrastructure consolidation extension* to support consolidation endeavors by introducing physical and logical application components as well as the location entity, and
- *motivation extension* to enable measurement of business performance by introducing concepts as driver, goal, and objective.

For each of the above extensions, TOGAF describes the situation in which the respective extension should be used and the benefits it bears. Table 3.16 summarizes the key characteristics of TOGAF and the content meta model in special classified against the background of the language framework from Section 2.2.2.

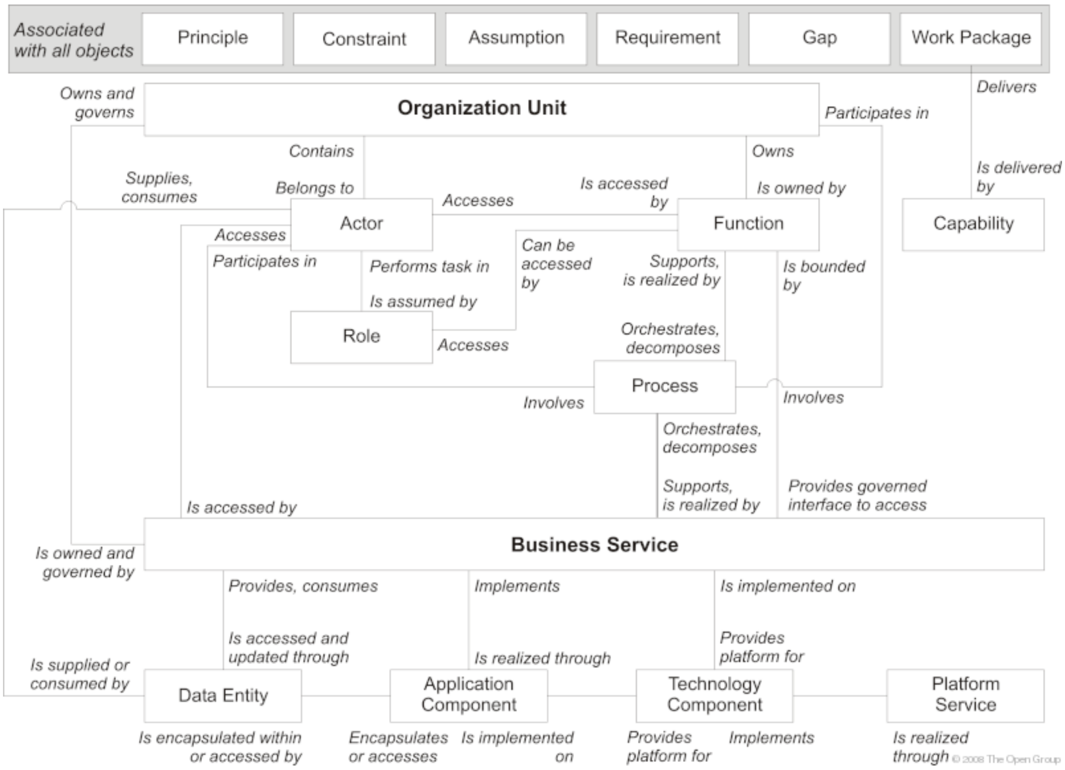

Figure 3.15: The core content metamodel of TOGAF [Th09a, page 376]

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.16: Language classification for TOGAF

3.9 Extended Enterprise Architecture (E2A)

EA management approach	
Name of approach:	Extended Enterprise Architecture (E2A)
Issuing organization:	Institute For Enterprise Architecture Developments (IFEAD)
Focus area:	Method
Tool support:	-
Period of activity:	since 2001 (foundation of the IFEAD)
Publications:	[SK04], [Sc06d], [Sc06c], [Sc06a], [Sc06b], [Sc08a], [Sc10]
Inner organization:	explicit organization

In 2001, Jaap Schekkerman who has more than 25 years of experience in managing complex and large enterprise architecture programs in the governmental area, healthcare, and high tech industry, founded the *Institute For Enterprise Architecture Developments (IFEAD)*. IFEAD is a non-profit research and information organization aiming at fostering the EA-related knowledge exchange. In 2002, the IFEAD published the first version of the *Extended Enterprise Architecture (E2A)* framework. The E2A framework is influenced by the Zachman framework (see Section 3.1) and by the IAF (see Section 3.3). Approaching the topic of EA management from a holistic perspective, the aspect of linking the EA management function with other closely related functions and processes, e.g. human capital management, information security management, and budgeting, is alluded to in [Sc08a, pages 36–37].

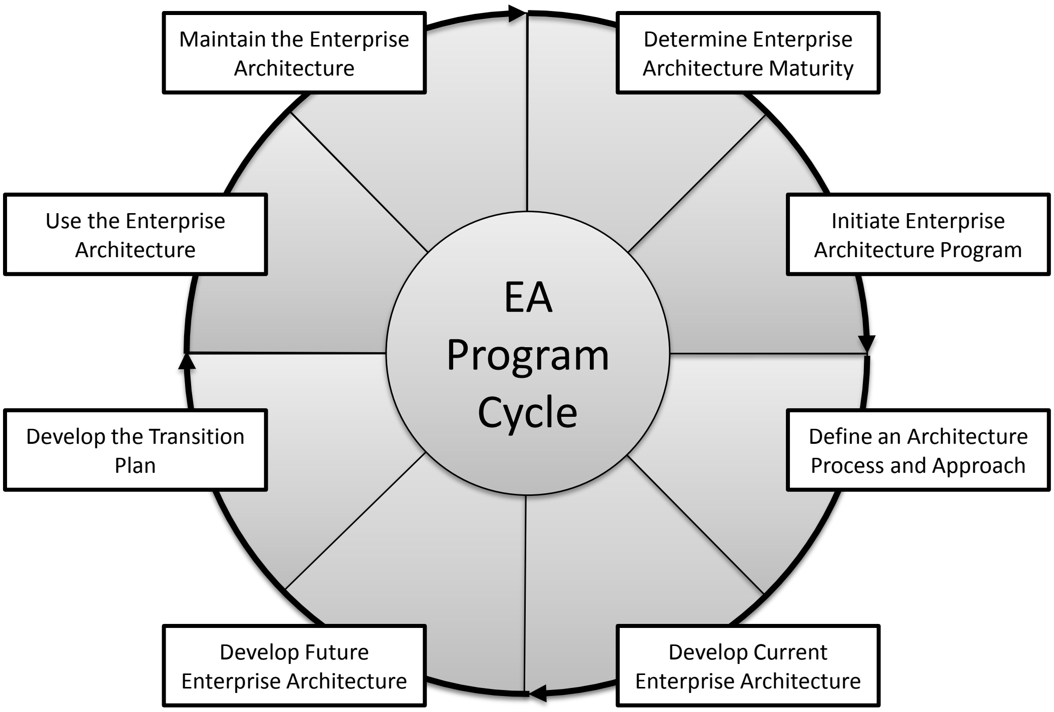

Figure 3.16: The enterprise architecture program cycle according to Schekkerman in [Sc08a, page 38]

A core contribution of the E2A is the so-called *Enterprise Architecture Program (EAP)*, which details on implementation steps for establishing an EA management function (see Figure 3.16). The EAP consists of eight steps, which are described subsequently.

- *Determine enterprise architecture maturity:* Preparing the establishment of an EA management function the current maturity level needs to be determined, therefore the E2A proposes the utilization of either existing maturity models or provides an own maturity model called *Extended Enterprise Architecture Maturity Model (E2AMM)* [Sc06c].
- *Initiate enterprise architecture program:* The set up of the EA management function is provided in this step. This includes linking the relevant management functions, e.g. IT portfolio management, and the establishment of a management structure and control, e.g. the EA steering committee, the chief enterprise architect, and the EA core team. For each role, E2A presents the associated responsibilities assigned to the team members, e.g. the information architect is responsible for documenting and analyzing business information and associated relationships (cf. Schekkerman in [Sc08a, pages 213–215]). Furthermore, the activities and results of the EA program are defined.
- *Define an architecture process and approach:* In this step, the intended use (goals), scope, and depth of the EA management endeavor are defined to ensure an EA management function sufficient for its purpose. The definition of scope thereby includes decisions on, e.g. geographical areas to be considered or the relevance of timeframes ([Sc08a, page 59]). The utilization of existing EA frameworks is proposed in order to answer the questions on goals, scope, and depth (a description of different EA frameworks is given in [SK04]). The final activity of this step is the definition of the EA process. Supporting this step the utilization of existing frameworks like TOGAF (see Section 3.8) or the Enterprise architecture process model introduced below is proposed as well as the selection of an appropriate tool is discussed.
- *Develop current enterprise architecture:* Phases of this step are a) discovery and data collection, b) design and preliminary results generation, c) review and revision, and d) publication and delivery of the EA results to an appropriate repository [Sc08a, pages 93–97]. For each of the aforementioned phases, methods and techniques how they could be performed, e.g. via interviews, 'quick looks', or documentation review, are discussed and basic questions to be answered are presented.
- *Develop future enterprise architecture:* In the same vein as the current state of the EA is documented in the preceding phase, the target state of the EA is documented. Essentials in creating the future state of the EA are thereby discussed, as e.g. the alignment with the strategic plan or the focus on business areas with the greatest potential payoff.
- *Develop the transition plan:* Based on the descriptions of the current and future state of the EA, a transformation plan is derived via a gap analysis. Additional dependency analyses between projects and the transition plans respectively are performed to provide input to project portfolio management [Sc08a, pages 99–104].
- *Use the enterprise architecture:* Enacting the transition plan as developed in the preceding step, this step provides good practices how the EA management function interacts with other enterprise-level management functions as e.g. project portfolio management.

To ensure architectural compliance of projects on the one hand, trainings, reviews, consequences, etc. for deviations are proposed (cf. Schekkerman in [Sc08a, pages 107–118]) on the other hand, the importance of reflecting the performed changes in the 'new' current EA description is referred to in the next step.

- *Maintain the enterprise architecture:* As organizations represent highly dynamic systems, which evolve over time, this step is concerned with maintaining the EA artifacts, e.g. current and future state, and with the continuous control and oversee of the overall EA management function. The latter should be performed as a continuous process, which takes quick and decisive actions to correct problems. Examples of that actions are redefinition of purpose and scope of the EA management function, introduction of or strengthening of existing control mechanisms to ensure continuous improvement of the overall function (cf. Schekkerman in [Sc08a, page 126]). In [Sc04], Schekkerman proposes an *Enterprise Architecture Score Card*^{TM} for assessing the performance of the EA management function.

Besides the above introduced program and its steps, the E2A proposes an *enterprise architecture process model* consisting of the steps a) enterprise architecture visioning, b) EA scope & context, c) EA goals / objectives & requirements, d) opportunities & solutions, e) organizational impact, f) benefits / business case, g) transformation planning, and h) implementation governance structure. A spiral model, which iterates through the aforementioned steps is proposed in order to adapt the idea of "think big but start small" [Sc08a, pages 81–84].

Addressing the communication challenge E2A proposes techniques to identify and classify stakeholders utilizing a power-interest matrix and proposes different sets of viewpoint types [Sc06a]. Furthermore, aspects of how to establish the EA governance, i.e. centralized, decentralized, or federated are alluded to in [Sc08a, pages 132–136]. Further, aspects as roles and responsibilities are discussed.

E2A introduced three different types of principles (cf. Schekkerman in [Sc08a, page 236]: *enterprise principles* providing support for decision making on an enterprise level by informing how an organization seeks to fulfill its mission; *EA principles* reflect the spirit and thinking of the EA states and govern the EA management function as well as the implementation of its plans; *information technology principles* guide the use and deployment of all IT resources and assets. A general method how these principles can be developed as well as the responsible roles are provided by the E2A. Complementing best practice principles are presented [Sc08a, pages 238–255].

Recently, the IFEAD published a new and agile approach to EA management called STREAM – Speedy, Traceable, Result-driven Enterprise Architecture Management [Sc10]. The characteristics of STREAM are traceability of choices and decisions from the business side (traceable), focus on elements that directly contribute to the objectives (pragmatic), deliver results within a short time frame (rapid), deliver predefined type of results (productive), and always start at the business side and deliver significant value (relevant). The STREAM approach consists of five steps of which two address the current situation of Business and IT (step 1 & 2), two the future situation (step 3 & 4) and the final step the transformation plan. The steps are carried out in an agile way with iterations within and over the different phases.

The overall evaluation of the method-related prescriptions of the E2A are shown in Table 3.17.

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.17: Method classification for E2A

Considering the management body of the EA management function, the E2F emphasizes on the evolution aspect, i.e. that goals change over time and that the EA management function needs to be adapted in order to meet the new requirements (cf. Schekkerman in [Sc08a, page 58]. The EA according to the *Extended Enterprise Architecture Framework (E2AF)* is structured in four layers, the *business*, *information*, *information-system*, and *technology infrastructure* architecture. Thereby, managing the interactions between the elements that make up the different architectures provides valuable means in enterprise transformation [Sc08a, page 158]. Inspired by the Zachman framework, the E2AF proposes six questions providing different perspectives, which the EA should answer: *why*, *with who*, *what*, *how*, *whith what*, and *when*. In addition three aspects influencing the overall architecture are added *security*, *governance*, and *privacy* (see Figure 3.17).

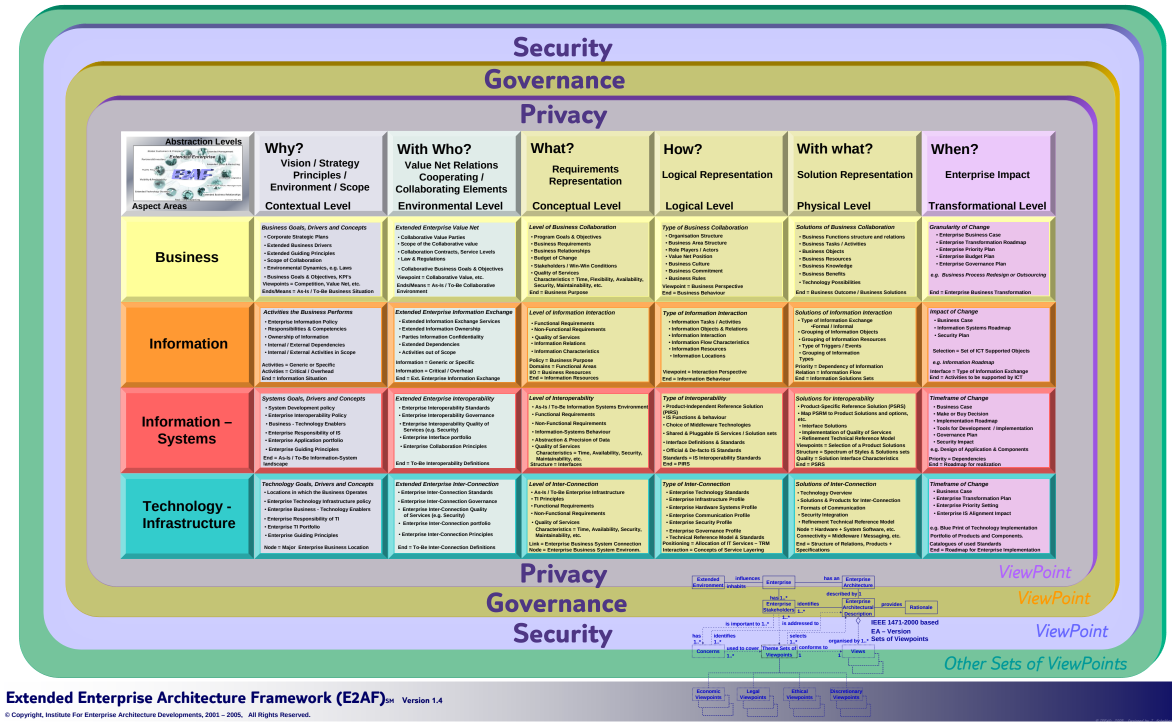

Figure 3.17: The extended enterprise architecture framework according to Schekkerman in [Sc06b]

While the role of projects and their relationships to the different elements that make up the EA is not directly alluded to, principles are mentioned on all architectural layers in the dimension why. In line with this structuring, the aspect of measurement can be approached answering the questions why to measure – on the business architecture – what to measure – on the information architecture – how to measure – on the information-system architecture – and with what to measure as well as when to measure – on the technology infrastructure architecture (cf [Sc08a, pages 159–170]). Thus, the E2A proposes the utilization of a goal-question-metric approach. In this sense, questions can be regarded as affecting elements on different architectural layers, contributing to an overall evaluation of the E2A and its language-related prescriptions as shown in Table 3.18.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.18: Language classification for E2A

3.10 The EA management approach of MIT

EA management approach	
Name of approach:	(Approach of MIT)
Issuing organization:	MIT Sloan Center for Information Systems Research (CISR)
Focus area:	Method
Tool support:	-
Period of activity:	since 2003
Publications:	[Ro03], [WR04], [RB06], [RWR06]
Inner organization:	monolith

In 1995, researchers at the Massachusetts Institute of Technology (MIT) started their work in the area of EA management. Their work focuses on the governance aspect of EA management (cf. Weill and Ross in [WR04]) and is mostly based on empirical surveys and case studies from industry. According to Ross et al. in [RWR06, pages 8–10], to execute the enterprises strategy a foundation consisting of an *operating model*, an *enterprise architecture*, and an *IT engagement model* has to be build. The operating model reflects the (envisioned) situation of the enterprise in respect to the two dimensions business process integration as well as standardization and therefore involves a commitment to the way the organization will operate. The EA provides a holistic view on the organization's business processes, systems, and technologies. The IT engagement model describes the governance mechanisms utilized to ensure the achievement of objectives, by coordinating decisions from business and IT and linking the enterprise-level management functions. Figure 3.18 illustrates how the three disciplines work together to create a foundation for execution.

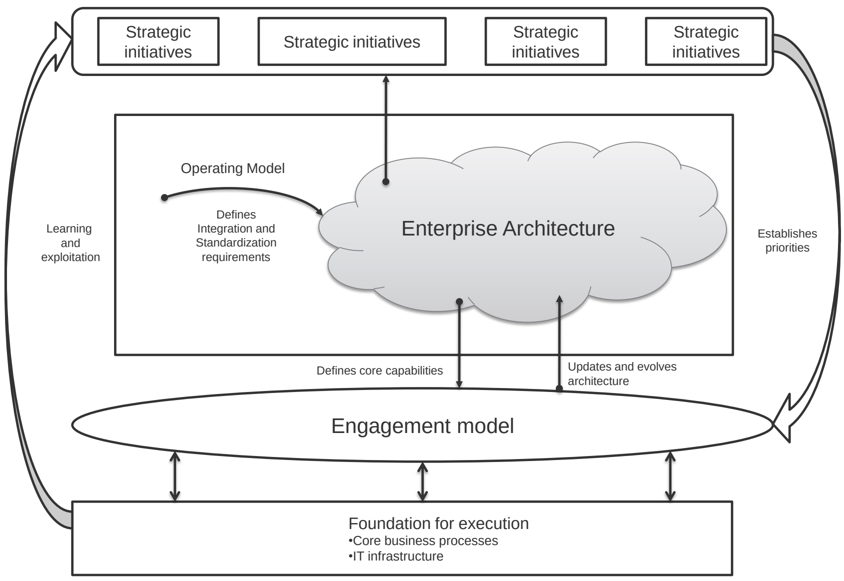

Figure 3.18: The foundation for execution according to Ross et al. [RWR06, page 10]

The four different types of operating models as introduced by Ross et al. in [RWR06, pages 28–33] can be interpreted as different types of organizational context and goal descriptions, which define requirements how the EA management function has to be configured. These requirements and influences on the design of an EA management function are exemplified by the authors via the description of case studies. Nevertheless, no general procedure how to configure the EA management function to the specific needs of a specific organization are detailed.

Ross et al. propose in [RWR06, page 195] a six step iterative approach to design and revise an EA management function, which read as follows: 1) Analyze your existing foundation for execution[10], 2) define your operating model, 3) design your EA, 4) set priorities, 5) design and implement an IT engagement model, and 6) exploit your foundation for execution for growth.

In order to define an operation model, one of the four different types has to be chosen, thereby, different operation models may apply for distinct parts of the organization. In the second step, Ross et al. propose to develop an "envisioned state" of the EA based on a so-called *core model*. A core model describes the relevant elements of the organization, which typically contain four common elements: business processes, data, technologies, and customers. For each operating model, Ross et al. propose a 'best practice' core diagram, which can be used as starting point for the development. Figure 3.19 shows the core diagram for the operating model unification. Each core diagram requires organization-specific adaptation and specification, i.e. the relevant elements have to be defined in an iterative collaboration process by the respective managers, e.g. senior managers or IT leaders [RWR06, pages 65–67].

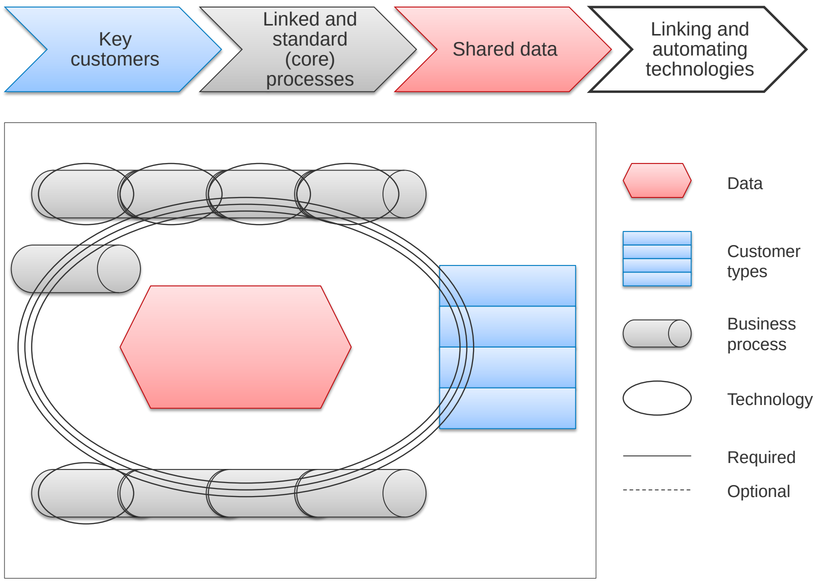

Figure 3.19: The unification core diagram according to Ross et al. [RWR06, page 54]

While the approach presented by Ross et al. in [RWR06] discusses general aspects of EA management, no detailed method description on how to perform the documentation, communication, or analysis tasks are presented. Nevertheless, the presented case studies from

[10] This step is the starting point in the first iteration, during initialization the method starts with the second step.

industry sketched such methods as e.g. during the Toyota case study, for instance, the concept of architectural principles is introduced and different methods, e.g. incentives, funding, or enforcement, to enact the principles are discussed [RWR06, pages 130–135]. For the communicate & enact activity, Ross et al. have derived key principles of successful engagement based on case studies from eighteen organizations, which can be seen as hints for the design of an organization-specific communication method [RWR06, pages 135–136].

In line with the idea of an organization as a vivid system, Ross et al. propose four stages of EA management maturity [RWR06, pages 71–79], which provide a path for the development of the EA management function. Considering the scope and reach of the EA management function, the maturity stages are defined as:

- *business silos architecture*, i.e. focusing the IT investments on individual business units needs,
- *standardized technology architecture*, i.e. shift from local optimization to global optimization via centralization of technology management and establishment of standards,
- *optimized core architecture*, i.e. shift from local applications and share data to enterprise systems through organization-wide data and process standardization, and
- *business modularity architecture*, which enables strategic agility through reusability of loosely coupled IT-enabled business processes based on global standards [RWR06, pages 71–79].

Ross et al. discuss implications which emerge while moving from one step to the next and further present different architectural elements, tasks, and responsibilities that have to change during the maturity process [RWR06, pages 79–86]. Furthermore, Ross presents case studies utilizing the maturity stages to evolve their EA management function in [Ro03] discusses lessons learned. Ross et al. further discuss the utilization of the operating models and maturity stages in different application contexts, e.g. merger and acquisitions [RWR06, pages 176–182] or outsourcing [RB06].

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.19: Method classification for the approach of MIT

As the research findings of Ross et al. in [RWR06] focus on governance and methodological aspects of EA management, the complementing language aspect is only casually referred to. Projects, for instance, are introduced by Ross et al. as means to implement change and to advance architecture maturity [RWR06, page 112] but their impact on dedicated elements of the EA is not discussed. Similarly, goals of an EA management initiative, e.g. increased IT responsiveness, improved risk management, or increased management satisfaction are listed in [RWR06, pages 91–101] but not linked to the elements that make up the EA.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.20: Language classification for the approach of MIT

3.11 The EA management approach of TU Lisbon

EA management approach	
Name of approach:	(Approach of TU Lisbon)
Issuing organization:	TU Lisbon
Focus area:	Modeling
Tool support:	-
Period of activity:	since 2003
Publications:	[Va01], [VST03], [Va04], [VST05], [Ca07], [MaZT07], [VST07], [AMT08], [VST08], [AST10a], [AST10b], [Av10], [CST10], [MZT10], [Za10]
Inner organization:	implicit organization

The research group of José Tribolet at TU Lisbon has a long history in what they call in their publications "information system architecture" (ISA) (cf. [VST03, pages 78–79]). In the recurring definition information system architecture is regarded as architecture on an intermediary level between enterprise architecture and software architecture laying a focus on "the representation of the IS components structure, its relationships, principles, and directives, with the main purpose of supporting business". With this broad definition and in the light of more recent publications of Vasconcelos et al. [VST07] and of Aveiro et al. [AST10b], it is sensible to reconcile the group's research as contribution to the field of EA management. In these publications further more emphasis is laid on method aspects related to EA, whereas the focus of the work initially was on modeling ISAs. For supporting concise modeling, Vasconcelos et al. propose in [VST03, page 79] the so called "CEO framework" that introduces a high-level meta-model for describing ISAs, subsequently refined in [CST09] to an embracing framework targeting EAs in five views[11]: an *organizational view*, a *business view*, an *information view*, a *system's application view*, and a *system's technological view*.

The approach of TU Lisbon presents itself in close relation to established disciplines as business process modeling and management, goal modeling, as well as information system modeling and development. Motivated by the missing links between these disciplines (called "matching problem" in [Va01, page 71]) the approach outlines a triple of "ISA modeling", "ISA evaluation", and "IS/Business alignment assessment" (cf. [Va04]) as key activities necessary for solving the problem. The early work with its strong emphasis on modeling-related aspects abstains from detailing actual steps and tasks for performing these activities, but gives some abstract indications, e.g. on how to adapt analysis methods from related disciplines like the architecture trade-off analysis method (ATAM) (cf. [VST05]). This idea picked up in [VST07, pages 95–111], where more concrete analysis prescriptions are given as specific metrics, which are described via a uniform template called "ISA metric template". In later work, namely [MaZT07, pages 62–63] the authors discuss on the importance of ISAs and EAs as means of communication and elaborate on the stakeholder-specificity of communication methods reflecting "well-articulated preferences [of users]". Pointing in a similar direction, they provide a statement on the "different perspectives and viewpoint from which the company is considered" in [VST07, page 92]. In [MaZT07, pages 66–67] also the necessity to "co-evolve" the describe organization and the descriptive methods is briefly discussed along the "boundary"

[11] According to the terminology adopted in our work, the term *viewpoint* would be more appropriate here.

nature of the EA inviting “reflexivity about the organization”. While these statements may be regarded as indications towards the need to configure and adapt EA management methods as well as EA description languages, more detailed descriptions are not provided as part of the approach. In [AMT08] Aveiro et al. discuss on the topic of communicating architecture states via semantic wikis, delineating the need to distinguish between current, planned, and target states therein. The recent publication of Aveiro et al. [AST10b] furthers these discussions and develops the abstract indications on management methods towards more detailed prescriptions on activities in “operational engineering”, namely “(Re)generation, Operation and Deletion of the enterprise”. Central to these considerations is a “viability” perspective on the enterprise, regarding the associated management processes as means to ensure viability of the organization by resolving dysfunctional interplay, i.e. to keep the organization working as intended in a corresponding to-be model. Where the work [AST10b, page 157] stays to abstract descriptions of the contained feedback loops “in tune with the well known Plan-Do-Check-Act (PDCA) cycle”, more concrete prescriptions can be found in [AST10a, pages 231–233]. A generic “exception handling cycle” presented there and complemented with an organization-specific “monitoring, diagnosis, exception and recovery table” adds some detail on how viability of the enterprise system may be ensured taking a holistic, i.e. EA, perspective. The focus of the approach is nevertheless on modeling the feedback loops in such systems by providing a meta-model capable for describing actions taken as well as actors and resources involved (cf. Aveiro [AST10a, page 238]). With the intention to generically cover organizational feedback and control processes, no actual prescriptions are made in respect to the tasks and responsibilities involved in activities for ensuring viability. Aveiro nevertheless builds on these prefabrics in [Av10] and establishes methods for deriving planned states and for establishing EA principles. Regarding the question of the interplay between EA management and related management functions, Caetano et al. introduce in [Ca07] the concept of the *competency*. Competencies are therein understood as classifications of human actors according to their ability of performing certain tasks. Based on these classifications, the actor’s involvement in management activities comprised of several tasks can be discussed. Furthering this understanding, Marques et al. explain in [MZT10] the complex net of interlinked competencies necessary to manage on an enterprise level, thus giving strong indications on how to understand and establish integration between management processes. In the light of the strict requirements set in Section 2.2.1, we classify the approach of TU Lisbon as shown in Table 3.21.

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.21: Method classification for the approach of TU Lisbon

The CEO framework introduced by Vasconcelos et al. in [VST03, page 79] is mirrored in a meta-model profile for the UML [Ob10b] covering three “sub architectures” (cf. Vasconcelos et al. [Va01]), namely the “informational architecture” containing business relevant data types, the “application architecture” describing supportive applications, and the “technological architecture" representing the technologies used for application implementation. This triple of IS subarchitectures is embedded into the context of the EA (cf. Figure 3.20) by establishing links to business goals, business processes and resources in an orthogonal manner. The corresponding basic meta-model (cf. Figure 3.21) establishes the abstract notion of the Block to represent ISA concepts of any sub architecture. Further concretizing this abstract representation of ISA concepts, Vasconcelos et al. denote in [VST03, pages 79–81] different IS and IT concepts as "applications" or "platforms" but also subtype blocks to concepts providing a more functional perspective on the IS sub architecture as e.g. "business service" or "IS service".

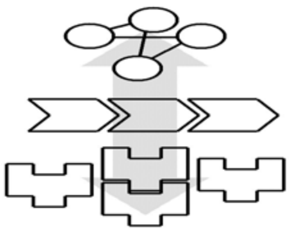

Figure 3.20: Goal/process/system framework according to Vasconcelos et al. [Va01, page 72]

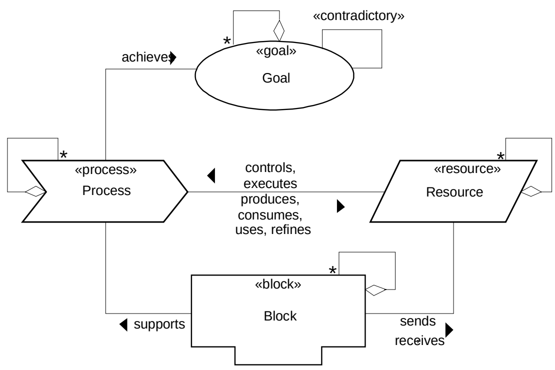

Figure 3.21: Meta-model of CEO framework according to Vasconcelos et al. [VST03, page 79]

The CEO framework does not impose restrictions on the subtypes to be used, although the different techniques and metrics described in related publications target only a subset of the provided types (cf. [Va04], [VST05] and [VST07]). In this sense, one might argue that the approach of the CEO framework is retains a notion of configurability but does not make prescriptions on when to use which of the subtypes. Building on the concept of the Block and its subtypes an exemplary case in [VST03, pages 81–83] outlines how current states of the architecture as well as target states thereof can be represented. The embracing perspective on structural aspects of the EA ranging from business to IT as well on functional aspects of the intermediary aspects of the IS is reverberates through the more recent publications of Zacarias et al. [Za10] as well as Caetano et al. [CST10]. In [Za10] the goal and resource perspective on business processes (cf. Figure 3.21) is furthered with a concrete understanding of input and output factors as well as associated agents taking a particular role in the execution. Based on such functional view on business processes, Caetano et al. [CST10] describe mechanisms for decomposing processes into sub-processes that are independently embedded into their organizational, motivational, and operational contexts. In [Va04, VST05] Vasconcelos et al. discuss the concept of architecture analysis. The analyses target different aspects of the ISA with [Va04] putting emphasis on metrics for evaluating "IS/business alignment", whereas [VST05] discusses the applicability of 'classical' software architecture metrics, as "lack of cohesion" on ISA level. In [VST07] Vasconcelos et al. raise a multitude of relevant quality attributes, reified in corresponding questions linking to a subset of the 16 concrete metrics presented therein. For each metrics actual computation rules based on an according EA description language are provided. A comprehensive overview on the relationship between different metrics and according "ISA qualities" (questions) is further provided by Vasconcelos et al. in [VST08], emphasizing that these metrics and questions cover all sub architectures, with functional analyses of the business perspective being discussed by Zacarias et al. in [Za10]. Further, the questions target generic quality attributes of an architecture and are hence not linked to architecture-related goals as modeled according to the CEO framework's meta-model (cf. [VST03, page 79]). In the recent publication [AST10b] Aveiro et al. delineate two information models, namely the "viability model space" targeting the management function concerned with EAs and the "GOD model space". Latter information model targets the "Generation, Operationalization, and Discontinuation of objects", i.e. of architectural elements. While the viability perspective is specifically concerned with adaptations to the management methods, the GOD perspective is capable of describing arbitrary Engineer Transactions that change the architecture or parts thereof. In this sense, the notion of the project is introduced into the canon of the description language in a cross-cutting manner targeting architectural elements from business to infrastructure level. In [AST10a, page 235] the notion is further developed and extended towards a lifecycle modeling for arbitrary architectural elements as well as for modeling the actors responsible for performing "Engineering Processes", i.e. changes to the architectural elements. A minor drawback nevertheless exists with this mechanism. In describing the GOD model space Aveiro et al. [AST10b, page 155] do not provide indications on how this information model is to be linked to the meta-model of the CEO framework. This may be ascribed to the focus of the work but leaves a willing user of the approach puzzled in a twofold way, not only as a description of the linkage is missing, but also as different underlying meta-languages (UML [Ob10b] and ORM [Ha05]) are used. Table 3.22 shows the classification of the approach of TU Lisbon according to the requirements delineated in Section 2.2.2.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.22: Language classification for the approach of TU Lisbon

3.12 The Systemic Enterprise Architecture Methodology (SEAM)

EA management approach	
Name of approach:	Systemic Enterprise Architecture Methodology (SEAM)
Issuing organization:	École Polytechnique Fédérale de Lausanne (EPFL)
Focus area:	Modeling
Tool support:	SeamCAD [LW06]
Period of activity:	since 2003
Publications:	[We03], [RBW03], [LW04b], [LW05], [LW06], [BW06], [RW06], [RW07], [We07a], [We07b], [We08], [Re09]
Inner organization:	explicit organization

The systemic enterprise architecture methodology (SEAM) roots in the work [We03] of Wegmann, where he elaborates on the multi-disciplinary nature (see also Rychkova et al. [RBW03], Lê and Wegmann [LW04b], etc.) of EA projects and the resulting need to support these projects with methods and models. Here thereby outlines the central method-model-dichotomy, also alluded to as "method-notation"-dichotomy by Rychkova et al. in [RBW03], of SEAM. Diagnosing a lack of support in respect to the method part of EA project support, SEAM seeks to complement existing approaches with additional methodical guidance. An example for such complementation is given by Wegmann et al. in [We08], where the Zachman Framework (cf. Section 3.1 and [Za87, SZ92]) is augmented with a "systemic conceptualization" based on SEAM. The systemic nature of SEAM-based conceptualizations makes up another predominant characteristics of the approach reverberating through the different publications. Wegmann formulates in [We03, pages 486–488] the underlying "systemic paradigm" based on the SEAM ontology, which itself builds on the prefabrics of RM-ODP [In96], and a dedicated epistemology, namely *constructivism*. In line with the constructivism principle, SEAM assumes that knowledge about a system is relative to the observer, meaning that no observer-independent descriptions of reality exist. By this fact Wegmann motivates in [We03, page 487] a hierarchical understanding of any system considering different "levels of reality" owned by dedicated stakeholders. This understanding reverberates through the work on SEAM, especially through the foundational language descriptions by Lê and Wegmann [LW04b, LW05], and is further mirrored in the methodology's tool support (SeamCAD) presented by Lê and Wegmann in [LW06].

Central to SEAM is the notion of the "EA project" as discussed by Wegmann in [We03, page 485]. Such project is initiated by an organization "to react to or to anticipate change" and starts with creating an "as-is model" reflecting the project-relevant entities. Complementing this model, a "to-be model" outlining the expected reaction to the change is created. Rychkova et al. describe in [RBW03, pages 11–12] a stepwise method for creating both the as-is and the to-be model. This method is recursive in its nature spanning different levels of abstraction in the system hierarchy, such that lowest level models provide "all necessary details for [subsystem] implementation". As part of the method gap analyses are to be conducted on each abstraction level, identifying one gap on each level such that a multi-level set of gaps has to be accounted for in finding the optimal design (cf. [We03, page 485]). Figure 3.22 taken from [We08] summarizes the cyclic design process as incorporated in SEAM.

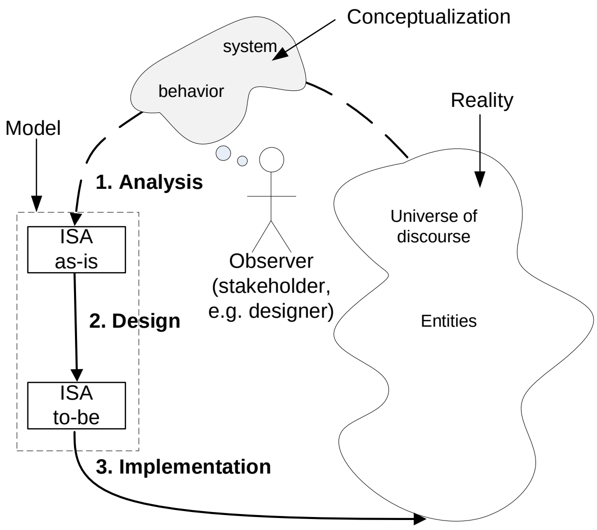

Figure 3.22: Basic design process of SEAM according to Wegmann et al. [We08]

Due to the fact that SEAM is applied in an EA project, the to-be models created in the design process actually reflect planned states for the EA or parts thereof. The project nature of the SEAM context is reflected upon by Wegmann et al. in [We07a, pages 397–398], where also a linkage to change and requirements management is briefly alluded to. Such linkage is mediated via graphical models of as-is and to-be, which are discussed in different works by Rychkova and Wegmann [RW07] as well as by Wegmann et al. [We07a]. The importance of stakeholder-specific models is further emphasized by Wegmann et al. in [We07b, pages 118–119], where they – refraining the systemic paradigm from [We03] – delineate that different designers utilize different "views" representing relevant parts of the overall system. These discussions are complemented with some remarks on the ways for developing and designing SEAM models namely via workshops or using collaborative tools. In an earlier publication [BW06] Balabko and Wegmann reflected on the topic of methods for designing architecture models. As part of this reflection they analyzed various methods from different disciplines in respect to their suitability for designing as-is and to-be models on various hierarchy levels. Based on the analysis' results an EA project can select a method well-suited for the specific design purpose, although Balabko and Wegmann abstain from giving details on how to integrate different methods. Further concretizing the analysis of SEAM's basic design process (cf. Figure 3.22) Rychkova et al. propose in [RBW03] a conceptual groundwork for analyses. Thereby, they complement the SEAM notation with an operational semantics describe in terms of abstract state machines (ASMs). More precisely, they outline transformations for translating a SEAM model into a model of an ASM. The underlying idea is concretized and furthered by Rychkova and Wegmann in [RW06], where the transformation is rewritten based on the ASM description language of AsmL[12] via an "AsmL interpretation of SEAM graphical models". Based on the executable ASM description as well as the notion of "behavioral substitutabiliy", Rychkova and Wegmann devise a method for verifying the alignment of a system design and the according behavioral requirements. Complementing these behavioral verification, Wegmann et

[12]For a description of AsmL see `http://research.microsoft.com/en-us/projects/asml/`.

al. exemplify in [We07b, pages 113–118] three methods for analyzing the planned design with from a customer and an organizational perspective, critically relying on belief vocated by the system's stakeholders. This analysis perspective is revisited by Regev et al. in [Re09], where further a classification of stakeholders in "favored", "disfavored", and "ignored" ones is introduced. The classification establishes a conceptual basis for defining key qualities of any system such as "utility" or "risk" via the existence or absence of a perception for different stakeholder groups. Table 3.23 summarizes the key characteristics of SEAM classified against the background of the method framework from Section 2.2.1.

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.23: Method classification for the Systemic Enterprise Architecture Methodology

A key characteristics of 'the' modeling language associated with SEAM is outlined by Wegmann in [We03, page 488] as part of the ethics of the systemic paradigm. He states that each concrete EA project needs to develop an actual enterprise model. In this sense the language prescriptions of SEAM are formulated on an abstract level allowing the project team to reify them and thereby to configure the language as needed. Central aspect of configurability in respect to the language is the intrinsically hierarchic understanding of the enterprise system reverberating through the method, see e.g. Rychkova et al. [RBW03] or Lê and Wegmann [LW04b]. In the latter article, foundational theories for the language are outlined, namely the "Living System Theory" of Miller [Mi95] and the conceptualization of RM-ODP [In96]. Based on these theories a set of three basic concepts Computational Objects, Information Objects and Actions of EAs is devised. In subsequent work of Lê and Wegmann [LW05] these concepts are complemented with a textual description of their semantics as well as a formalization based on the Alloy 2.0[13] language. At this point in time a different formalization perspective on the SEAM language had already been outlined by Rychkova et al. in [RBW03], being concretized in the subsequent work [RW06] of Rychkova and Wegmann. This work describes an executable ASM interpretation of the language concepts. The complete information model underlying the SEAM language (cf. [LW05, page 185]) puts special emphasis on the hierarchical nature of the according descriptions by annotating the layer-spanning relationships via specific UML stereotypes. This reflects the aforementioned dimension of configuration, in which a using EA project team has to find an appropriate level on its own. While only implicitly alluded to in the information model specification, the same language is used to describe as-is and to-be states of the underlying system building on two different kinds of abstractions (cf. Wegmann et al. [We05, WRL05]): an "organizational" and a "functional" one. Where the functional abstraction is used to specify **what** the system under consideration does or is intended to do, the organizational perspective details **how** this functionality is provided internally. The linkage between these perspectives further allows to conduct gap analysis between actual and intended function abstracting from

[13]For additional information about Alloy see `http://alloy.mit.edu/`.

implementation specific details. The ASM-based technique for performing such analyses, as described by Rychkova and Wegmann in [RW06], introduces further language concepts, namely relationships between Actions and Properties, of which the latter reflect specific qualities of Computational Objects. The three types of relationships are concretized by Rychkova and Wegmann in [RW07] making use of the Business Process Modeling Notation (BPMN) [Ob10a] to model action-action-relationships as well as of a pre- and post-condition based technique for action-property-relationships. In this context, post-conditions are used to describe property changes mediated by the according action. Finally, two types of property-property-relationships are introduced: "part-of" and "used", reflecting hierarchical (inter-layer) and lateral (intra-layer) dependencies between properties, respectively. In the most recent publication [Re09] of Regev et al. SEAM models are revisited against the background of four generic questions "utility", "warranty", "value" and "risk". Each of these questions is stated on an abstract and general level, for which a using project must find a specific operationalization into concrete metrics. Table 3.24 summarizes the key characteristics of the description language employed by SEAM classified against the background of the framework from Section 2.2.2.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.24: Language classification for the Systemic Enterprise Architecture Methodology

3.13 Archimate

EA management approach	
Name of approach:	ArchiMate
Issuing organization:	Telematica Institute / Novay
Focus area:	Modeling
Tool support:	ArchiMate Workbench [La05, La09]
Period of activity:	since 2003
Publications:	[Jo03], [Jo04b], [La04], [La05], [Jo06], [Ar07], [La09], [Jo10], [QEJ10] and `http://www.opengroup.org/archimate/doc/ts_archimate/`
Inner organization:	monolith

The ArchiMate modeling language for EAs has a long development history starting with the early work [Jo03] of Jonkers et al., who therein outline the key requirements and principles of what would later become a "language for coherent enterprise architecture descriptions". In particular, they introduce a notion of flexibility in respect to the model, plurality in respect to visualizations as well as viewpoints, and integrability with respect to existing modeling documentations. Building on this basic understanding, Jonkers et al. describe the three core aspects of an enterprise that any suitable modeling language should account for, namely "structure", "information", and "behavior". For each of these aspects as well as for the three relevant layer, namely "business", "application", and "technology", the ArchiMate modeling language provides appropriate concepts and conceptualizations. Figure 3.23 summarizes the "architecture framework" behind the ArchiMate language as defined by the aforementioned aspects and layers.

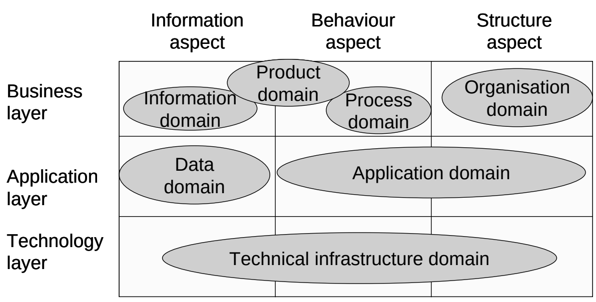

Figure 3.23: The ArchiMate architecture framework according to Jonkers et al. [Jo03]

Ever since the early days ArchiMate has become more than the modeling language, it was initially meant to be. Already the first edition of Lankhorst's book "Enterprise Architecture at Work" [La05] embeds the modeling language into the context of a comprehensive set of related methods and guidelines. The work further describes tool support for ArchiMate EA modeling using a tool called "ArchiMate Workbench". In its most recent edition [La09] the book presents an even more mature ArchiMate EA management approach, which has in April 2009 also been adopted as ArchiMate 1.0 specification by The Open Group[14]. Both most recent publications – the ArchiMate 1.0 specification and the book of Lankhorst [La09] – cover most of what can

[14]The ArchiMate 1.0 specification is online available at `http://www.opengroup.org/archimate/doc/ts_archimate/`.

be called the 'ArchiMate approach'. Where appropriate, we nevertheless add references to not so recent works of the group, especially when additional or more in-depth information on a topic is available therein.

In [Ar07] Arbab et al. describe the "architecture life cycle" that should be supported by appropriate architecture models. Similar versions of this life cycle can also be found in [La05, pages 46–47] and in [La09, pages 49–50]. This life cycle consists of the phases a) design, b) communication, c) realization, and d) feedback. Especially with respect to the communication of architectures, the importance to represent architecture aspects relevant to particular stakeholders is repeatedly discussed e.g. in [Jo03, Jo04b] and especially emphasized upon by Jonkers et al. in [Jo06]. This central communication challenge is further mirrored by Lankhorst et al. in [La04], where he deems EA modeling to be an "issue of integration" that has to bring together information from many different sources, and as Lankhorst later puts in [La09, pages 12–22] related governance instruments. Adding more detail to answer the question how EAs can be communicated in an appropriate and stakeholder-specific way, Lankhorst introduces in [La09, pages 80–84] the notion of the "architectural conversation". Such conversations are employed to proliferate architectural knowledge respecting the scope and perspective of the intended audience. Further, conversations relate to specific knowledge goals as "introduce", "agree" or "commit", of which each demands for a specific conversation technique. Making the relationship between knowledge goal and conversation technique more explicit, Lankhorst presents a "suitability matrix" in [La09, page 83]. In a similar sense, Lankhorst adds an in-depth discussion on how to select and adapt viewpoints for creating stakeholder-specific visualizations. Special attention is thereby paid to stakeholder "commitment", i.e. stakeholder awareness and agreement on the possible social implications of a certain viewpoint (cf. [La09, page 171]). Central to this discussion is the understanding that every viewpoint creates transparency with respect to a certain part of the organization, meaning that stakeholders responsible for this part might be 'overseen' by ones having access to according visualizations. In line with this argumentation, Lankhorst discusses on the topic of "scoping" viewpoints to convey the information needed to perform certain activities but not necessarily more. Exemplary viewpoints related to dedicated EA stakeholders and possible activities are delineate in [La09, pages 176–194]. First outlined by Jonkers et al. in [Jo03], techniques for analyzing EAs represented in corresponding models are detailed both in [La05] and [La09]. In the former publication Lankhorst discusses two different types of possibly interesting EA characteristics, namely "quantitative" and "functional" ones. Detailing on these concepts, he delineates quantitative methods for assessing "performance", "reliability" and "costs", where for the latter case process algebras for assesing the dynamic behavior of an architecture are introduced. In the more recent edition of the book [La09, pages 231–155], he extends the analyses' subject to "architecture alignment", which may be regarded an intrinsic property of the EA. More precisely, as alignment may be regarded a key goal of EA management the presented "guidelines regarding architecture alignment" as well as the complementing analysis techniques provide part of a governance structure for EA management, itself. In two recent Open Group whitepapers [Jo10, QEJ10] more detailed prescriptions on how to perform EA management are provided. The first whitepaper targets the linkage between EA management and the enterprise-wide requirements management processes, describing how change demands and information on the current state of the EA can be used to derive "architecture requirements" [QEJ10, page 8–9]. These are subsequently incorporated in a target state of the EA, from which in turn "realization requirements" are derived and finally converted to realization plans. This yields the linkage to the activities described in the second whitepaper [Jo10], which delineates a how projects and as well as their results are reflected in the EA management function. Relating the notion of the project from an EA management-perspective with the understanding of projects promoted by PRINCE2 [Of09], the presented techniques are useful as means of integrating EA management and project management. Summarizing the above, ArchiMate presents itself as approach with a strong model focus, whose method-related prescriptions are limited as indicated in Table 3.25.

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.25: Method classification for Archimate

In [Jo04b] Jonkers et al. call ArchiMate an "umbrella language" that is used to integrate different architecture-related design models from different languages. Central to this understanding is the flexibility of ArchiMate in respect to its level of specificity, while a remark puts that the ArchiMate language resides on a higher level of abstraction than the design languages. According to Jonkers et al. this ascribes to the fact that "a single language covering all domains [...] would probably result in an unworkable behemoth". Jonkers et al. discuss in [Jo03] key requirements for the ArchiMate language, of which flexibility in respect to the viewpoints as well as to the meta-model are deemed of highest interest. With respect to the former requirement, Jonkers et al. discuss a mechanisms to decouple visualizations for underlying EA models, making it possible to specify stakeholder-specific viewpoints. The latter requirement is addressed with a core meta-model consisting only of six concepts as shown in Figure 3.24.

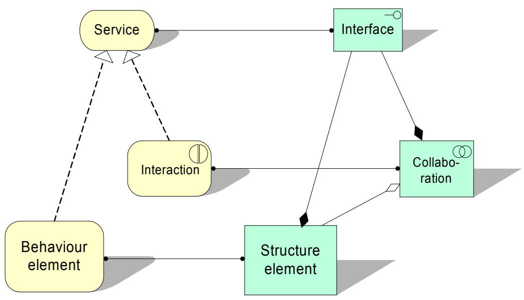

Figure 3.24: The core concepts of the ArchiMate meta-model according to Arbab et al. [Ar07]

As stated by Arbab et al. in [Ar07] these concepts reflect three central dichotomies of an EA, namely a) "internal-external" which resembles a white-box/black-box differentiation, b) "individual-collective", and c) "behavior-structure". In [La05, pages 90–105] and later in [La09, pages 91–106] Lankhorst gives a more detailed exposition of the meta-model, therein describing that the six concepts are subtyped on the three architectural layers of business, application and technology. Complementing what would otherwise turn out to be a language for describing layered architectures, Lankhorst further introduces a set of so-called "structural relations" that may be used to link structural elements on arbitrary layers. In addition, "behavioral relations" are introduced as means to link behavior on different architectural levels. These two set of relations may be considered a core source of ArchiMate's flexibility, as they allow to interrelate the layers even if no strict layering can be achieved. More precisely, an interaction element from the business layer can be directly linked to the technical collaboration that it stems from. On the opposite, this flexibility calls for responsible modeling. With respect to analytical methods applied on the model, Lankhorst describes in [La09, pages 206–208] a technique for annotating models with the necessary quantitative information, mirrored by corresponding attributes that augment the core meta-model. Based on these augmentations, he devises general rules and mathematical models that may be used to compute or derive quantitative information on one architectural layer from corresponding information on another layer. In particular, two sets of rules are described, namely one for "top-down workload calculation" and one for "bottom-up performance calculation". When it comes to the customization of the modeling language, both editions of Lankhorst's book [La05, La09] provide "guidelines for modeling", describing a preliminary phase "before to start" and a phase for deciding "what to capture" in the model. Complementing these content-related discussions, also guidelines on the visual appearance of the language are given by Lankhorst in [La09, pages 144–150]. There, topics of layout, utilization of color, and establishing a unique 'symbolic terminology' are accounted for. In the recent whitepaper [QEJ10, pages 14–15] Quartel et al. introduce additional language concepts that can be used to model goals and principles affecting the EA. For the former concept, a decomposition into "assessment", e.g. concretizing metrics, is incorporated into the language. The principles contrariwise remain on an abstract level and canot be concretized using standards or constraints. Another recent whitepaper [Jo10, page 9] specifically targets transformation aspects of EA management as reflected in the concepts of the "project" and the "project result", respectively. Using these means, the language can express planned states for the EA, although specific aspects of time are not incorporated. In particular, the language prescriptions remain limited with respect to the lifecycle of EA elements. Summarizingly, it can be said that the ArchiMate language presents itself as a comprehensive EA modeling language that is useful for describing current (and future) states of an EA. This leads to a classification of the ArchiMate language as shown in Table 3.26.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.26: Language classification for Archimate

3.14 The EA management approach of KTH Stockholm

EA management approach	
Name of approach:	(Approach of KTH Stockholm)
Issuing organization:	KTH Stockholm
Focus area:	Modeling
Tool support:	EA Tool [Ek09]
Period of activity:	since 2004
Publications:	[Ek04], [Jo04a], [GLS06], [JNL06], [JE07], [Jo07], [La07], [La08], [LJ08], [Nä08], [Bu09f], [Ek09], [KUJ09], [RNE09]
Inner organization:	monolith

The EA management approach developed at KTH Stockholm aims at providing decision support for IT management in enterprises, among others for the CIO, as key responsible for strategic IT-related decisions. Ekstedt et al. outline in [Ek04] this focus for the topic of EA, further relating it to the disciplines of software engineering and IS engineering, on whose methods EA management is required to build. With the focus on support for decision making, it is not surprising that the approach's contributions center around techniques and models for analyzing EAs with respect to specific qualities. Thereby, the approach of KTH Stockholm seeks to complement existing approaches for modeling EAs, such as the pattern-based approach to TU München (cf. [Bu09f]) or the ArchiMate modeling language [Nä08]. Complementing the work on evaluating architecture qualities, Ekstedt et al. outline in [Ek09] an analysis tool.

In [Ek04] Ekstedt et al. discuss the basic method for performing analyses on an EA-relevant level. A user of the method, e.g. the CIO, is required to prioritize the EA-related questions that should be answered, thereby assigning an expected utility to each question. From this, the set of required information is derived and complemented with estimates on the costs for gathering the information. Evaluating the cost/utility ratio for each question, an appropriate organization-specific information model is derived, corresponding information is gathered and the analyses are finally performed. Aforementioned steps may further be embedded into the environment of the method outlined by Johnson et al. in [Jo04a]. In the first step of the method, architects create "scenarios", i.e. modified versions of the current state of the architecture. Quality criteria for these scenarios are established in the second step and the scenarios are analyzed in the third step (both steps using aforementioned method). In a final step, one scenario is selected for implementation. This well aligns with the understanding of "IT management" as outlined by Gammelgård et al. in [GLS06, page 30], where three steps "understand", "decide" and "monitor" are outlined. In [JE07, pages 272–273] Johnson and Ekstedt further discuss methods for EA management, proposing a process as shown in Figure 3.25, in which step is assigned individual responsibilities and participants. The process os further linked with typical processes of IS management, as reflected in the CobIT guidelines [IT09]. In particular, CobIT-specific artifacts are denoted as input and output artifacts of the different process steps.

A consolidated view on the method-related prescriptions provided by the approach of KTH Stockholm is taken by Källgren et al. in [KUJ09], where guidelines for constructing a company-specific "EA model framework" are delineated. The guidelines reflect the basic idea of Ekstedt [Ek04] and are constituted of three major steps "make EA categorization", "identify desired

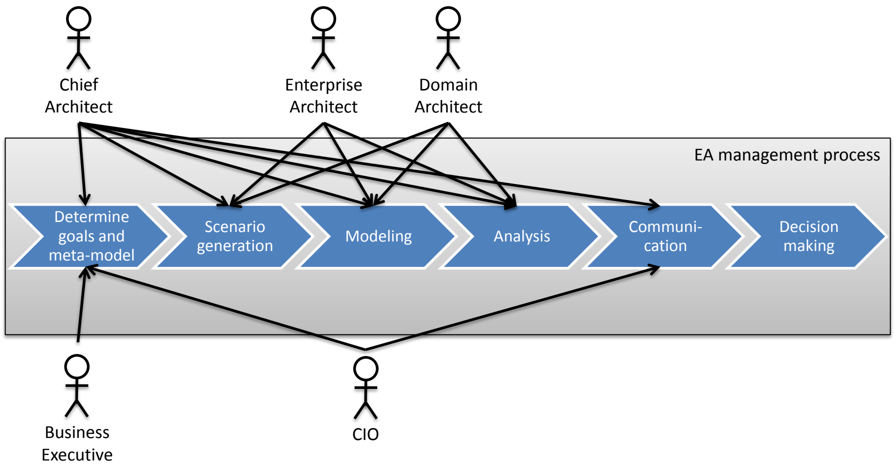

Figure 3.25: The “normative” EA management process of Johnson and Ekstedt [JE07]

information” and “finalize EA model framework”. In the first step the relevant business and IS goals for EA management are selected, which are in the second step linked to the relevant EA stakeholders. In the third step the appropriate viewpoints for the EA management function are selected, i.e. using the list of viewpoints provided by Johnson and Ekstedt in [JE07]. These viewpoints are further linked to the underlying information models, which are in turn related to existing information sources in the organization to develop a collection procedure for the integrated information model. The decision and control centric focus of the approach of KTH Stockholm is reflected in the classification as shown in Table 3.27.

INTEGRATION	unidirectional		bidirectional		
DEVELOP & DESCRIBE	current	planned	target	principle	question
COMMUNICATE & ENACT	current	planned	target	principle	question
ANALYZE & EVALUATE	current	planned	target	delta analysis	
CONFIGURE TO	organizational context		scope and reach		
ADAPT TO	organizational context		scope and reach		

Table 3.27: Method classification for approach of KTH Stockholm

The KTH approach specifically accounts for the cost/utility ratio of EA-related information and does hence not aim to provide a comprehensive information model. Ekstedt et al. provide in [Ek04] a core information model, linking business-level concepts as “business processes” and “organizational units” to IS-related concepts, e.g. “software components”. The model is nevertheless complemented with guidelines for utility-based adaptation, specifically highlighting that every using organization has to select the concepts with highest utility compared to the costs. In [JNL06, Jo07] Johnson et al. introduce a key concept of their approach, the so-called “extended influence diagram”. Building on these diagrams, Johnson et al. devise a technique that can be used to describe quality characteristics mirroring goals and relate these abstract characteristics to measurable, i.e. operationalized, characteristics of the architecture. In particular, the influence diagrams can be used to designate which architectural characteristics may be influenced by management activities, i.e. are “controllable”. Exemplifying, how the technique may be used to operationalize the architectural characteristic of “maintainability”, Lagerström provides in [La07] an exemplary extended influence diagram and derives a corresponding information model for maintainability analysis. This model is strongly system-centric, i.e. describes application systems in the context of the maintenance processes, the operating platform and the available documentation. In [LJ08] Lagerström and Johnson further detail this model, in particular the maintenance processes into “change activities” on system and component level. In more recent publications, the notion of the influence diagram is replaced with the more formalistic understanding of the “probabilistic relational model” (cf. Getoor et al. [Ge07]). This modeling technique based on conventional relational models allows go beyond the influence diagrams by making explicit uncertainty with respect to the contained information. The publications of Raderius et al. [RNE09] and of Sommestad et al. [SEJ08] show the applicability of a technique on the architectural characteristics of “availability” and “security”, respectively. Both publications provide specific information models reflecting the characteristics constituting the relationship modeling. Närman et al. take in [Nä08] a more general perspective on different architectural qualities using Bayesian networks to complement the relationship models, which are themselves based on the ArchiMate architecture modeling language (cf. Section 3.13). In [JE07] Johnson and Ekstedt supply a set of information models corresponding to different architectural viewpoints, e.g. an “organization viewpoint”, a “business process viewpoint” and “application usage viewpoint”, of which the latter introduces a service perspective on applications. A further information model is concerned with goals, more precisely with their interdependencies. Understanding for the need to integrate information models originating from different sources, Lagerströ et al. describe in [La08] how Bayesian belief networks can be used to discover correspondences between such models. In particular, the networks can be used to find out, which classes originating from different information models may be identified with each others. The network-based technique can be employed in step three of the guidelines provided by Källgren et al. in [KUJ09], where the information models underlying the stakeholder-specific viewpoints are integrated into a comprehensive information model. In the light of above properties, the approach of KTH Stockholm is classified as shown in Table 3.28.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.28: Language classification for approach of KTH Stockholm

3.15 Building blocks for Enterprise Architecture Management Solutions (BEAMS)

EA management approach	
Name of approach:	Building blocks for Enterprise Architecture Management Solutions (BEAMS)
Issuing organization:	TU Munich
Focus area:	-
Tool support:	System cartography tool [Bu07a]
Period of activity:	since 2004
Publications:	[MW04], [FMW05], [LMW05b], [LMW05a], [se05], [Bu07b], [Bu07a], [Wi07], [Bu08a], [Bu08b], [LS08], [Bu09a], [Bu09c], [Bu09b], [BMS09b], [Bu09d], [BMS09a], [Ch09], [BMS10c], [BMS10b], [Bu10a], [Er10]
Inner organization:	explicit organization

The EA management research at TU Munich is rooted in the research project "software cartography" dating back to [MW04], in which Matthes and Wittenburg motivate the need for a description technique targeting the entirety of the applications, the "application landscape" in an enterprise. This technique, software cartography, was further developed by Lankes et al. in [LMW05b] and an understanding of the multiple viewpoints on the application landscape was developed alongside the IEEE Std. 1471 by Lankes et al. in [LMW05a]. The year 2005 obviously marks a turning point in EA-related research at TU Munich, especially as the first "Enterprise Architecture Management Tool Survey" [se05] departs from the strictly visualization centric research undertaken since then. In particular, method and modeling related topics entered the center of attention.

Fischer et al. describe in [FMW05] the relationship between the EA management process and related enterprise-level management processes such as project portfolio management. They emphasize on the different artifacts that are exchanged between the processes in order to support comprehensive management of EA as a whole. Furthering the basic idea of the process linkages, Wittenburg et al. describe in [Wi07] the exchanged artifacts in more detail, identifying them with architecture visualizations conforming to a distinct viewpoint and grounded in a specific information model. Having identified manifold visualizations that are used in practice, Buckl et al. describe in [Bu07b] how patterns, so called "EAM patterns" may be used to document proven practice solutions in EA management. Following this paradigm, Buckl et al. collected the EA management pattern catalog [Bu08a] containing among other more than 40 so called "methodology patterns"[15]. Each of these patterns, as further exemplified in [Bu08b], denotes a typical EA management problem in a distinct context and describes a practice-proven solution. The initial version of the pattern catalog [Bu08a] originating from observations at more than 20 project partners of TU Munich was over the next year refined and augmented with additional patterns with methods for mergers [Bu09a], for defining an EA vision [Bu09c], and with EA management anti-patterns [Bu09d]. Latter patterns describe solutions that have proven not to work in practice. With the methodology patterns being observed solutions from a practical setting, the EA management pattern catalog does not

[15]As these pattern actually present management methods, they should be alluded to as "method patterns".

provide an overarching framework for composing these solutions or integrating them into a comprehensive EA management process. In [Bu09b] Buckl et al. show that such integration is nevertheless possible, linking the catalog's patterns to the phases of the TOGAF ADM (see Section 3.8). Exploring the relationships between patterns of the same type, i.e. between different methodology pattern, the pattern catalog is refined to a pattern language that in turn is presented as hyperlinked wiki [Ch09]. In [Er10] Ernst shows the applicability of the pattern approach for developing organization-specific EA management functions along different practice cases and discusses on techniques for integrating different patterns into a comprehensive function.

Building on the groundwork of the EA management pattern language, Buckl et al. explore a knowledge management perspective in [BMS09a] and a viable system perspective in [BMS09b] on the EA management function in order to devise an overarching method framework. In particular, they elicit a PDCA-like structure (cf. [De82a, Sh86]) that an EA management function typically commits to, defining four phases as follows: a) describe, b) implement, c) analyze, and d) adapt. These phases in turn lay the basis for a conceptualization that allows to understand redundancies in the pattern language, calling for a refined mechanism to structure practice-proven knowledge on EA management methods. Such mechanism is outlined by Buckl et al. in [BMS10c], introducing the term of the "EA management design theory" to structure context-specific, redundancy-free prescriptions for designing an EA management function. Furthering the idea, Buckl et al. delineate in [BMS10b] how the theories may be interlinked into a theory nexus that allows selecting the most appropriate design prescription for a given problem in a given context. In [Bu10a] Buckl et al. exemplify how design theories, so called "building blocks" there, for EA management functions can be derived from the methodology patterns of the pattern language and explain on how these building blocks may be integrated into a comprehensive EA management function. Refining the method framework mentioned above, Buckl et al. further introduce their structuring framework for the building block-based design of EA management functions as shown in Figure 3.26.

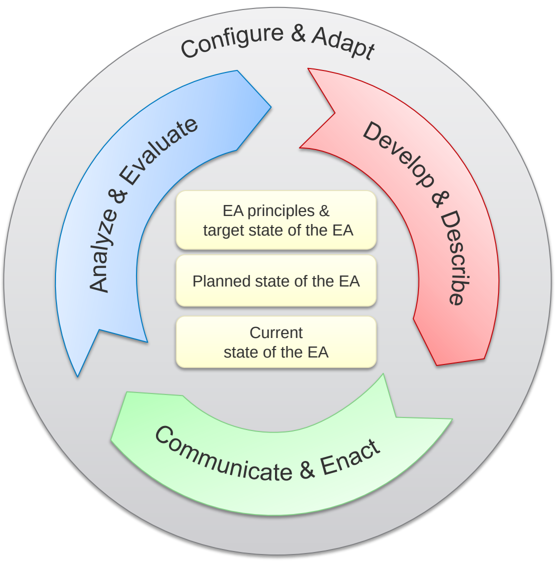

Figure 3.26: The EA management method framework of BEAMS, cf. Buckl et al. [Bu10a]
.

Summarizing the method related prescriptions provided by BEAMS, we classify the approach as shown in Table 3.29.

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.29: Method classification for BEAMS

The creation of visualization-specific information models is a recurring topic in the EA management approach of TU Munich, especially in the early days. Buckl et al. describe in [Bu07a] a technique for generating visualizations from underlying data. A prerequisite for this technique is the existence of a suitable information model that covers the information, which should be conveyed in the corresponding visualization. As Lankes and Schweda show in [LS08] information models may also be used as basis for quantitative analyses. In particular, they augment an information model describing the communication between business applications with additional attributes to reflect failure propagation and to derive failure probabilities in a given architectural setting. Mirroring the dichotomy of visualization-defining viewpoints and underlying information models the pattern catalog [Bu08a] describes practice-proven visualizations and the corresponding information demands. In particular, for each described viewpoint and each corresponding information model, the addressed problem, the intended context and the observed solution are described. Relationships between the corresponding patterns are further used to describe which viewpoints can build on the same information model, thereby outlining possible directions for evolving an existing EA description language by enriching it with additional perspectives. Another recurring topic of the approach is the modeling of the evolution of the EA. In [Bu08c] Buckl et al. discuss on the importance of linking projects to the architectural changes that they drive. Responding to this requirement, they derive in [Bu09e] an information model capable of expressing the change of the application landscape over time. Reflecting the way how project-dependency is introduced into the EA information model, Buckl et al. diagnose in [BMS10e] that "classic" object-oriented meta-modeling languages are not appropriately suited to cover the "true ontologic nature" of modeling project-induced change. In response, they advocate for the utilization of additional meta-modeling concepts, among other the "mixin" concept. A mixin allows adding certain attributes and relationships to a given concept without changing the very nature of this concept. Exemplified with the relationship between a project and the corresponding architecture element, Buckl et al. (cf. [BMS10a]) devise an information model fragment as shown in Figure 3.27.

Figure 3.27: The information model fragment for projects [BMS10a]

.

Complementing the ontologically concise modeling of projects, Buckl et al. further discuss in [BMS10f] and [BMS10a] how metrics and standardization-related concepts may be added to arbitrary information models. Thereby, they call on the design theory mechanism described above to refine composeable building blocks from the information model pattern of the EA management pattern catalog [Bu08a]. Building on the relationships between the original pattern, Buckl et al. further devise in [BMS10d] the concept of the "concern relationship" that allows relating different information models that cover a similar area-of-interest in the organization while varying in respect to the level of detail. These relationships are used by a method to evolve an organization's information model. In this light, BEAMS can be classified according to the framework from Section 2.2 as shown in Table 3.30.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.30: Language classification for BEAMS

3.16 Finnish Enterprise Architecture Research (FEAR)

EA management approach	
Name of approach:	Finnish Enterprise Architecture Research (FEAR)
Issuing organization:	University of Jyväskylä
Focus area:	Method
Tool support:	-
Period of activity:	since 2004
Publications:	[HP04], [PH05], [Pu06], [VS08], [IL08], [LHS08], [PNL07], [SHL09], [VSL09]
Inner organization:	monolith

The Finnish Enterprise Architecture Research (FEAR) at the university of Jyväskylä is conducted to support EA management in the Finnish public administration. It may be ascribed to this focus that central parts of results and documentations of the FEAR are published in Finnish, most notably the guidelines for "adaptation and adoption of Finnish government EA method" [VS08]. The research has notwithstanding contributed a lot of insights on the role as well as the challenges of EA management in the public sector, explored especially in a several empiric surveys, e.g. of Isomäki and Liimatainen [IL08] as well as of Seppänen et al. [SHL09]. Both surveys show nevertheless show that typical issues of implementing EA management in a company are also likely to be found in the public administration, most notably "insufficient support for the EA development" by the higher-level management, or "insufficient resources". Beyond these prevalent impediments to successful EA management Isomäki et Liimatainen [IL08] identify key goals of EA management in the public administration, e.g. "advancement of interoperability" between agencies, "shared understandings" for the relevant architectural concepts, and the establishment of a "shared IT infrastructure". For addressing these challenges in the public administration environment, FEAR devised an EA management method, which is according to Seppänen et al. [SHL09] based on a TOGAF variant 3.8. Central to the approach is the EA grid [HP04] that introduces four EA components as well as three levels of decision making, see Figure 3.28.

	Business architecture	Information architecture	IS architecture	Technology architecture
Enterprise level				
Domain level				
Systems level				

Figure 3.28: The EA grid of perspectives and levels of decision making

A high-level overview on the EA management process is given by Pulkkinnen an Hirvonen in [PH05], suggesting three phases as follows. In the "initiation" phase the target architecture is described and the expected benefits are delineated, which are complementingly mapped to the decision areas in the EA grid. The "working" phase details the requirements to actually needed changes and derives necessary courses of action. In the "ending" phase, relevant plans, designs and architecture evaluations are communicated in the organization, e.g. for being picked up by implementation projects. Furthering this idea, Liimatainen et al. devise in [LHS08] a framework for evaluating compliance of development programs with the EA. The framework consists of five steps, ranging from a general “elibility analysis” determining whether the program touches a relevant part of the EA to a detailed analysis of the program’s “operating model” with respect to its alignment with the EA management presecriptions. The overall make-up of the EA management process is described by Valtonen and Seppänen in [VS08] as summarized in Figure 3.29.

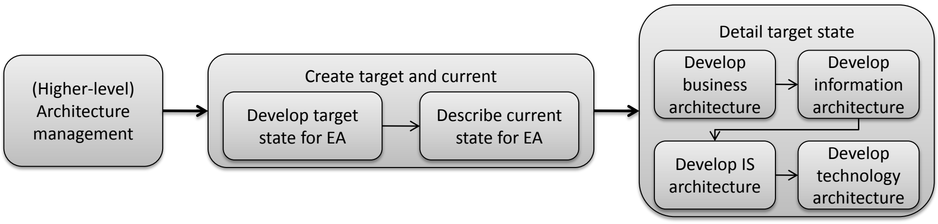

Figure 3.29: EA management process described in [VS08, page 42]

Building on the prescriptions of higher-level architecture management, the design phase develops both a target state of the EA and a description of the current state thereof. Complementing the strictly sequential creation method described in the figure, Valtonen and Seppänen describe in [VS08, page 40] that both states may be developed iteratively, further accounting for the fact that only parts of the current state should be documented which are relevant in respect to the target state. The method also reflects a hierarchical understanding of the EA, in which decisions on a higher level make prescriptions for lower level architecting. This aligns with public administration focus of the overall framework, such that national EA decision influence agency-wide EA decision that in turn influence project-wide decisions (cf. Valtonen et al. in [VSL09]). In the sense of the discussion of Liimatainen [LHS08], the method of Valtonen and Seppänen [VS08] elaborates on the importance to communicate and enact the target state of the EA as well as a corresponding transformation roadmap, although concrete prescriptions on how to do so, are scarce in the approach. For the specific aspect of information security, Pulkkinen et al. describe in [PNL07] steps to be taken in more detail along the example of an applying organizational. In particular, some indications on how to develop the transformation roadmap form a target security state are given. Complementing, Pulkkinen reflects in [Pu06] on the feedback nature of EA management, delineating that decisions taken on a lower level may have to be mirrored and fed back to higher levels in order to ensure a consistent management. This leads to an overall classification of the approach as shown in Table 3.31.

Integration	unidirectional	bidirectional			
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context	scope and reach			
Adapt to	organizational context	scope and reach			

Table 3.31: Method classification for the approach of FEAR

With FEAR having a focus on methods and processes for EA management, almost no language-related prescriptions are made. The EA grid as detailed by Pulkkinen in [Pu06] provides some indications towards the relevant aspect of an EA to be covered, exemplarily naming “products”, “applications” or “infrastructure platform”. More detailed prescriptions are thereby not provided. This is further mirrored in [VS08, page 48], where Valtonen and Seppänen describe how prevalent IS-related description languages, like the UML, as well as the ArchiMate (cf. Section 3.13) modeling language may be used to create descriptions, e.g. “system maps”, as part of the EA management method. Agsinst this background, we choose not to classify the language-related contribution of the FEAR approach.

3.17 Methodology for (re)design and (re)engineering organizations (DEMO)

EA management approach	
Name of approach:	DEMO
Issuing organization:	TU Delft
Focus area:	Method
Tool support:	-
Period of activity:	since 2005
Publications:	[Di05], [Di06], [LD08], [ED09]
Inner organization:	monolith

Against the background of over 15 years of practice, Dietz [Di06] has developed a "methodology for (re)designing and (re)engineering organizations" called DEMO. With its sound theoretical foundation in a theory called Ψ-theory the method takes a different perspective on the enterprise focusing on the so-called "enterprise ontology". Dietz uses this term to denote a "coherent, comprehensive, consistent and concise model of the essence of the enterprise". Critical to his definition is thereby the notion of "essence" that in the sense of Dietz targets the deep behavioral nature of the enterprise, but not realization and implementation specific details. In this sense, the model of Dietz abstracts from technical or infrastructural aspects of an enterprise, but stays to the "acts" that drive the enterprise's performance in its environment. Central to these considerations are the four axioms of Ψ-theory which themselves are grounded in the "initiator-executor"-dichotomy, distinguishing between the requestor of a service (execution) and the provider for the corresponding service:

- The *operation axiom* defines a distinction between two types of acts – production acts (P-acts) and coordination acts (C-acts).
- The *transaction axiom* defines transactions consisting of acts in different phases, namely an "order phase" of C-acts, an "execution phase" of P-acts, and a "result phase" of C-acts.
- The *composition axiom* defines multiple transactions may be hierarchical connected in the production of non atomic P-facts (production results).
- The *distinction axiom* defines three-facets of an action namely the "ontological", i.e. commitment, facet, the "infological", i.e. expressing and interpreting, and the "datalogical", i.e. communicating, facet.

Ettema and Dietz describe in [ED09] how the abstract and axiomatically grounded perspective of DEMO may complement more 'classic' EA management approaches as the one of Archimate (cf. Section 3.13). This especially applies during the execution of organizational change projects, where the DEMO may be used as reference point describing the stable essence of the enterprise.

In [Di06, pages 139–158] Dietz describes the six core activities constituting a successful application of the DEMO methodology. The activities are thereby subdivided into two distinct sets, namely "analysis" and "synthesis" activities. Dietz discusses in [Di06, page 143] that the activities not necessarily have to be executed in the order given below, but that an experienced user of the method may freely iterate through the described steps. For each activity, the method description gives examples of the steps to be undertaken an illustrates the artifacts that may be used as input or are to be created as output of the activities, respectively. As primary input to all different activities, Dietz calls on "all available documentation about the enterprise", which may in line with prior arguments be also elicited as part of a preparatory phase. In the *performa-informa-forma analysis* knowledge about the enterprise is in line with the distinction axiom divided in three sets, of which only the ontological item are fed forward to the next activity. The *coordination-actor-production analysis* applies the operation axiom to structure the ontological items into C-acts as well as C-facts, P-acts as well as P-facts, and actor roles. The thereby elicited elementals of the ontological essence of the enterprise are in the subsequent activity *transaction pattern synthesis* organized into singular transactions conforming to the transaction axiom. With the transactions as coarse grained elementals, the activity of *result structure analysis* decomposes non atomic P-facts into their components (P-facts) and hence establishes a hierarchy between the transactions. According to this hierarchy, the transactions are integrated in the activity *construction synthesis*, whereas further the role of the involved actors is accounted for and initiator-executor-relationships are formulated. In a final activity (*organization synthesis*) preparing an organization (re)engineering the previously devised model is partitioned into a part to be studied (object system – OS) and its environment (using system – US). Figure 3.30 depicts how the distinction between OS and US is critically reflected in the basic design process.

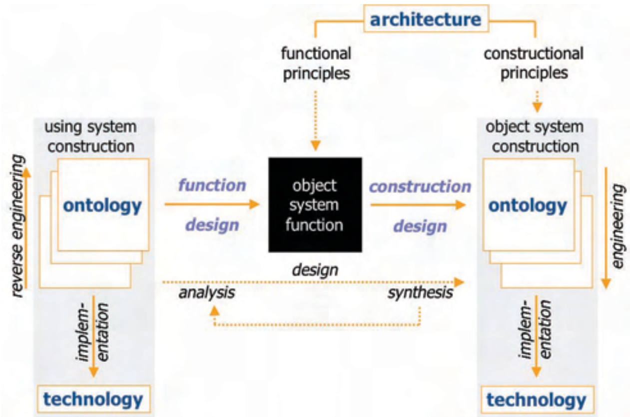

Figure 3.30: Basic design process according to Dietz [Di06]

With the successful application of the DEMO methodology the enterprise ontology and hence the ontologies of OS as well as US are derived. In a subsequent step these ontologies are taken as input for an actual design process that derives a new ontology for the OS from requirements regarding the function of the OS. Op't Land and Dietz exemplarily concretize an enterprise ontology-based design process for organizational restructuring in [LD08] prescribing actual steps to be taken on the ontological descriptions of the enterprise. The example highlights the role in which DEMO understands itself, namely as method for supporting change processes in an enterprise context via a theoretically and well-founded mechanism for abstracting the essence of the enterprise system. Against the background of DEMO's focus, we classify the method as shown in Table 3.32.

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.32: Method classification for DEMO

The method of DEMO provides an approach to develop enterprise ontologies in a systematic way [Di06, page 139]. Thereby, the ontology covers the essence of the organization under consideration, i.e. reflects commitment-related information, while it abstains from giving detail on implementation-related aspects. In line with the four basic axioms of Ψ-theory the ontological model of the enterprise is constituted of four distinct submodels as shown in Figure 3.31.

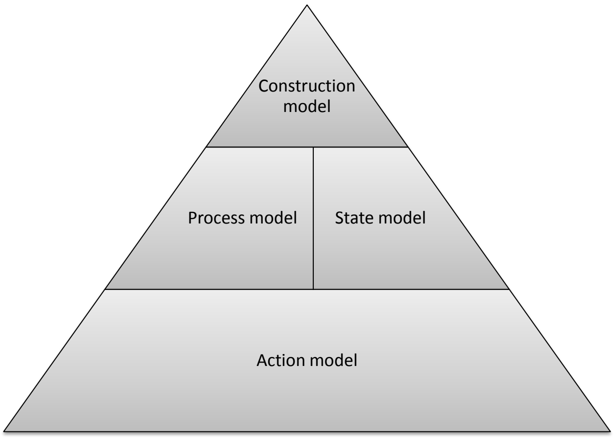

Figure 3.31: Basic design process according to Dietz [Di06]

The **construction model** specifies the construction of the organization embodied in the identified transaction types as well as actor roles. Detailing the coordination aspect of the transactions, the *process model* describes causal and conditional relationships between different transactions. Complementing this perspective the *state model* outlines the state space of P-facts, i.e. of production results, while P-acts are not part of the state model, as they may be derived from the corresponding process model. The *action model* describes the enterprise ontology on the most detailed model, such that – as Dietz states in [Di06, page 185] – the other models may be derived from the action model, and are hence only provided for ease of use. The different abstracted models (construction, process and state model) are complemented each with a specific description language, of which especially the language behind the state model deserves special attention. The so called "world ontology specification language" (WOSL) (cf. Dietz [Di05]) provides the basic language elements for describing *rigid* and *non-rigid* structures, i.e. states that exist universally over time and states that may change. Applying WOSL to the context of the enterprise ontology, Dietz refines rigid structures to C- and P-states, respectively, whereas C-facts (and P-facts, optionally) are derived from the general notion of the non-rigid structure. Thereby, WOSL provides a highly configurable and organization-specific way to establish a basic ontological description of the state space. The construction and process model languages present themselves as two sides of the same coin taking a blackbox and a whitebox perspective on the organizational transactions further mirrored in the prescriptive understanding of an EA complementing the enterprise ontology with "functional" and "constructional principles" (cf. Figure 3.30 and [Di06]). The specific focus of the DEMO method and its complementary description languages leads to a classification in respect to the language framework from Section 2.2.2 as shown in Table 3.33.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.33: Language classification for DEMO

3.18 The EA^3 Cube™

EA management approach	
Name of approach:	EA^3 Cube™
Issuing organization:	Scott A. Bernard, Syracuse University
Focus area:	Method
Tool support:	-
Period of activity:	since 2005
Publications:	[Be05], [Do08][16], [BG09], [Do09a], [Do09c], [Do09b]
Inner organization:	monolith

In [Be05], Bernard presents his experience gained through his work in practice and academia in the area of EA management. Identifying the need for a textbook for students, Bernard wrote *An Introduction to Enterprise Architecture Management* in which he presented the *EA^3 Cube™* approach. According to Bernard EA management a framework for EA management follows the dichotomy of language (*what*) and method (*how*) [Be05, page 75] and consists of six basic aspects:

1. an EA governance process, which links the the EA management function to other enterprise-level management functions,
2. a repeatable methodology describing the management function,
3. a framework representing the core elements and layers, i.e. the scope, of the initiative,
4. an integrated set of artifacts, i.e. architectural descriptions,
5. documentation tools with a repository to support architectural descriptions, and
6. associated best practices, which guide the implementation of the management function [BG09, page 220].

These constituents are further detailed in subsequently along the EA^3 Cube™.

In [Be05, pages 38–40] introduces a framework for EA descriptions – the EA cube, which consists of three dimensions (see Figure 3.32). The first dimension is concerned with the different architectural *levels* ranging from high-level strategic goals and initiatives to technical network and infrastructure aspects on the bottom. The *segments* dimension divides the overall EA in different parts covering one ore more lines of business, i.e. distinct areas of activity in the organization, from a holistic perspective, i.e. all architectural levels. Complementingly, the third dimension *artifacts* revers to the components that make up the organization. Thereby, vertical components that serve one line of business but may affect more than one architectural levels and horizontal, i.e. crosscutting components, which serve several lines of business, are distinguished.

The above introduced concept of cross-cutting components can not be put on a level with the idea of cross-cutting aspects as introduced in Section 2.2. As the cross-cutting components as introduced by Bernad in [Be05, page 40] relate to instances, while the cross-cutting aspects introduced in Section 2.2 related to the class level. Goals of an EA management endeavor, which represent a cross-cutting aspect in terms of the analysis framework, are referred to by

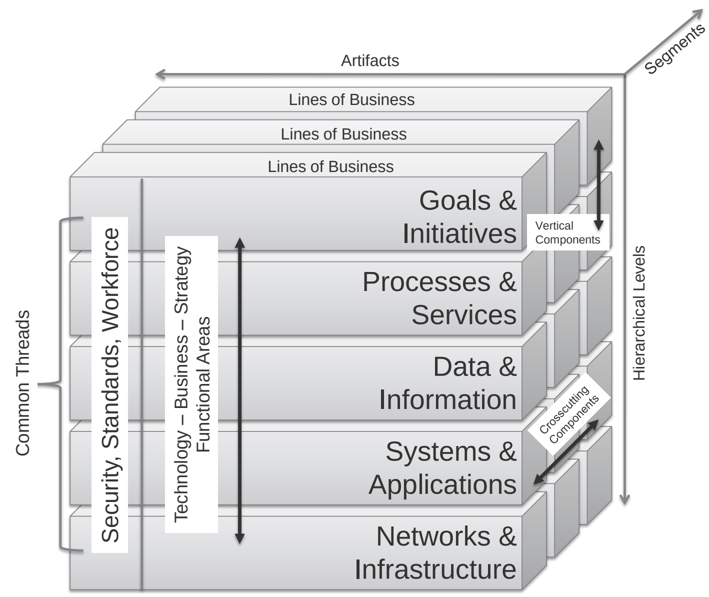

Figure 3.32: The EA^3 cube description framework [Be05, page 40]

Bernard in [Be05, pages 64–69] and their relation to supporting components of the EA is detailed [Be05, page 181]. Similarly, strategies, goals, and measures are discussed to quantify the EA management endeavor [Be05, pages 72–74] but only discussed from a methodological perspective.

The need to understand the organizational contexts, i.e. the "amalgamation of values, beliefs, habits, and preferences of all of the people throughout the enterprise" [Be05, page 56] to design a suitable EA management function is emphasized by Bernard. Besides a list of prospective key success factors for avoiding cultural misinterpretations, Bernard presents concrete examples for the impact of the organizational context on the design of an EA management function, e.g. he proposes a *segmented approach*, which is limited in scope for large or decentralized organizations [Be05, page 55] or he indicates that the schedule for updating architectural descriptions has to be defined [Be05, page 83]. While providing hints and suggestions, which method is suitable in which organizational context, he abstains from directly linking them or providing mechanisms for configuration.

INTEGRATION	unidirectional		bidirectional		
DEVELOP & DESCRIBE	current	planned	target	principle	question
COMMUNICATE & ENACT	current	planned	target	principle	question
ANALYZE & EVALUATE	current	planned	target	delta analysis	
CONFIGURE TO	organizational context		scope and reach		
ADAPT TO	organizational context		scope and reach		

Table 3.34: Method classification for EA^3 Cube Framework

In [Be05, pages 83–94], a 20-step process to implement an EA management function is introduced. The process can be grouped in four phases: Establishment of the EA management function, framework and tool selection, documentation of the EA, and use and maintain the EA. In the first phase, aspects as integration with other enterprise-level management processes, e.g. the project management and investment planning (cf. [Be05, pages 198–211]), as well as the configuration aspects in respect to the organizational context and the intended scope and reach (cf [Be05, page 87]) are discussed. Thereby, no explicit mechanisms how to perform this configuration are presented but their importance is accentuated and questions which guide the configuration are provided (cf. [Be05, page 83]). In the second phase especially the aspect of tool support and best practices for EA management is alluded to. Phase three is concerned with the development and description of current and future states of the architecture. Thus, Bernard distinguishes between two different types of future views, the long-term strategic future views (with a time horizon of 4-10 years) and the medium-term, planned, tactic views (1-3 years) [Be05, page 41]. The latter ones are derived from changes as described in the planned initiatives [Be05, page 160]. Thereby, also the importance of variant development and historization is alluded to (cf. [Be05, pages 160–164]). The *Concept of Operations* (CONOPS) (cf. Neilson in [Ne00]) is described as exemplary method to develop the future states. Complementing the time snapshots on the EA, an EA management plan representing a roadmap illustrating the transformation from the current to the future state is developed. This plan further includes the definition of roles, responsibilities [Be05, page 170] and boards and their competencies providing the link to other enterprise-level management processes [Be05, page 204]. While Bernard abstains from detailing methods in terms of tasks and actors to develop descriptions of the current and future state as well as roadmaps, he provides exemplary viewpoints how an EA description could look alike.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.35: Language classification for EA^3 Cube Framework

3.19 Dynamic Architecture for modelling and development (DYA)

EA management approach	
Name of approach:	DYA
Issuing organization:	Sogeti Netherlands
Focus area:	Method
Tool support:	-
Period of activity:	since 2001[17]
Publications:	[Wa05], [St05], [BS06], [SBB07a], [SBB08], [SBB07b], [SB08], [Lu09], [SB09], [Br10] [St10a]
Inner organization:	monolith

Starting with their research in 2001, a group of researchers at Sogeti Netherlands the approach of *DYnamic Architecture for modelling and development (DYA)*, which should help architects answering the question "When should I design which part of the architecture, with whom should I consult in doing this, and what will happen with the results?" [Wa05, page 3]. According to Wagter et al in [Wa05, pages 14–23], today's organizations face the challenges on the one hand to foster coherence, i.e. the interaction of various processes and the presentation of the organization as a uniform entity, and on the other hand to address the increasing demand for agility of the market. The objective of the DYA model is to support organizations in finding the right balance between agility and coherence. To achieve this aim, DYA offers means and methods to create agility without returning to ad hoc solutions and enables bridging the gap between strategy and realization. Thus, DYA proposes a continuous, cyclical, just-enough, just-in-time process that allows permissible deviations from architecture if necessary (cf. Wagter et al. in [Wa05, pages 204–206]). Put in other words, according to the DYA method no architectural descriptions are compiled in advance, instead only those domain architectures that are needed at a time are developed by a function, which is fully embedded in the organization's change process.

The approach presented by Wagter et al. centers around the notion of *dynamic* architecture (cf. [Wa05, pages 37–49]). Central to the understanding of dynamic architecture, is the chronological aspect of architecture. DYA distinguishes between three definitions, namely the description of the current situation, a blueprint for a desired future situation, and a set of guidelines for carrying out changes. While abstaining from presenting a dedicated information model for EA management, DYA in line with its focus on the methodological aspect of EA management[18] classifies the matter of subject into three types of architecture. The *business architecture* sketches the elements and structures that enable an organization to effectively pursue its business objectives, i.e. the products and services offered, the producing processes, and the organizational structure supporting these processes. The *information architecture* representing the contour of information provided in an organization, i.e. the data required and the distributing applications. Complementingly, the *technical architecture* is concerned with the make-up of the technical infrastructure necessary to support the organization, i.e.

[18] See Wagter et al. in [Wa05, page 43].

the hardware platforms, the network components, and the software/middleware required for information sharing. The third and final aspect when talking about architecture in DYA is the level of abstraction. The different levels of abstractions representing different views on an architecture can be exemplified by the means of principles and standards. General principles are defined on a general, i.e. abstract level, but can also be mapped to more concrete policy guidelines and rules. Models connecting these guidelines and rules to concrete elements of the EA represent the most detailed level of abstraction. Figure 3.33 illustrates the DYA architectural framework.

	Business objectives							
	Business architecture			Information architecture		Technical archtiecture		
	Prod / service	Process	Orga-nization	Data	Appli-cation	Middle-ware	Plat-form	Net-work
General principles								
Policy lines								
Models								

Figure 3.33: The DYA architectural framework [BS05, page 4]

Complementing principles as cross-cutting aspects of an EA, the aspect of goal-orientation and traceability is discussed in [SB08] via the so-called *architecture effectiveness model* (AEM). Whereas DYA emphasizes the role of business objectives as the source of all architectural decisions, the question how to express or reflect these business objectives in the information model are not addressed [BS06, pages 22-23]. The AEM provides a general structure to express how EA management contributes to the business goals of an organization, namely *architectural results*, *organizational performance*, and *business goals* [SB08, pages 609–610].

In order to cope with the increasing complexity of organizations, Bruls et al. further propose the utilization of *domain architectures* (cf. Bruls et al. in [Br10]). The idea of domain architectures is also referred to by van den Berg und van Steenbergen in [BS06], who distinguish three levels of architectures *enterprise architecture*, *domain architecture*, and *project start architecture*. The latter representing the architecture for a specific project on the operational level, which serves as means to communicate target architectures to the project team [BS06, page 40].

Table 3.36 shows how the DYA approach can be classified in respect to the language framework devised in Section 2.2.2. It nevertheless has to be added that the language prescriptions abstain on a very abstract level due to the focus on the method aspects. Hence, no detailed descriptions of the information models can be found in the DYA approach.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.36: Language classification for the DYA approach

DYA consists of three distinct processes that support an organization in its evolution with the full benefits of using architecture. The processes of the DYA model affect the dynamic architecture and are supported by a respective governance structure as illustrated in Figure 3.34.

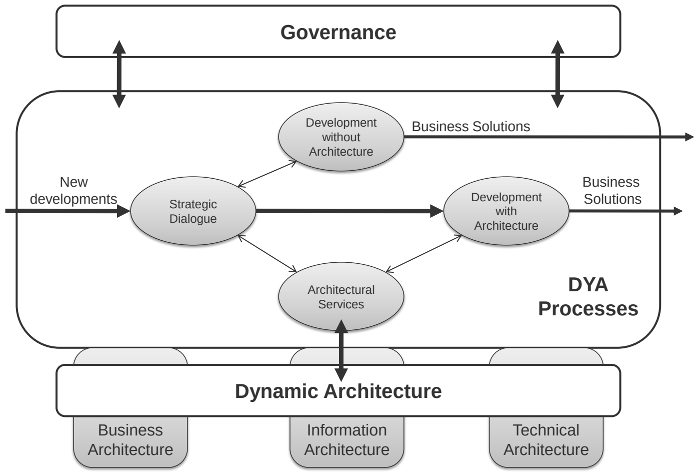

Figure 3.34: The DYA Model according to Wagter et al. [BS06, page 39]

- In the process of *strategic dialog*, the business objectives to be pursued are determined and after due consideration and exchange between business and IT management are further refined as project proposals. This process ensures that the right things are done at the right time.
- The process *architectural services* is triggered if a change on the strategic level is desired i.e. triggered by the strategic dialog. The process follows a cyclic nature and develops principles, guidelines, and models that enable realization of the business case. The architectural services ensure that things are done correctly.

- The processes *development with/without architecture* the concrete project proposals with desired time frame, level of quality and acceptable costs. Both processes represent alternatives at which *with* architecture represents the standard.
  - In the process *development with architecture* each project proposal is furnished with a project-start architecture, i.e. a description of concrete standards, norms, and guidelines to be followed by the project (anticipatory strategy).
  - If specially circumstances apply, e.g. extremely urgent situations, the process *development without architecture* can be chosen. Thereby certain aspects of architecture are temporarily ignored in a controlled and orderly fashion in the sense that consensus exists, how and when the temporary business solution is exchanged by a permanent architecture-conform solution and how the expenses therefore are covered (defensive or offensive strategy).

Using an fictitious telecommunication company, Wagter et al. describe the above processes in more detail and discuss the tasks to be performed, the roles that should be involved, and the artifacts, e.g. strategic document, impact analysis document, etc., that should be created. Furthermore, white papers detailing on single processes or process steps exists (cf. the Project Start Architecture by Luijpers in [Lu09]). Figure 3.35 illustrates the processes, involved actors, and resulting artifacts.

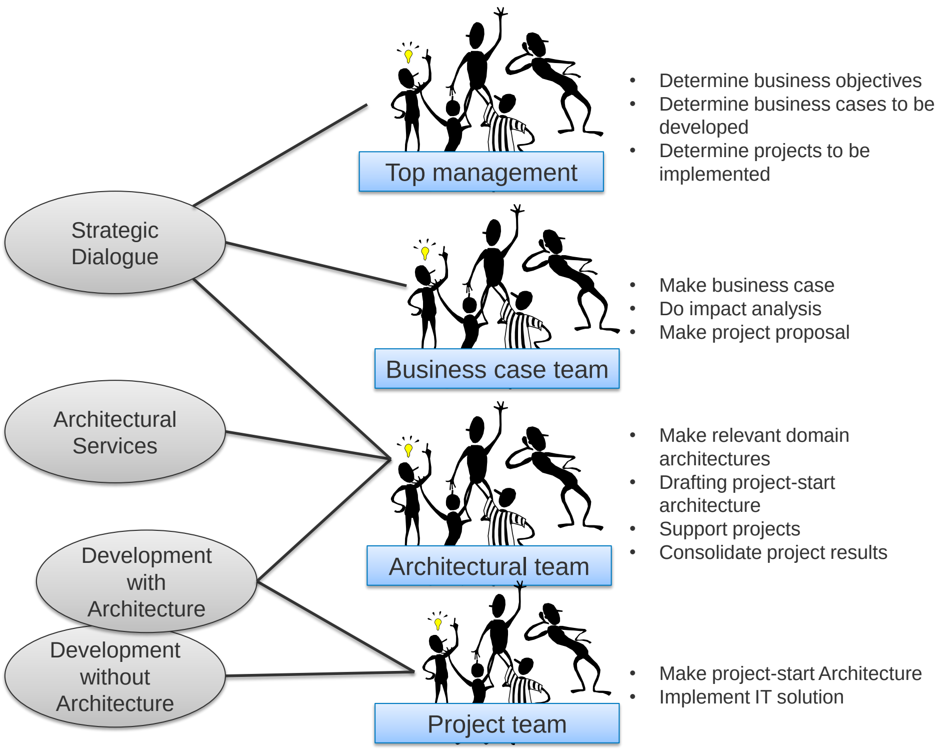

Figure 3.35: DYA processes according to Wagter et al. [Wa05, page 205]

Taking into account the system nature of enterprises and the EA management function respectively, Steenbergen (cf. [St05, St10a]) proposes a maturity matrix for DYA to guide enterprises in their evolution. The maturity model describes 18 focus areas, e.g. involvement of business, development process, or operation, commitment and motivation, or architecture tools [St05, page 2]. For each of these focus areas a number of maturity stages is distinguished (ranging from two maturity stages, i.e. A and B, to four, i.e. A to D). The position of the letters in the matrix from left to right indicates the order in which an organization should strive to reach the distinct stages. For each stage of each focus area a set of questions to be answered is provided. The question can be used on the one hand to determine the current maturity stage of an organization, and on the other hand provide guidance for future evolution and adaptation of the EA management function. Table 3.37 shows how the DYA approach can be classified in respect to the method framework devised in Section 2.2.1.

Integration	unidirectional	bidirectional			
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context	scope and reach			
Adapt to	organizational context	scope and reach			

Table 3.37: Method classification for the DYA approach

3.20 The EA management approach of Niemann

EA management approach	
Name of approach:	Niemann
Issuing organization:	act! consulting (consultant)
Focus area:	Method
Tool support:	-
Period of activity:	since 2006
Publications:	[Ni06]
Inner organization:	monolith

Klaus D. Niemann is the managing director of act! consulting, which is specialized in the development of EAs. Klaus D. Niemann has more than 20 years experience in the area of EA management(cf. [Ni06]), which he has written down in a book "From Enterprise Architecture to IT Governance" [Ni06].

The book presents the so-called *EA Cycle* consisting of the phases document, analyze, plan, act, and a central check activity (cf. Niemann in [Ni06, page 37]) as illustrated in Figure 3.36. According to Niemann in [Ni06, pages 170–177], EA management as to be integrated in existing management structures, i.e. the interaction with other processes and functions has to be defined. Relevant functions are program and service management and requirements and portfolio management, which may have a bidirectional connection in the sense that e.g. the portfolio management on the one hand receives decisions from the requirements management as input, whose decisions in turn provide input for the planning activity of EA management on the other hand.

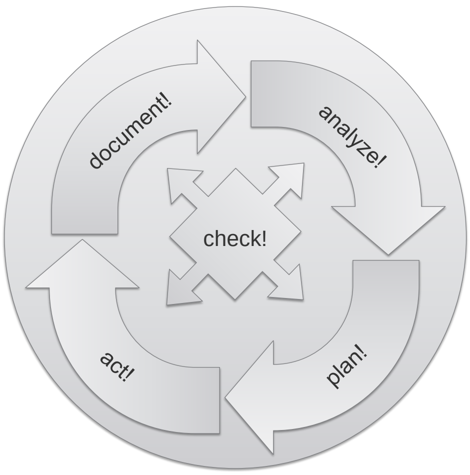

Figure 3.36: The EA Cycle of Niemann [Ni06]

The documentation method is concerned with defining the scope and reach of the EA management function, implementing, and populating the model [Ni06, page 41]. Thereby, typical pitfalls of EA management are sketched and solutions to avoid these shortcomings are proposed. Niemann proposes to structure an EA model according to three main levels called *business architecture*, *application architecture*, and *systems architecture* [Ni06, page 77]. For each of the layers a detailed description of the elements and relationships contained is given and cross-layer relationships are introduced. Furthermore, functional and non-functional requirements are introduced as cross-cutting aspects, which influence elements on all layers (cf. Figure 3.37).

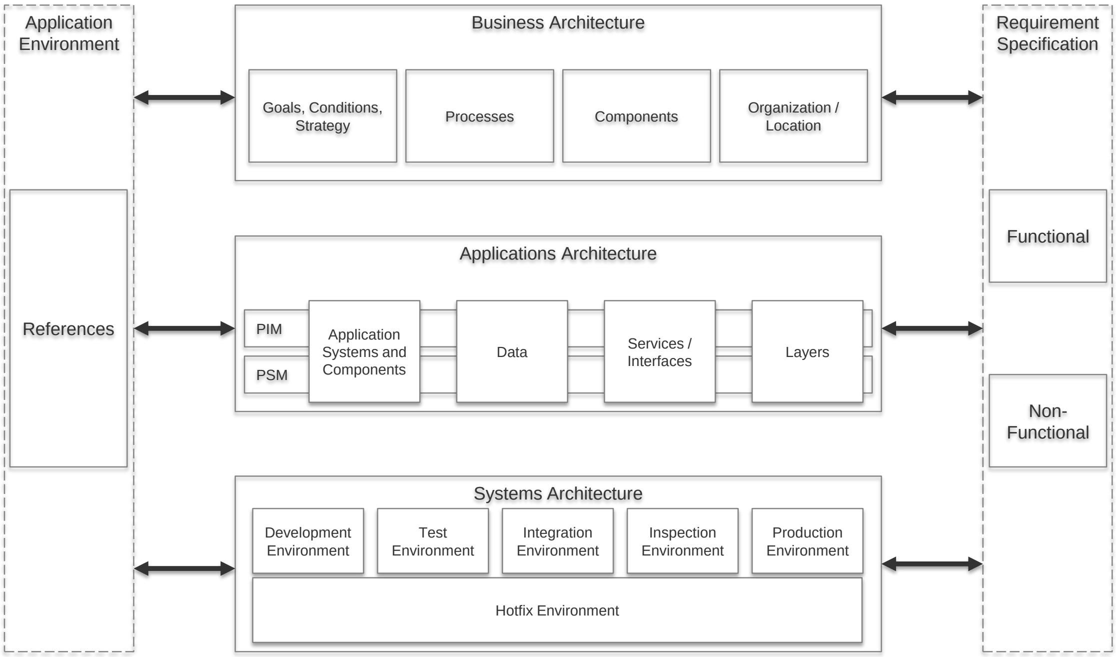

Figure 3.37: The information model of Niemann [Ni06]

During the document phase, Niemann emphasizes on the description of the current state of the EA, the development of planned future states is sketched as part of the plan activity. Thus, projects from the project portfolio management are considered in development planning [Ni06, page 161]. The coverage of projects is neither exemplified nor is the effect of projects on the EA referred to. In order to derive and establish principles, i.e. reference models in the terminology of Niemann [Ni06, page 97], guiding the future evolution of the EA, Niemann proposes to evaluate existing development lines and set up standards. Niemann lists different application scenarios for utilizing these reference architecture models but limits their reach to the IT-related elements (cf. [Ni06, pages 102–105]). While Niemann specifies different viewpoints applicable to populate architectural descriptions, no hints *how* this population can be performed, e.g. via the intranet or e-mail, are given. Similarly, the importance of providing "stakeholder-specific views" [Ni06, page 82] is explicitly referred to, while in contrast no dedicated audience for the proposed visualizations is given.

In order to analyze different states of the EA, Niemann proposes several questions for e.g. evaluating dependency, coverage, heterogeneity, or complexity (cf. [Ni06, pages 126–152], which are linked to all architectural layers. Thereby, qualitative as well as quantitative analysis techniques are provided. According to Niemann, these questions can be used to analyze the current state of the EA and identify potentials for improvement. For each of the proposed questions an analysis method is provided. In contrast, methods for evaluating planned states or for performing delta analyses are not provided by Niemann, although he provides criteria for such an analyses, like cost efficiency, ability to reduce risks, or impact on functional requirements [Ni06, page 163].

To support the establishment of an EA management function, Niemann details on different line organizations of an enterprise, which should be considered and require a different organization of the EA management function (cf. [Ni06, pages 178–181]). In the same sense, the author discusses the need of "steering" [Ni06, page 121] EA management by emphasizing on aspects like scope and reach and the need for adapting an EA management function. Nevertheless, concrete mechanisms to adapt the methods of the EA cycle or how to include an organization-specific terminology are not given.

Table 3.38 summarizes the analysis results of the EA management approach of Niemann in respect to methodological aspects.

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.38: Method classification for the approach of Niemann

Table 3.39 summarizes the analysis results of the EA management approach of Niemann in respect to linguistic aspects.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.39: Language classification for the approach of Niemann

3.21 The EA management approach of the University of St. Gallen

EA management approach	
Name of approach:	(Approach of the university of St. Gallen)
Issuing organization:	University of St. Gallen
Focus area:	Method
Tool support:	ADOben [Ai09b, Ai09a]
Period of activity:	since 2007 (2003)
Publications:	[ÖW03], [WF06], [AS07], [Br07], [KW07], [SS07], [Ös07], [Ai08b], [ARW08a], [Fi08], [HW08], [WS08], [Ai09b], [Ai09a], [AW09], [Ku09], [KW09], [RA09], [AG10]
Inner organization:	explicit organization

At the university of St. Gallen there is a long lasting engagement in the field of "business engineering" building on the groundworks of Österle and Winter in [ÖW03]. Understanding business engineering as holistic approach for designing organizations in the information-age, hence closely related to EA management, one can claim that the university of St. Gallen is into the topic of EA management since 2003. Nevertheless, at least since 2007 more and more of the work becomes specifically devoted to EA management topics, with Winter and Fischer [WF06] introducing the layered framework for the EA as shown in Figure 3.38.

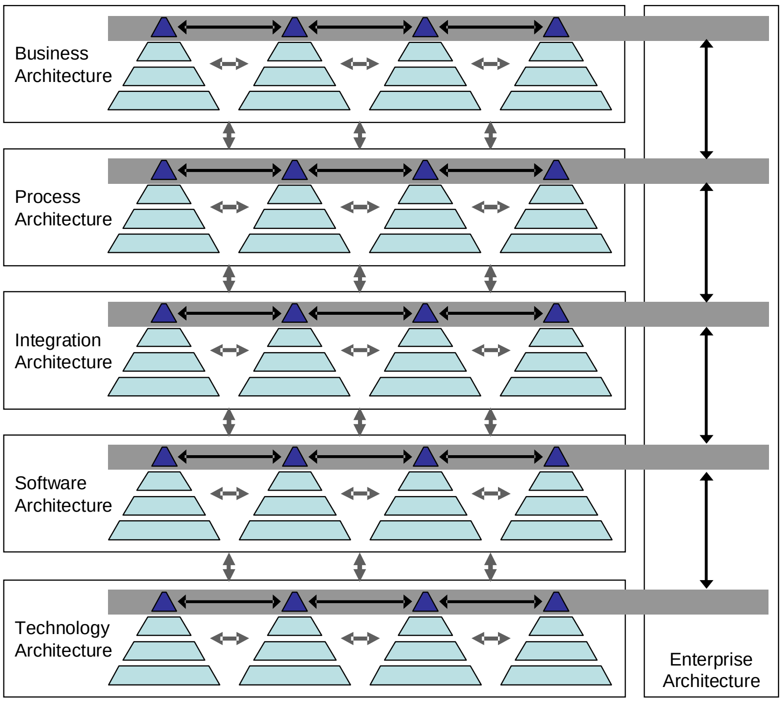

Figure 3.38: Essential layers of an EA [WF06]

In the understanding of Winter and Fischer EA seeks to provide a "cross-layer view of aggregate artifacts" in order to address challenges that are not confined to a single layer. In particular, three main aims of EA management are denoted: a) support business/IT alignment, b) support business development, especially business re-engineering as well as IS re-engineering, and c) support maintenance. Aier et al. put in [ARW08a] this topic under inverstigation again, applying statistical techniques on the outcomes of a survey. In this survey practitioners were asked to rate 15 properties of an EA according to the level of implementation experienced in the specific organization. Clustering the survey's results, Aier et al. discover what they call three different "EA scenarios", reflecting typical stages that an organization managing its EA may be in.

In [RA09] Riege and Aier further the above idea towards a "contingency framework" for EA management, i.e. an organized set of factors that may influence the actual make-up of an EA management function in an organization. Relating the factors back to the aims that the corresponding organization seeks to pursue with EA management, Riege and Aier are able to predict which of the subsequent aims is – based on the contingency factors – most important for an organization:

- support of business strategy developement,
- support of business operations, or
- support of IT management.

In [SS07] Schelp and Stutz approach the EA from a strategic perspective, showing how to apply the balanced scorecard mechanism onto the topic. They devise the "EA scorecard framework" relating the different perspectives of the scorecard to different EA layers. Further, they delineate the method for applying scorecards consisting of four stereotypic steps (cf. [SS07, page 9]) as 1) "develop strategy and metrics on business level", 2) "define business goals", 3) "monitor metrics with the framework", and 4) "adjust strategy, goals and metrics". While not much detail on the steps is added, the four step PDCA-like (cf. [De82a, Sh86]) method is a recurring principle throughout the EA management approach. A technique for analyzing EAs is outlined by Aier and Schelp in [AS07]. Understanding the architecture as untyped graph, they apply clustering algorithms for determining (candidates for) structuring principles, so called "domains". This idea is later furthered by Aier and Winter in [AW09] to create proposals on how to organize the decoupling of business and IT. In [WS08, pages 549–550] Winter and Schelp reflect on the different types of analyses that may be performed on an EA organized in layers as shown in Figure 3.38. Basically distinguishing between intra-layer, inter-layer and extra-layer analyses, they expatiate on seven different kinds of analyses ranging from simple "dependency analyses" over "complexity analyses" to more economically motivated "cost" or "benefit analyses". These analyses may be understood as techniques embedded into the larger whole of a consistent EA management method or procedure model. Hafner and Winter reflect on the requirements and the general make-up of such process model in [HW08], demanding at foremost that the model is both "scalable" with respect to the covered part of the organization and "evolutionary" accomodating a changing level of process maturity. Further, they require a process model to be "organizationally compatible" meaning that each organization has its specific culture, stakeholder setting and involved actors, which the process model has to adapt to. Deriving from three case studies at Credit Suisse, Die Mobiliar and HypoVereinsBank as well as from theoretic underpinnings in literature, Hafner and Winter delineate the four core activities of the process model with related sub-activities as show in Figure 3.39.

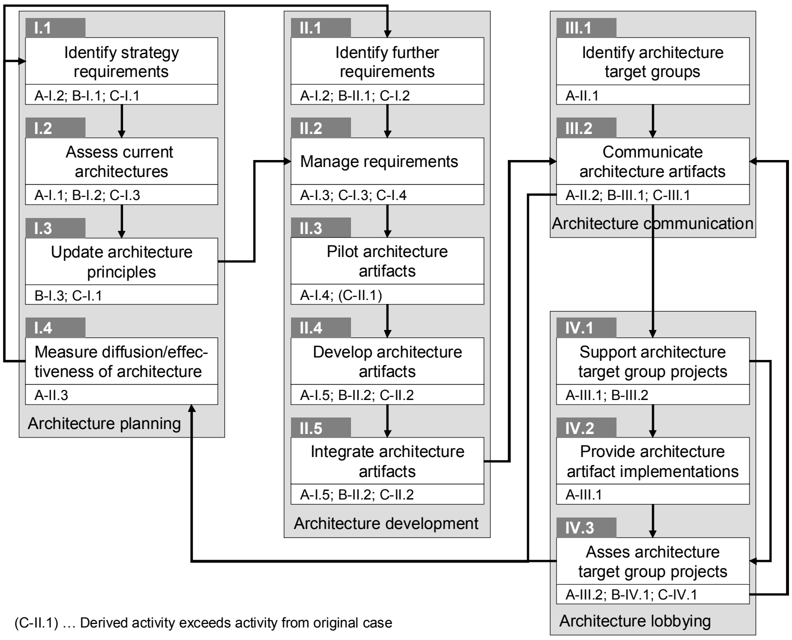

Figure 3.39: EA process model of Hafner and Winter [HW08]

The four core activities may be identified with typical activities in knowledge intense disciplines with the possible exception of the "architecture lobbying", which is quite specific for EA management. This may be ascribed to the EA management function often not being empowered to actually make prescriptions for the organization. The core activities of Hafner and Winter are further mirrored in the work [Fi08, pages 114–118] of Fischer, although the latter identifies four slightly different core activities[19] namely "strategic dialog", "architecture development", "architecture implementation", and "architecture maintenance". Following a well-defined method, Fischer details the activities over an intermediary M1-level to concrete processes described in an activity-diagram like syntax [Ob10c] on M0-level. Iterating over the the different processes, Fischer describes [Fi08, pages 145–184] the distinct tasks, their execution order as well as the assigned actors in detail. Furthering the multi-level understanding of the process model, Fischer further discusses the organizational structures and roles (cf. [Fi08, pages 185–205]) that are required to support the aforementioned M0-level processes. Beside this organizational embedding of the processes, no references to other contingency factors of EA management are provided.

Complementing the findings of Fischer, Aier et al. refrain in [Ai08b] their understanding of EA management as a design discipline mirroring characteristics of classical engineering disciplines, summarized in line with Shaw [Sh90] as "creating cost-effective solutions for relevant problems using scientific knowledge in service to society". From this, Aier et al. derive two relevant consequences, namely the question of "depth vs. width" and a set of general mechanisms used in EA management. While the former question is of relevance for the EA description language (see below), the general mechanisms described may be used as part of the M0-level processes of Fischer. In particular, Aier et al. describe model navigation mechanisms as well

[19] The process model consisting of these activities is called "M2" model.

as viewpoints that may be used to comprise specific information to a stakeholder. These mechanisms are revisited by Aier et al. in [Ai09b] and [Ai09a], where they show how the mechanisms can be implemented in a meta-modeling platform (ADOxx[20] of BOC), creating the "business engineering navigator" ADOben. The question of organization-specificity of EA management that reverberates throughout the entire approach of the University of St. Gallen is central subject of the discussions undertaken by Kurpjuweit and Winter in [KW09]. This article, furthering the considerations from [KW07], especially describes on how to configure the EA perspective to the stakeholder's architecture perception. In the recent work [AG10, page 60] of Aier and Gleichauf the topic of "EA planning" is discussed and the understanding of different states of the EA (current, planned and roadmap) is put on a sound methodical basis. In particular, dedicated activities for transformation design, namely a) delta analysis, b) identify projects, c) identify temporal interdependencies, and d) schedule projects, are described and linked to the underlying conceptualizations of the EA. In the light of the results and contributions described above the method related prescriptions of the EA management approach of the University of St. Gallen can be classified as shown in Table 3.40.

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.40: Method classification for the University of St. Gallen

Backbone of the EA description language employed in the approach of the University of St. Gallen is the "core business meta-model" described by Österle et al. in [Ös07]. This meta-model, shown in Figure 3.40, introduces the basic concepts that reside on the different layers of an EA.

The concepts contained in the core business meta-model are described on an abstract level, focusing on relationships and abstaining from giving details on the attributes of the concepts. Further, the Österle et al. call for an intuitive understanding of the concepts giving no explicit glossary of terms. In [Br07] Braun complements the core business meta-model[21] with more explicit sub-models for "strategy modeling", "organization modeling" and "IS modeling". The corresponding sub-models introduce the relevant model concepts together with a textual definition of their semantics and additionally provide a notation, i.e. define the symbols used to represent a specific concept. The different sub-models are further complemented with techniques and viewpoints that are to be applied as part of performing EA management. Braun puts special emphasis on the relationships between the sub-models [Br07, pages 171–178] and delineates how other modeling language, as the UML, may link to specific sub-models [Br07, pages 178–181]. Notwithstanding, the different sub-models ought to be used as a whole in order to achieve comprehensive business modeling. In a brief side-note, Braun discusses how

[20] For more information on the ADOxx meta-modeling platform, see e.g. `http://www.openmodels.at/c/document_library/get_file?p_l_id=65121&folderId=65129&name=DLFE-2505.pdf` (last accessed 09-07-2010).

[21] According to the terminology from Section 2 the meta-model can be identified with the information model.

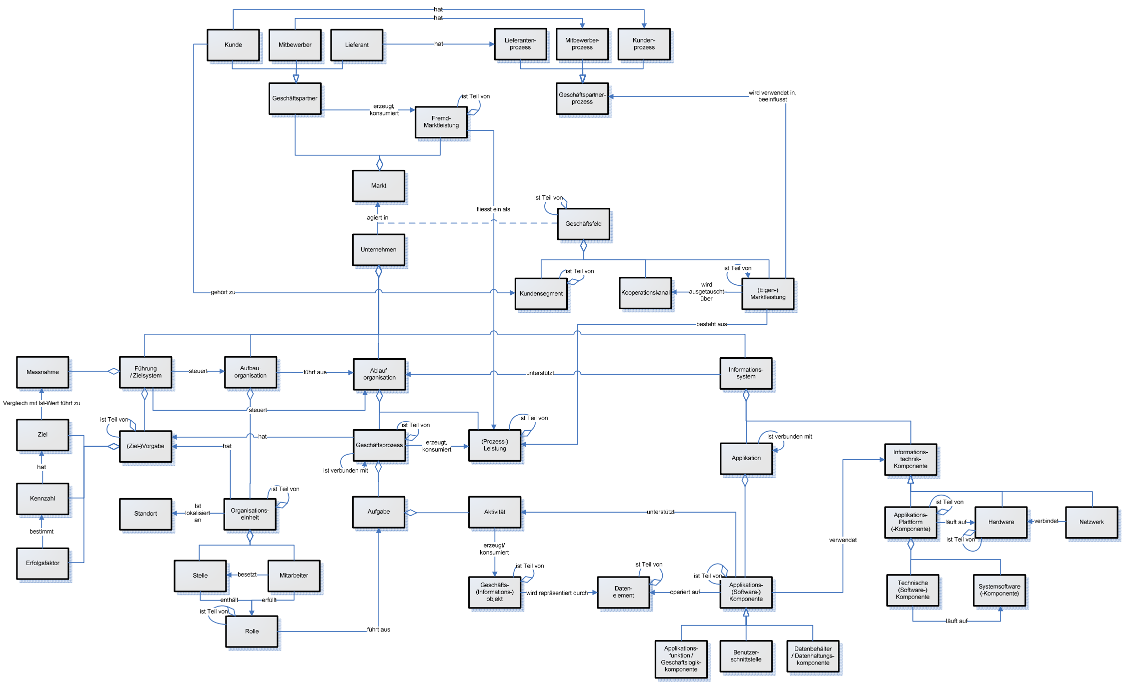

Figure 3.40: Core business meta-model of Österle et al. [Ös07]

service-oriented concepts may be incorporated into the meta-model, an idea taken up by Aier and Winter in [AW09]. There, they advocate for introducing an additional architectural layer, called "alignment architecture" to decouple "supply" and "demand" layers in an architecture. Exemplifying this along a decoupling in the IS support for business processes, Aier and Winter introduce the concept of the "enterprise service" on an intermediary level. These services are identified using the domain clustering technique as described by Aier and Schelp in [AS07].

Aside from aforementioned discussions on the additional abstractions in architecture modeling, Kurpjuweit and Winter discuss in [KW07, pages 6–10] the need for organization-specific meta-models. This is explained with the high maintenance effort connected to describing parts of the overall organization that are not needed for EA management. As the understanding of what is needed for EA management may differ from organization to organization, Kurpjuweit and Winter describe a "systematic approach to meta-model engineering" that enables a using organization to develop a specific meta-model. The approach consists of five steps as follows:

1. identify relevant concerns, i.e. areas-of-interest,
2. elicit stakeholder requirements and derive situated metrics,
3. select viewpoints and create viewpoint relationship overview,
4. select or design meta-model fragments, and
5. integrate meta-model fragments.

Whereas the general approach may be regarded as adaptation of classic domain modeling techniques, especially step 4 deserves special attention. The selection of meta-model fragments points towards a collection of such fragments that was at that time yet to be compiled. Two years later, Kurpjuweit made available such collection in [Ku09]. In the thesis he describes different partitions of the core business meta-model that may be regarded meta-model fragments in their own rights. Each fragment is further complemented with at least one viewpoint that may be applied to visualize the specific information contents conveyed in the fragment, more precisely in instantiations thereof. For particular examples of viewpoints in application cases Kurpjuweit [Ku09, pages 224–228] further details which additional information may be attached to a viewpoint in order to embed it into the situated context of an EA management method. Based on the common terminology of the core business meta-model, Kurpjuweit describes on techniques for integrating different meta-model fragments and delineates further details the system approach for meta-model engineering as introduced in [KW07]. Summarizing the characteristics of the language prescriptions in the EA management approach of the University of St. Gallen, we classify it as shown in Table 3.41.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.41: Language classification for the University of St. Gallen

3.22 Strategic IT management of Hanschke

EA management approach	
Name of approach:	Strategic IT management
Issuing organization:	iteratec GmbH (consultancy)
Focus area:	-
Tool support:	iteraplan
Period of activity:	since 2010
Publications:	[Ha10]
Inner organization:	explicit organization

In [Ha10] Hanschke presents what she calls "strategic IT management", a collection of best-practices further labeled as a "practical toolkit"[22] for EA management. In the preface of the work Hanschke calls upon the need to have such workable toolkit, as literature on approaches for strategic IT management is abundant, but "remote from real-life practice and do not permit ad-hoc use" [Ha10, page 65]. This reflects the gap that the work of Hanschke seeks to close by describing practice-proven prescriptions and guidelines for the context of EA management. It may be ascribed to this self-image of the approach that the majority of statements contained in [Ha10] are of pragmatic nature, not detailing the intricacies of doing research on a sound and coherent terminological basis. Moreover, the presented approach is complemented with an open source EA management tool, "iteraplan", in which the prescriptions and guidelines, especially concerning the language perspective, are implemented. Central to the approach is a framework deconstructing the EA into a set of interrelated sub-architectures as shown in Figure 3.41. For each of these architectures, the strategic IT management toolkit provides best-practices and prescriptions, although the focus of [Ha10] lays on the application architecture and the corresponding management activity of "IT landscape management".

From a methodical perspective, Hanschke's strategic IT management presents methods for three management subjects in an EA, namely for the "business landscape", the "IT landscape" and the "technology landscape". Strong emphasis is thereby laid on the IT landscape, whereas for the "infrastructure landscape" (cf. Figure 3.41) no dedicated methodical prescriptions are made. Regarding the management of the IT landscape, Hanschke describes four distinct main-processes: "documenting", "analyzing", "planning", and "governing". For each of these processes, the IT management toolkit further details activities to be performed, constraints to be accounted for, and actors to be involved in. This may well be exemplified with IT landscape documentation, for which Hanschke recommends to devise a "maintenance concept". The concept not only describes how is responsible for keeping which information up-to-date but further supplies information on related enterprise-level management processes, e.g. project portfolio management, that may serve as sources of according information. In respect to analyzing IT landscapes the approach (cf. [Ha10, pages 140–142]) describes coarse-grained categories for typical questions and gives a template that may be used to describe concrete, i.e. organization-specific, "analysis patterns". Exemplary applications of this template illustrate how concrete steps for performing analyses may look alike, further detailing so called "gap analyses" that are useful in comparing current and to-be (plan) states of the IT landscape.

[22] Publisher's description of the book available at `http://www.iteratec.com/download/StrategicITManagement.pdf`, last accessed 11-16-2010.

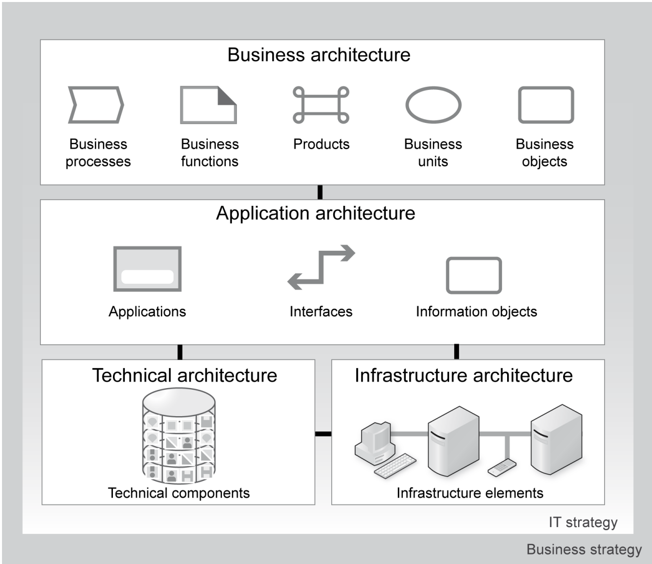

Figure 3.41: EA framework "Best-practice enterprise architecture" of Hanschke [Ha10, page 66]

IT landscape planning is according to Hanschke a circular and ongoing process (cf. Figure 3.42), in whose steps design/documentation and analysis of dedicated landscape states is performed iteratively. Detailing this process, the approach describes how intermediary planned landscapes can be derived from the to-be landscape, which in turn is based on the current landscape and on additional information on business requirements and strategic objectives. Thereby, so called "planning patterns" are applied which reflect proven-practice method fragments helpful in the context. Complementing the triple of documenting, analyzing, and planning, the topic of communicating EA plans in a stakeholder- and organization-specific manner reverberates through the work [Ha10] of Hanschke, further concretized in multiple exemplary visualizations, while dedicated communication processes are provided especially for the current landscape along with mechanisms to integrate IT landscape management with other enterprise-level management processes (cf. [Ha10, pages 190–193]). The topic of communication is picked up with respect to standards as part of "technology landscape management". In detail the approach describes steps for developing, maintaining, communicating, and enacting technological standards reflecting EA principles. Complementing the aforementioned activities the strategic IT management approach of Hanschke discusses the EA management governance, i.e. the structures, roles and responsibilities necessary to successfully implement the "toolkit" in a using organization. Different "maturity levels" of IT landscape management (cf. [Ha10, pages 194–206]) describe a possible roadmap for adapting scope and reach of the management function, but also discuss how new stakeholders may be involved into the management processes, making them "beneficiaries" of the artifacts, documentations, and visualizations.

Table 3.42 shows how the approach of Hanschke [Ha10] can be classified in respect to its methodical coverage against the framework devised in Section 2.2. It nevertheless has to be added that the methodical prescriptions are limited to methods for IT landscape management and technology landscape management, hence not targeting an embracing EA management approach.

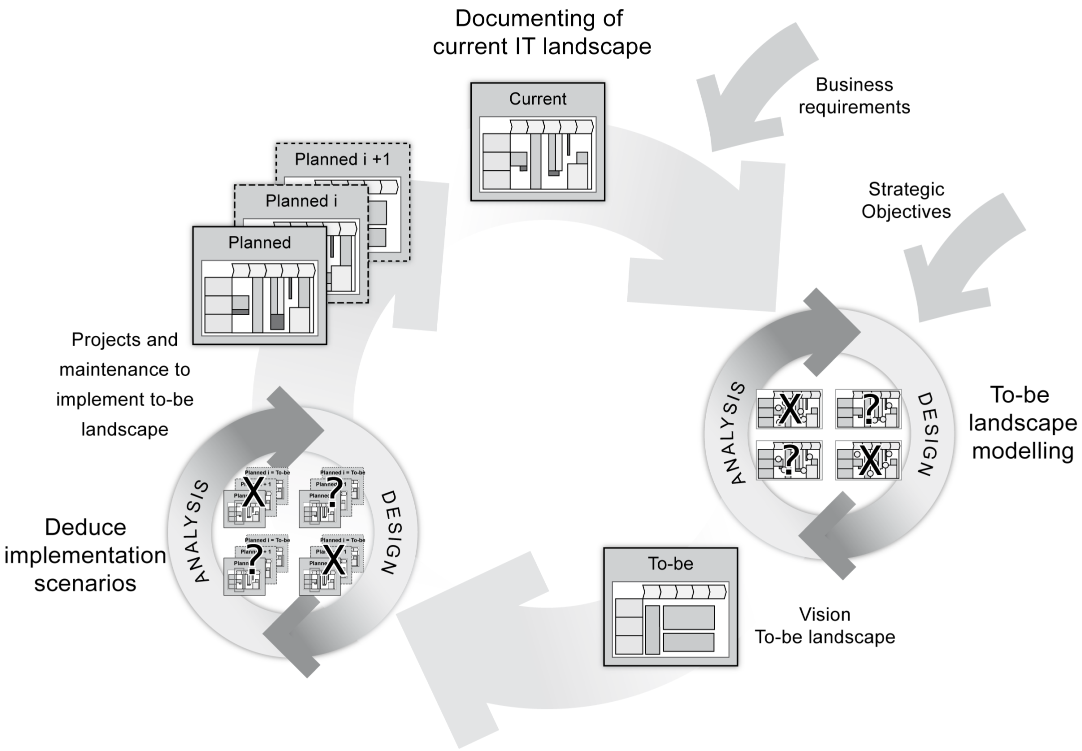

Figure 3.42: IT landscape planning process according to Hanschke [Ha10, page 158]

Integration	unidirectional		bidirectional		
Develop & describe	current	planned	target	principle	question
Communicate & enact	current	planned	target	principle	question
Analyze & evaluate	current	planned	target	delta analysis	
Configure to	organizational context		scope and reach		
Adapt to	organizational context		scope and reach		

Table 3.42: Method classification for Strategic IT management

With its emphasis on providing an executable toolkit, the strategic IT management approach of Hanschke [Ha10] adds profound information on language related aspects. More precisely, all four landscapes are to some degree detailed with information models that outline the relevant concepts on the business, the application, and the infrastructure side as well as concepts for standardizing applications and infrastructure elements. On the business level these concepts target both an external perspective, describing the organization's offerings to its customers e.g. "products", as well as an internal perspective of service delivery in business processes and via business functions. On the application and the infrastructure layer structural aspects are alluded to, describing the business applications and their interconnections as well as the linkages to hardware and network devices, always with a technical perspective. Special language mechanisms are applied on different concepts to account for a volatility in respect to the taken level-of-detail. Exemplifying this, the approach brings along a language primitive "hierarchization" that denotes that an instance of the corresponding concept may be repeatedly decomposed into sub-instances. The information model of the approach as shown in Figure 3.43[23] indicates hiearchization by the "H"-sign in the top-right corner of the corresponding concept.

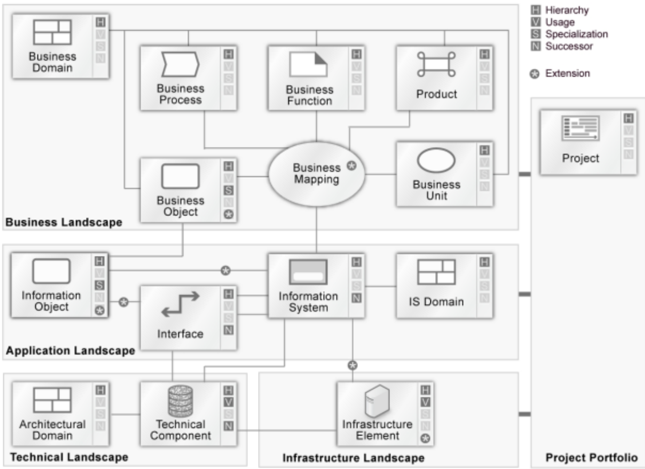

Figure 3.43: EA information model

Projects are essential elements in the information model and may be linked to EA concepts on business, application and technical level. In this sense, the language brings along mechanisms for describing how projects influence the corresponding landscapes on an abstract level, whereas the actual transformations performed by the projects can only be inferred on the application level using application lifecycle information. With the "technology landscape" being a central constituent of the approach's understanding of the EA, different concepts belonging to this landscape may be used to describe the technological standardization of the applications as well as their interfaces and interconnections. This allows to describe concretizations of architectural principles which are in turn devised as part of "technical standardization" (cf. [Ha10, pages 223–242]). In the context of standardization, again aspects of lifecycle modeling are alluded to, making it possible to specify, whether a technology "should be used in the future", is "in a phase out", or has already been "retired". The analysis mechanisms of the approach add further concepts for describing typical analysis questions in terms of so-called "analysis pattern" [Ha10, pages 151–157] that may be applied on any architectural level. These patterns do, as discussed in the method section, not only describe steps to be taken in analyzing landscapes, but also anchor the corresponding analyses in the affected architectural concepts. Therefore, textual descriptions of the analysis procedures are given and architectural attributes are referenced textually. These prescriptions could in the majority of cases be converted to algorithmic analysis techniques reified in EA-related questions, although the approach only sketches such procedure. Additionally, the approach describes how these analyses relate to the concept of the "IT goal", which is briefly alluded to by Hanschke [Ha10, pages 23–26]. At this point nevertheless no conceptual linkage between goals and the affected elements of the application landscape is established. In the context of IT landscape documentation methods, the approach of Hanschke [Ha10, pages 206–215] supplies what is called "guidelines for personalization" there. In these guidelines, the need to adapt the description language to

[23]See `https://www.iteraplan.de/wiki/display/iteraplan/iteratec+Best-Practice+Enterprise+Architecture`

the specific information demands of the stakeholders (called "beneficiaries" there) is alluded to, and concrete prescriptions on how to analyze the concern-stakeholder-relationships are provided. In this sense, mandatory and "nice-to-have" modeling concepts are distinguished and an understanding of the needed data quality is discussed. Against the background of the embracing information model provided by the approach, no dedicated mechanisms for evolving an existing model are considered. Reflecting these characteristics of Hanschke's strategic IT management [Ha10] as further implemented in the open-source EA management tool iteraplan against the background of the language analysis framework discussed in Section 2.2.2, we can classify the approach as depicted in Table 3.43.

Black-box perspective	business & organization	application & information	infrastructure & data
White-box perspective	business & organization	application & information	infrastructure & data
Strategies & projects	business & organization	application & information	infrastructure & data
Visions & goals	business & organization	application & information	infrastructure & data
Principles & standards	business & organization	application & information	infrastructure & data
Questions & KPIs	business & organization	application & information	infrastructure & data
Configure & adapt	initially	evolutionary	

Table 3.43: Language classification for Strategic IT management

CHAPTER 4

Conclusion and outlook

Publication titles like "state-of-the-art in ..." may be regarded as indications towards an ongoing consolidation process in a discipline. Such process can in turn be diagnosed for the field of EA management as the analyses in Section 3 showed. Revisiting the prominent approaches against the background of the general frameworks from Section 2.2, we could further show that as of today a comprehensive approach addressing all aspects of EA management is still missing.

In the following, we start with summarizing the method-related guidance:

Develop and describe is well covered by a majority of approaches, especially when it comes to prescriptions on how to document the current state and how to create a vision for a target state of the EA as well as a transformation roadmap. Prescriptions on how to develop architecture principles and how to develop as well as describe architecture questions are on the contrary scarce. The former gap aligns well with the apparent lack of descriptive techniques for architecture principles as obvious in the language-related evaluations.

Communicate and enact is also well addressed by many approaches, nevertheless with a significant drop in the frequency of communication-related prescriptions, when it comes to architecture roadmaps, architecture principles, and architecture-related questions. Latter fact may nevertheless not be considered a surprise, as different literature on EA management [Ku09, La09, Ch09] reports on difficulties in communicating stakeholder concerns. This would in turn be a prerequisite of communicating the relevant questions.

Analyze and evaluate has a fairly good coverage, but the lowest of the three core processes. In particular, only a few approaches concern themselves with methods for analyzing target states and development roadmaps. A practitioner's interest especially in the latter topic is nevertheless documented in the "synchronization management" scenarios of [se05] and [Ma08], such that we diagnose a critical lack in this field.

Integration is, despite EA management being a topic heavily relying on coordinating and informing existing management functions, not addressed by about one third of the approaches. Regarding the other approaches both integration scenarios are nearly equally alluded to. This points to the fact that many EA management approaches seek to install EA management as a super- or sub-ordinate management function instead of establishing a dense web of bidirectional linkages.

Configure and Adapt is not well covered by the approaches. In particular questions on how to tailor and re-tailor the methods in a changing organization is not addressed in depth, although some approaches discuss on "contingency factors" for an EA management function. In contrary, many approaches do nevertheless mention the need for organization-specific tailoring of EA management functions.

In the light of above summary (see also Table 4.1), one may say that concerning methods for EA management much is 'already in place', although mechanisms to find, select, and adapt the prescriptions to the using organization are really needed. This in particular advocates for an integrating and situational approach that allows bringing together the different prescriptions, thus making them available to an organization that is willing to establish a specific EA management function. The linguistic convergence in the field of EA management, as diagnosed by Schelp and Winter in [SW09], as well as the ongoing consolidation process can provide a good basis for research targeting the integration and combination of EA management approaches.

Revisiting the language-related prescriptions provided by the approaches, we again find an unequal coverage of the different language aspects. Along the dimensions of the classification framework (see Section 2.2), we summarize the language-related guidance as follows:

White box perspective is employed by a majority of approaches, almost equally targeting business, application, and infrastructure aspects. With the methods' barycenter on documentation, this strong coverage of the 'raw' make-up of an EA is not surprising. The equal coverage on the three layers can further be interpreted as objection against the often raised argument of EA management being IT-centric. Even if not quite a few of the analyzed approaches root in IS development, the structure of the supported business is well accounted for.

Black box perspective on business and application aspects is well covered by many approaches, which may be ascribed to the importance of service-orientation on both application and process level. When it conversely comes to a black box perspective on the infrastructure, the situation presents itself rather different, although the recent virtualization 'rally' may call for a more strict distinction between infrastructure components and their services in the near future.

Strategies and projects are covered by less than half of the approaches. On the one hand this reflects an understanding of EA management having project character, as promoted by some of the approaches. On the other hand the omission of projects reflects a critical lack in some of the approaches, that only account for the visionary nature of target architecture states but do not consider their implementation in detail.

Visions and goals are made explicit by slightly less than half of the approaches, at least when it comes to business or application level goals. This is particularly surprising, as there is a strong agreement on having to develop and to communicate target states for the architecture, whereas means for rationalizing the target state are scarce. This may ascribe to the fact that documenting visions is regarded an 'art' employing natural language 'models'.

Principles and standards are only covered by a few approaches with special focus on standards on both application and infrastructure layer. What may at first sight mirror the IS-centricity of the EA management approaches, moreover their rooting in IS development, can also be ascribed to the fact that as of today no concise understanding of the role of principles in EA management has developed (cf. [St09]). Especially, the regulatory nature of principles constraing the possible EA design space is only alluded to by quite a few approaches.

Questions and metrics are addressed by half of the approaches, nearly equally concerning metrics on application and infrastructure level with business level metrics being not only slightly behind. This reflects a situation as discussed by Frank et al. in [Fr08] stating that the architecture-related metrics are decoupled from business relevant ones, e.g. originating from IT controlling.

Configure and adapt is only covered by about half of the approaches. This is especially surprising as the need for an organization-specific information model has repeatedly been discussed in literature, e.g. in [Bu07b, Ai08a, KW09]. Having an inappropriate information model may be detrimental for an EA management approach in manifold ways. In particular the need to collect not relevant information may stall an EA management endeavor before it really starts.

Above summary (see also Table 4.2) sheds a light on the situation of EA description languages showing that EA management is more than one step from having a 'lingua franca' that embracingly covers the management subject. In particular, planning related concepts as projects, principles or goals are not well accounted for by the majority of the current approaches. This in turn calls for research heading towards an integration of different description languages especially providing mechanisms to only incorporate those concepts that are really needed by the using organization.

Acknowledgments

For anyone, who has been researching a discipline as embracing as EA management for some years, it may be clear that developing a comprehensive image of the state-of-the-art is no easy to accomplish task. Moreover, it is no task that two research assistants may have done on their own. For this, it is a pleasure for us to express our gratitude to the students and student assistants that helped us to do, what would have been impossible without their help. A warm, heartfelt and 'alphabetically ordered' thank you goes to Thomas Dierl, Mariana Mykhashchuk, and Michael Schätzlein.

Admitting that writing a document as lengthy as this report inevitably leads to some routine-blindness, we further would like to express our gratitude to the colleagues that helped reviewing the document. Ivan Monahov, Sascha Roth and Christopher Schulz did a great job in improving the overall quality of the report on both a linguistic and a scientific level. Thank you very much.

Integration	unidirectional (3.2, 3.3, 3.6, 3.7, 3.16)		bidirectional (3.9, 3.10, 3.11, 3.13, 3.15, 3.18, 3.19, 3.20, 3.21, 3.22)		
Develop & describe	current (3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.18, 3.20, 3.21, 3.22)	planned (3.8, 3.9, 3.11, 3.12, 3.13, 3.14, 3.15, 3.18, 3.20, 3.21, 3.22)	target (3.2, 3.3, 3.4, 3.6, 3.8, 3.9, 3.10, 3.11, 3.13, 3.15, 3.16, 3.17, 3.18, 3.19, 3.22)	principle (3.8, 3.11, 3.13, 3.19, 3.20, 3.22)	question (3.7, 3.11, 3.13, 3.14, 3.20, 3.22)
Communicate & enact	current (3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.9, 3.11, 3.12, 3.13, 3.14, 3.15, 3.16, 3.17, 3.20, 3.21, 3.22)	planned (3.9, 3.10, 3.11, 3.12, 3.15, 3.21, 3.22)	target (3.2, 3.3, 3.4, 3.6, 3.9, 3.10, 3.11, 3.13, 3.15, 3.16, 3.17)	principle (3.19, 3.22)	question
Analyze & evaluate	current (3.5, 3.7, 3.11, 3.13, 3.14, 3.15, 3.17, 3.20, 3.21, 3.22)	planned (3.12, 3.14, 3.15, 3.21)	target (3.11, 3.13, 3.15, 3.16, 3.22)	delta analysis (3.4, 3.8, 3.12, 3.15, 3.21, 3.22)	
Configure to	organizational context (3.8, 3.10, 3.15, 3.18, 3.20, 3.22)		scope and reach (3.4, 3.6, 3.13, 3.14, 3.15, 3.18, 3.19, 3.21)		
Adapt to	organizational context (3.9, 3.10, 3.22)		scope and reach (3.9, 3.13, 3.14, 3.19, 3.21, 3.22)		

Table 4.1: Summary of method classifications

Black-box perspective	business & organization (3.2, 3.3, 3.11, 3.6, 3.7, 3.8, 3.12, 3.13, 3.17, 3.22)	application & information (3.3, 3.8, 3.11, 3.12, 3.13, 3.14, 3.15, 3.21)	infrastructure & data (3.3, 3.13, 3.15)
White-box perspective	business & organization (3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13, 3.14, 3.15, 3.17, 3.18, 3.19, 3.20, 3.21, 3.22)	application & information (3.1, 3.2, 3.3, 3.4, 3.5, 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13, 3.14, 3.15, 3.18, 3.19, 3.20, 3.21, 3.22)	infrastructure & data (3.1, 3.3, 3.4, 3.5, 3.7, 3.8, 3.9, 3.10, 3.11, 3.13, 3.14, 3.15, 3.18, 3.19, 3.20, 3.21, 3.22)
Strategies & projects	business & organization (3.5, 3.8, 3.11, 3.12, 3.13, 3.15, 3.21, 3.22)	application & information (3.5, 3.8, 3.11, 3.12, 3.13, 3.15, 3.21, 3.22)	infrastructure & data (3.5, 3.8, 3.11, 3.13, 3.15, 3.22)
Visions & goals	business & organization (3.2, 3.5, 3.6, 3.7, 3.9, 3.11, 3.13, 3.20, 3.21)	application & information (3.5, 3.8, 3.9, 3.11, 3.13, 3.20, 3.21, 3.22)	infrastructure & data (3.5, 3.9, 3.13, 3.20)
Principles & standards	business & organization (3.8, 3.9, 3.13, 3.19)	application & information (3.8, 3.9, 3.13, 3.15, 3.19, 3.20, 3.22)	infrastructure & data (3.8, 3.9, 3.13, 3.15, 3.18, 3.19, 3.20)
Questions & metrics	business & organization (3.7, 3.9, 3.13, 3.14, 3.22)	application & information (3.7, 3.8, 3.9, 3.11, 3.13, 3.14, 3.15, 3.20, 3.22)	infrastructure & data (3.7, 3.9, 3.11, 3.13, 3.14, 3.15, 3.20, 3.22)
Configure & adapt	initially (3.3, 3.4, 3.6, 3.8, 3.14, 3.18, 3.21, 3.22)	evolutionary (3.6, 3.9, 3.21)	

Table 4.2: Summary of language classifications

Bibliography

[AG10] Aier, S.; Gleichauf, B.: *Application of Enterprise Models for Engineering Enterprise Transformation. Enterprise Modelling and Information System Architectures.* 5:56–72. 2010.

[Ai08a] Aier, S.; Kurpjuweit, S.; Riege, C.; Saat, J.: *Stakeholderorientierte Dokumentation und Analyse der Unternehmensarchitektur.* In (Hegering, H.-G.; Lehmann, A.; Ohlbach, H. J.; Scheideler, C., Ed.): *GI Jahrestagung (2).* volume 134 of *LNI.* pages 559–565. Bonn, Germany. 2008. Gesellschaft für Informatik.

[Ai08b] Aier, S.; Kurpjuweit, S.; Schmitz, O.; Schulz, J.; Thomas, A.; Winter, R.: *An Engineering Approach to Enterprise Architecture Design and its Application at a Financial Service Provider.* In (Loos, P.; Nüttgens, M.; Turowski, K.; Werth, D., Ed.): *Modellierung betrieblicher Informationssysteme (MobIS 2008).* pages 115–130. 2008.

[Ai09a] Aier, S.; Kurpjuweit, S.; Saat, J.; Winter, R.: *Business Engineering Navigator – A "Business to IT" Approach to Enterprise Architecture Management.* In (Bernard, S.; Doucet, G.; Gøtze, J.; Saha, P., Ed.): *Coherency Management – Architecting the Enterprise for Alignment, Agility, and Assurance.* pages 77–98. Bloomington. 2009. Author House.

[Ai09b] Aier, S.; Kurpjuweit, S.; Saat, J.; Winter, R.: *Enterprise Architecture Design as an Engineering Discipline. AIS Transactions on Enterprise Systems.* 1:36–43. 2009.

[AMT08] Aveiro, D.; Mendes, J. a.; Tribolet, J.: *Organizational modeling with a semantic wiki.* In *Proceedings of the 2008 ACM symposium on Applied computing.* SAC '08. pages 592–593. New York, NY, USA. 2008. ACM.

[Ar07] Arbab, F.; Boer, F. S. d.; Bonsangue, M. M.; Lankhorst, M. M.; Proper, E. H.; Torre, L. W. N. v. d.: *Integrating Architectural Models - Symbolic, Semantic and Subjective Models in Enterprise Architecture. Enterprise Modelling and Information Systems Architectures.* 2(1):40–57. 2007.

[ARW08a] Aier, S.; Riege, C.; Winter, R.: *Classification of Enterprise Architecture Scenarios – An Exploratory Analysis. Enterprise Modelling and Information Systems Architectures.* 3:14–23. 2008.

[ARW08b] Aier, S.; Riege, C.; Winter, R.: *Unternehmensarchitektur – Literaturüberblick Stand der Praxis. Wirtschaftsinformatik.* 50(4):292–304. 2008.

[AS07] Aier, S.; Schönherr, M.: *Integrating an Enterprise Architecture Using Domain Clustering.* In (Lankhorst, M. M.; Johnson, P., Ed.): *Second Workshop on Trends in Enterprise Architecture Research.* pages 23–30. 2007.

[AST10a] Aveiro, D.; Silva, A. R.; Tribolet, J. M.: *Extending the Design and Engineering Methodology for Organizations with the Generation Operationalization and Discontinuation Organization.* In *DESRIST 2010.* pages 226–241. 2010.

[AST10b] Aveiro, D.; Silva, A. R.; Tribolet, J. M.: *Towards a GOD-theory for organizational engineering: continuously modeling the continuous (re)generation, operation and deletion of the enterprise.* In *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing.* pages 150–157. New York, NY, USA. 2010. ACM.

[Av10] Aveiro, D.: *G.O.D. (Generation, Operationalization & Discontinuation) and Control (sub)organizations: a DEMO-based approache for continuous real-time management of organizational change caused by exceptions.* PhD thesis. Instituto Superior Técnico, Universidade Técnica de Lisboa. Lisbon, PT. 2010.

[AW09] Aier, S.; Winter, R.: *Virtual Decoupling for IT/Business Alignment – Conceptual Foundations, Architecture Design and Implementation Example. Business & Information Systems Engineering.* 1(2):150–163. 2009.

[BCR94] Basili, V. R.; Caldiera, G.; Rombach, H. D.: *The Goal Question Metric Approach.* Wiley. New York. 1994.

[Be95] Bem, D. J.: *Writing a Review Article for Psychological Bulletin. Psychological Bulletin.* 118(2):172–177. 1995.

[Be05] Bernard, S. A.: *An Introduction to Enterprise Architecture.* Authorhouse. Bloomington, USA. 2nd edition. 2005.

[BG09] Bernard, S. A.; Grasso, J.: *Enterprise Architecture Formalization and Auditing.* In (Doucet, G.; Gøtze, J.; Saha, P.; Bernard, S., Ed.): *Coherency Management – Architecting the Enterprise for Alignment, Agility, and Assurance.* Bloomington, USA. 2009. AuthorHouse.

[BMS09a] Buckl, S.; Matthes, F.; Schweda, C. M.: *Future Research Topics in Enterprise Architecture Management – A Knowledge Management Perspective.* In (Aier, S.; Schelp, J.; Schönherr, M., Ed.): *4th Workshop on Trends in Enterprise Architecture Research (TEAR 2009).* pages 5–20. 2009.

[BMS09b] Buckl, S.; Matthes, F.; Schweda, C. M.: *A Viable System Perspective on Enterprise Architecture Management.* In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 11-14 October 2009.* pages 1483–1488. IEEE. 2009.

[BMS10a] Buckl, S.; Matthes, F.; Schweda, C. M.: *Conceptual Models for Cross-cutting Aspects in Enterprise Architecture Modeling*. In *5th International Workshop on Vocabularies, Ontologies, and Rules for the Enterprise (VORTE 2010)*. 2010.

[BMS10b] Buckl, S.; Matthes, F.; Schweda, C. M.: *A Design Theory Nexus for Situational Enterprise Architecture Management – Approach and Application Example*. In *14th IEEE International EDOC Conference (EDOC 2010)*. Vitoria, Brazil. 2010.

[BMS10c] Buckl, S.; Matthes, F.; Schweda, C. M.: *From EA management patterns towards a prescriptive theory for designing enterprise-specific EA management functions – Outline of a research stream*. In (Schumann, M.; Kolbe, L. M.; Breitner, M. H.; Freirichs, A., Ed.): *Multikonferenz Wirtschaftsinformatik (MKWI2010)*. pages 67–78. Göttingen, Germany. 2010.

[BMS10d] Buckl, S.; Matthes, F.; Schweda, C. M.: *Interrelating Concerns in EA Documentation – Towards a Conceptual Framework of Relationships*. In (Engels, G.; Luckey, M.; Pretschner, A.; Reussner, R., Ed.): *2nd European Workshop on Patterns for Enterprise Architecture Management (PEAM2010)*. pages 243–252. Paderborn, Germany. 2010.

[BMS10e] Buckl, S.; Matthes, F.; Schweda, C. M.: *A Meta-Language for EA Information Modeling – State-of-the-art and Requirements Elicitation*. In (Bider, I.; Halpin, T.; Krogstie, J.; Nurcan, S.; Proper, E.; Schmidt, R.; Ukor, R., Ed.): *Enterprise, Business-Process and Information Systems Modeling*. Lecture Notes in Business Information Systems. pages 169–181. Springer. 2010.

[BMS10f] Buckl, S.; Matthes, F.; Schweda, C. M.: *A technique for Annotating EA Information Models*. In (Barjis, J., Ed.): *Enterprise and Organizational Modeling and Simulation: 6th International Workshop, EOMAS 2010 held at CAiSE 2010, Hammamet, Tunisia, June 2010 Selected Papers*. Lecture Notes in Business Information Systems. pages 113–127. Berlin, Heidelberg, Germany. 2010. Springer.

[BN94] Bernus, P.; Nemes, L.: *A Framework to Define Generic Enterprise Reference Architecture and Methodology*. In *Proceedings of the International Conference on Automation, Robotics and Computer Vision (ICARCV'94), Singapore, November 10–12*. pages 88–92. 1994.

[BN96] Bernus, P.; Nemes, L.: *A Framework to Define Generic Enterprise Reference Architecture and Methodology*. *Proceedings of the International Conference on Automation, Robotics and Computer Vision (ICARCV'94), Singapore, November 10–12*. 9:179–191. 1996.

[BNS03] Bernus, P.; Nemes, L.; Schmidt, G.: *Handbook on Enterprise Architecture*. Springer. Berlin, Heidelberg, Germany. 2003.

[Br07] Braun, C.: *Modellierung der Unternehmensarchitektur – Weiterentwicklung einer bestehenden Methode und deren Abbildung in einem Meta-Modellierungswerkzeug*. PhD thesis. Universität St.Gallen. 2007.

[Br10] Bruls, W. A. G.; Steenbergen, M. v.; Foorthuis, R. M.; Boos, R.; Brinkkemper, S.: *Domain Architectures as an Instrument to Refine Enterprise Architecture*. *Communications of the Associations for Information Systems*. 27:517–540. 2010.

[BS05] Berg, M. v. d.; Steenbergen, M. v.: *Dynamic Enterprise Architecture – Towards a new perception of architecture.* Technical report. Sogeti Nederland B.V. Diemen, The Netherlands. 2005.

[BS06] van Berg, M. d.; van Steenbergen, M.: *Building an Enterprise Architecture Practice – Tools, Tips, Best Practices, Ready-to-Use Insights.* Springer. Dordrecht, The Netherlands. 2006.

[Bu07a] Buckl, S.; Ernst, A. M.; Lankes, J.; Matthes, F.; Schweda, C. M.; Wittenburg, A.: *Generating Visualizations of Enterprise Architectures using Model Transformation (extended version). Enterprise Modelling and Information Systems Architectures – An International Journal.* 2(2):3–13. 2007.

[Bu07b] Buckl, S.; Ernst, A. M.; Lankes, J.; Schneider, K.; Schweda, C. M.: *A pattern based Approach for constructing Enterprise Architecture Management Information Models.* In (Oberweis, A.; Weinhardt, C.; Gimpel, H.; Koschmider, A.; Pankratius, V.; Schnizler, Ed.): *Wirtschaftsinformatik 2007.* pages 145–162. Karlsruhe, Germany. 2007. Universitätsverlag Karlsruhe.

[Bu08a] 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.

[Bu08b] Buckl, S.; Ernst, A. M.; Lankes, J.; Matthes, F.; Schweda, C. M.: *Enterprise Architecture Management Patterns – Exemplifying the Approach.* In *12th IEEE International EDOC Conference (EDOC 2008).* Munich, Germany. 2008. IEEE Computer Society.

[Bu08c] Buckl, S.; Ernst, A. M.; Matthes, F.; Schweda, C. M.: *An Information Model for Landscape Management – Discussing Temporality Aspects.* In (Johnson, P.; Schelp, J.; Aier, S., Ed.): *Pre-Proceedings of the 3rd Workshop on Trends in Enterprise Architecture Research.* pages 63–77. Sydney, Australia. 2008.

[Bu09a] Buckl, S.; Ernst, A. M.; Kopper, H.; Marliani, R.; Matthes, F.; Petschownik, P.; Schweda, C. M.: *EAM Pattern for Consolidations after Mergers.* In *SE 2009 – Workshopband.* pages 67–78. Kaiserslautern, Germany. 2009.

[Bu09b] Buckl, S.; Ernst, A. M.; Matthes, F.; Ramacher, R.; Schweda, C. M.: *Using Enterprise Architecture Management Patterns to complement TOGAF.* In *13th IEEE International EDOC Conference (EDOC 2009).* Auckland, New Zealand. 2009. IEEE Computer Society.

[Bu09c] Buckl, S.; Ernst, A. M.; Matthes, F.; Schweda, C. M.: *Enterprise Architecture Management Pattern for EA Visioning.* In *Proceedings of EuroPLoP 2009.* 2009.

[Bu09d] Buckl, S.; Ernst, A. M.; Matthes, F.; Schweda, C. M.: *How to make your enterprise architecture management endeavor fail!* In *Pattern Languages of Programs 2009 (PLoP 2009), Chicago.* 2009.

[Bu09e] Buckl, S.; Ernst, A. M.; Matthes, F.; Schweda, C. M.: *An Information Model for Managed Application Landscape Evolution. Journal of Enterprise Architecture (JEA).* 5(1):12–26. 2009.

[Bu09f] Buckl, S.; Franke, U.; Holschke, O.; Matthes, F.; Schweda, C. M.; Sommestad, T.; Ullberg, J.: *A Pattern-based Approach to Quantitative Enterprise Architecture Analysis.* In *15th Americas Conference on Information Systems (AMCIS).* San Francisco, CA, USA. 2009.

[Bu10a] Buckl, S.; Dierl, T.; Matthes, F.; Schweda, C. M.: *Building Blocks for Enterprise Architecture Management Solutions.* In (Harmsen, F. e. a., Ed.): *Practice-Driven Research on Enterprise Transformation, second working conference, PRET 2010, Delft.* pages 17–46. Berlin, Heidelberg, Germany. 2010. Springer.

[Bu10b] Buckl, S.; Matthes, F.; Roth, S.; Schulz, C.; Schweda, C. M.: *A Conceptual Framework for Enterprise Architecture Design.* In (Aalst, W.; Mylopoulos, J.; Sadeh, N. M.; Shaw, M. J.; Szyperski, C.; Proper, E.; Lankhorst, M. M. et al., Ed.): *Trends in Enterprise Architecture Research.* volume 70 of *Lecture Notes in Business Information Processing.* pages 44–56. Springer Berlin Heidelberg. 2010.

[BW06] Balabko, P.; Wegmann, A.: *Systemic classification of concern-based design methods in the context of enterprise architecture. Information Systems Frontiers.* 8(2):115–131. 2006.

[Ca07] Carlsson, S. A.: *Developing Knowledge Through IS Design Science Research: For Whom, What Type of Knowledge, and How? Scandinavian J. Inf. Systems.* 19(2):75 – 86. 2007.

[Ch09] Chair for Informatics 19 (sebis),Technische Universität München: *EAM Pattern Catalog Wiki.* `http://eampc-wiki.systemcartography.info` (cited 2010-02-25). 2009.

[CI04] CIMOSA Association: *CIM – Open System Architecture.* `http://www.pera.net/Methodologies/Cimosa/CIMOSA.html` (cited 2010-09-09). 2004.

[CST09] Caetano, A.; Silva, A. R.; Tribolet, J.: *A role-based enterprise architecture framework.* In *Proceedings of the 2009 ACM symposium on Applied Computing.* SAC '09. pages 253–258. New York, NY, USA. 2009. ACM.

[CST10] Caetano, A.; Silva, A. R.; Tribolet, J.: *Business Process Decomposition – An Approach Based on the Principle of Separation of Concerns. Enterprise Modeling and Information Systems Architectures.* 5(1):44–58. 2010.

[De82a] Deming, E. W.: *Out of the Crisis.* Massachusetts Institute of Technology. Cambridge, MA, USA. 1982.

[De82b] Deming, E. W.: *Quality, Productivity, and Competitive Position.* Massachusetts Institute of Technology. Cambridge, MA, USA. 1982.

[De09a] Department of Defense (DoD) USA: *DoD Architecture Framework Version 2.0: Volume 1: Introduction, Overview, and Concepts – Manager's Guide.* `http://www.defenselink.mil/cio-nii/docs/DoDAF%20V2%20-%20Volume%201.pdf` (cited 2010-02-25). 2009.

[De09b] Department of Defense (DoD) USA: *DoD Architecture Framework Version 2.0: Volume 2: Architectural Data and Models – Architect's Guide.* http://www.defenselink.mil/cio-nii/docs/DoDAF%20V2%20-%20Volume%202.pdf (cited 2010-02-25). 2009.

[De09c] Department of Defense (DoD) USA: *DoD Architecture Framework Version 2.0: Volume 3: DoDAF Meta-model Physical Exchange Specification – Developer's Guide.* http://www.defenselink.mil/cio-nii/docs/DoDAF%20V2%20-%20Volume%203.pdf (cited 2010-02-25). 2009.

[Di05] Dietz, J. L.: *A World Ontology Specification Language.* In *On the Move to Meaningful Internet Systems 2005: OTM Workshops.* pages 688–699. 2005.

[Di06] Dietz, J. L.: *Enterprise Ontology.* Springer. Heidelberg, Germany. 2006.

[Do08] Doucet, G.; Gøtze, J.; Saha, P.; Bernard, S. A.: *Coherency Management: Using Enterprise Architecture for Alignment, Agility, and Assurance. Journal on Enterprise Architecture.* 4(2). 2008.

[Do09a] Doucet, G.; Gøtze, J.; Saha, P.; Bernard, S. A.: *Coherency Management – Architecting the Enterprise for Alignment, Agility, and Assurance.* AuthorHouse. Bloomington, USA. 2009.

[Do09b] Doucet, G.; Gøtze, J.; Saha, P.; Bernard, S. A.: *Commencing the Journey: Realizing Coherency Management.* In (Doucet, G.; Gøtze, J.; Saha, P.; Bernard, S., Ed.): *Coherency Management – Architecting the Enterprise for Alignment, Agility, and Assurance.* Bloomington, USA. 2009. AuthorHouse.

[Do09c] Doucet, G.; Gøtze, J.; Saha, P.; Bernard, S. A.: *Introduction to Coherency Management: The Transformation of Enterprise Architecture.* In (Doucet, G.; Gøtze, J.; Saha, P.; Bernard, S., Ed.): *Coherency Management – Architecting the Enterprise for Alignment, Agility, and Assurance.* Bloomington, USA. 2009. AuthorHouse.

[ED09] Ettema, R.; Dietz, J. L.: *ArchiMate and DEMO – Mates to Date?* In *Advances in Enterprise Engineering III.* pages 172–186. Albani, Antonia and Barjis, Joseph and Dietz, Jan L.G. 2009.

[Ek04] Ekstedt, M.; Johnson, P.; Lindström, Å.; Gammelgård, M.; Johansson, E.; Plazaola, L.; Silva, E. et al.: *Consistent Enterprise Software System Architecture for the CIO – A Utility-Cost Based Approach.* In *37th Hawaii International Conference on System Sciences (HICSS 2004).* 2004.

[Ek09] Ekstedt, M.; Franke, U.; Johnson, P.; Lagerström, R.; Sommestad, T.; Ullberg, J.; Buschle, M.: *A Tool for Enterprise Architecture Analysis of Maintainability.* In (Winter, A.; Ferenc, R.; Knodel, J., Ed.): *13th European Conference on Software Maintenance and Reengineering, CSMR 2009.* pages 327–328. Kaiserslautern, Germany. 2009.

[Er10] Ernst, A. M.: *A Pattern-Based Approach to Enterprise Architecture Management.* PhD thesis. Technische Universität München. München, Germany. 2010.

[Fe94] Ferstl, O. K.; Sinz, E. J.; Amberg, M.; Hagemann, U.; Malischewski, C.: *Tool-Based Business Process Modeling Using the SOM Approach.* In *GI Jahrestagung.* pages 430–436. 1994.

[Fe06] Fettke, P.: *State-of-the-Art des State-of-the-Art – Eine Untersuchung der Forschungsmethode "Review" innerhalb der Wirtschaftsinformatik. WIRTSCHAFTSINFORMATIK.* 48(74):257–266. 2006.

[Fi08] Fischer, R.: *Organisation der Unternehmensarchitektur – Entwicklung der aufbau- und ablauforganisatorischen Strukturen unter besonderer Berücksichtigung des Gestaltungsziels Konsistenzerhaltung.* PhD thesis. Universität St.Gallen. 2008.

[FMW05] Fischer, F.; Matthes, F.; Wittenburg, A.: *Improving IT Management at the BMW Group by Integrating Existing IT Management Processes.* In *9th International Enterprise Computing Conference.* pages 19–23. 2005.

[Fr94] Frank, U.: *MEMO: A Tool Supported Methodology for Analyzing and (Re-)Designing Business Information Systems.* In (Ege, R.; Singh, M.; Meyer, B., Ed.): *Technology of Object-Oriented Languages and Systems.* pages 367–380. 1994.

[Fr98a] Frank, U.: *The MEMO Meta-Metamodel.* Technical Report 9. Arbeitsberichte des Instituts für Wirtschaftsinformatik Koblenz. 1998.

[Fr98b] Frank, U.: *The MEMO Object Modelling Language (MEMO-OML).* Technical Report 10. Arbeitsberichte des Instituts für Wirtschaftsinformatik Koblenz. 1998.

[Fr99] Frank, U.: *Memo: Visual Languages for Enterprise Modelling.* Technical Report 18. Arbeitsberichte des Instituts für Wirtschaftsinformatik Koblenz. 1999.

[Fr08] Frank, U.; Heise, D.; Kattenstroth, H.; Schauer, H.: *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.

[Fr09] Frank, U.: *The MEMO Meta Modelling Language (MML) and Language Architecture (ICB-Research Report).* Technical report. Institut für Informatik und Wirtschaftsinformatik. Duisburg-Essen, Germany. 2009.

[FS95] Ferstl, O. K.; Sinz, E. J.: *Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. Wirtschaftsinformatik.* 37(3):209–220. 1995.

[FS97] Ferstl, O. K.; Sinz, E. J.: *Modeling of Business Systems Using the Semantic Object Model (SOM) – A Methodological Framework.* pages 339–358. Springer. 1997.

[Ga75] Gadamer, H.-G.: *Wahrheit und Methode – Grundzüge einer philosophischen Hermeneutik.* J.C.B. Mohr. Tübingen, Germany. 3rd edition. 1975.

[Ge07] Getoor, L.; Friedman, N.; Koller, D.; Pfeffer, A.; Taskar, B.: *Probabilistic Relational Models.* In (Getoor, L.; Taskar, B., Ed.): *An Introduction to Statistical Relational Learning.* MIT Press. Cambridge, MA, USA. 2007.

[GLS06] Gammelgård, M.; Lindström, Å.; Simonsson, M.: *A reference model for IT management responsibilities.* In *10^{th} IEEE International EDOC Conference Workshops (EDOCW).* pages 26–33. Washington, DC, USA. 2006. IEEE Computer Society.

[Gr06] Gregor, S.: *The nature of theory in information systems.* *MIS Quarterly.* 30(3):491–506. 2006.

[Ha97] Harmsen, A. F.: *Situational Method Engineering.* PhD thesis. University of Twente. Twente, The Netherlands. 1997.

[Ha05] Halpin, T. A.: *ORM 2.* In *OTM Workshops.* pages 676–687. 2005.

[Ha10] Hanschke, I.: *Strategic IT Management – A Toolkit for Enterprise Architecture Management.* Springer. Berlin, Germany. 2010.

[He08] Heise, D.; Strecker, S.; Frank, U.; Jung, J.: *Erweiterung einer Unternehmensmodellierungsmethode zur Unterstützung des IT-Controllings.* In (Bichler, M.; Hess, T.; Krcmar, H.; Lechner, U.; Matthes, F.; Picot, A.; Speitkamp, B. et al., Ed.): *Multikonferenz Wirtschaftsinformatik.* GITO-Verlag, Berlin. 2008.

[HP04] Hirvonen, A. P.; Pulkkinen, M.: *A Practical Approach to EA Planning and Development: the EA Management Grid.* In (Abramowicz, W., Ed.): *Proceedings of 7th International Conference on Business Information Systems.* pages 284–302. Poznan, PL. 2004.

[HPK09] Harmsen, A. F.; Proper, E. H.; Kok, N.: *Informed Governance of Enterprise Transformations.* In (Proper, H. E.; Harmsen, F.; Dietz, J. L., Ed.): *Advances in Enterprise Engineering II.* pages 155–180. Berlin, Heidelberg, Germany. 2009. Springer.

[HV93] Henderson, J. C.; Venkatraman, N.: *Strategic alignment: leveraging information technology for transforming organizations.* *IBM Systems Journal.* 32(1):472–484. 1993.

[HW08] Hafner, M.; Winter, R.: *Processes for Enterprise Application Architecture Management.* In *41^{st} Hawaii International Conference on System Sciences (HICSS 2008).* page 396. Los Alamitos, CA, USA. 2008. IEEE Computer Society.

[IF99] IFIP-IFAC Task Force on Architecture for Enterprise Integration: *Geram: Generalised Enterprise Reference Architecture and Methodology – version 1.6.3.* 1999.

[IF03] IFIP-IFAC Task Force on Architecture for Enterprise Integration: *GERAM: The Generalised Enterprise Reference Architecture and Methodology.* In (Bernus, P.; Nemes, L.; Schmidt, G., Ed.): *Handbook on Enterprise Architecture.* pages 21–63. Berlin, Heidelberg, Germany. 2003. Springer.

[IL08] Isomäki, H.; Liimatainen, K.: *Challenges of Government Enterprise Architecture Work – Stakeholders' Views.* In (Wimmer, M.; Scholl, H. J.; Ferro, E., Ed.): *Electronic Government, 7th International Conference.* pages 364–374. Turin, Italy. 2008. Springer.

[In96] International Organization for Standardization: *ISO/IEC 10746 Reference Model of Open Distributed Processing (RM-ODP).* 1996.

[In99] International Organization for Standardization: *ISO 15704 (1999) ISO/DIS 15704: Industrial automation systems – Requirements for enterprise reference architectures and methodologies. ISO/TC 184/SC5/WG1.* 1999.

[In06] International Organization for Standardization: *ISO 19439:2006 Enterprise integration – Framework for enterprise modelling.* 2006.

[In07] International Organization for Standardization: *ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems.* 2007.

[IT09] IT Governance Institute: *Framework Control Objectives Management Guidelines Maturity Models.* `http://www.isaca.org/Knowledge-Center/cobit` (cited 2010-06-18). 2009.

[JE07] Johnson, P.; Ekstedt, M.: *Enterprise Architecture – Models and Analyses for Information Systems Decision Making.* Studentlitteratur. Pozkal, Poland. 2007.

[JNL06] Johnson, P.; Nordström, L.; Lagerström, R.: *Formalizing Analysis of Enterprise Architecture.* In *Enterprise Interoperability.* pages 35–44. Bordeaux, France. 2006. Springer.

[Jo03] Jonkers, H.; van Burren, R.; Arbab, F.; de Boer, F.; Bonsangue, M. M.; Bosma, H.; ter Doest, H. et al.: *Towards a language for coherent enterprise architecture descriptions.* In *7th IEEE International EDOC Conference (EDOC 2003).* Brisbane, Australia. 2003. IEEE Computer Society.

[Jo04a] Johnson, P.; Ekstedt, M.; Silva, E.; Plazola, L.: *Enterprise Architecture for CIO Decision-Making: On the Importance of Theory.* In *Proceedings of the Second Annual Conference on Systems Engineering Research.* pages 1–10. 2004.

[Jo04b] Jonkers, H.; Lankhorst, M. M.; Buuren, R. v.; Bonsangue, M. M.; Torre, L. W. N. v. d.: *Concepts for Modelling Enterprise Architectures. International Journal of Cooperative Information Systems.* 13:257–287. 2004.

[Jo06] Jonkers, H.; Lankhorst, M. M.; ter Doest, H.; Arbab, F.; Bosma, H.; Wieringa, R.: *Enterprise architecture: Management tool and blueprint for the organisation. Information Systems Frontiers.* 8:63–66. 2006.

[Jo07] Johnson, P.; Lagerström, R.; Närman, P.; Simonsson, M.: *Enterprise architecture analysis with extended influence diagrams. Information Systems Frontiers.* 9:163–180. 2007.

[Jo09] Josey, A. e. a.: *TOGAF Version 9 – A Pocket Guide.* Van Haren Publishing. Wilco, Amersfoort, Netherlands. 2nd edition. 2009.

[Jo10] Jonkers, H.; van den Berg, H.; Iacob, M.-E.; Quartel, D.: *ArchiMate Extension for Modeling TOGAF's Implementation and Migration phases.* `http://www.bizzdesign.com/index.php/component/docman/doc_download/18-archimate-extension-for-modeling-togafs-implementation-and-migration-phases`. 2010.

[Ju07] Jung, J.: *Entwurf einer Sprache für die Modellierung von Ressourcen im Kontext der Geschäftsprozessmodellierung.* PhD thesis. Universität Duisburg-Essen. Berlin, Germany. 2007.

[Ki99] Kirch, J.: *Prozessorientiertes Management von Client-Server-Systemen.* PhD thesis. University of Wiesbaden. 1999.

[Ki08] Kirchner, L.: *Eine Methode zur Unterstützung des IT-Managements im Rahmen der Unternehmensmodellierung.* PhD thesis. Universität Duisburg-Essen. Berlin, Germany. 2008.

[KNS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: *Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)".* Technical Report Heft 89. Institut für Wirtschaftsinformatik (IWi), Universität des Saarlandes. 1992.

[Ku09] Kurpjuweit, S.: *Stakeholder-orientierte Modellierung und Analyse der Unternehmensarchitektur.* PhD thesis. Universität St.Gallen. 2009.

[KUJ09] Källgren, A.; Ullberg, J.; Johnson, P.: *A Method for Constructing a Company Specific Enterprise Architecture Model Framework.* In (Kim, H.-K.; Lee, R. Y., Ed.): *10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, SNPD 2009.* pages 346–351. Daegu, Korea. 2009.

[KW07] Kurpjuweit, S.; Winter, R.: *Viewpoint-based Meta Model Engineering.* In (Reichert, M.; Strecker, S.; Turowski, K., Ed.): *2nd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2007).* LNI. pages 143–161. Bonn, Germany. 2007. Gesellschaft für Informatik.

[KW09] Kurpjuweit, S.; Winter, R.: *Concern-oriented business architecture engineering.* In *SAC'09: Proceedings of the 2009 ACM symposium on Applied Computing.* pages 265–272. New York, NY, USA. 2009. ACM.

[La04] Lankhorst, M. M.; van Buuren, R.; van Leeuwen, D.; Jonkers, H.; ter Doest, H.: *Enterprise architecture modelling-the issue of integration. Adv. Eng. Inform.* 18(4):205–216. 2004.

[La05] Lankhorst, M. M.: *Enterprise Architecture at Work: Modelling, Communication and Analysis.* Springer. Berlin, Heidelberg, Germany. 2005.

[La07] Lageström, R.: *Analyzing System Maintainability using Enterprise Architecture Models. Journal of Enterprise Architecture.* 3:33–42. 2007.

[La08] Lagerström, R.; Chenine, M.; Johnson, P.; Franke, U.: *Probabilistic Metamodel Merging.* In (Bellahsene, Z.; Woo, C.; Hunt, E.; Franch, X.; Coletta, R., Ed.): *CAiSE Forum.* pages 25–28. Montpellier, France. 2008.

[La09] Lankhorst, M. M.: *Enterprise Architecture at Work: Modelling, Communication and Analysis.* Springer. Berlin, Heidelberg, Germany. 2nd edition. 2009.

[LD08] Land, M. O.; Dietz, J. L.: *Enterprise ontology based splitting and contracting of organizations.* In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing.* pages 524–531. New York, NY, USA. 2008. ACM.

[LHS08] Liimatainen, K.; Heikkilä, J.; Seppänen, V.: *A Framework for Evaluating Compliance of Public Service Development Programs with Government Enterprise Architecture.* In *Proceedings of the 2nd European Conference on Information Management and Evaluation.* pages 269–276. London, UK. 2008.

[LJ08] Lagerström, R.; Johnson, P.: *Using Architectural Models to Predict the Maintainability of Enterprise Systems.* In *12th European Conference on Software Maintenance and Reengineering.* pages 248–252. Athens, Greece. 2008. IEEE.

[LMW05a] Lankes, J.; Matthes, F.; Wittenburg, A.: *Architekturbeschreibung von Anwendungslandschaften: Softwarekartographie und IEEE Std 1471-2000.* In *Software Engineering* [se05]. pages 43–54.

[LMW05b] Lankes, J.; Matthes, F.; Wittenburg, A.: *Softwarekartographie : Systematische Darstellungen von Anwendungslandschaften.* In (Ferstl, O.; Sinz, E.; Eckert, S.; Isselhorst, T., Ed.): *Wirtschaftsinformatik 2005.* pages 1443–1462. Bamberg, Germany. 2005. Physica-Verlag.

[LS08] Lankes, J.; Schweda, C. M.: *Using Metrics to Evaluate Failure Propagation and Failure Impacts in Application Landscapes.* In (Bichler, M.; Hess, T.; Krcmar, H.; Lechner, U.; Matthes, F.; Picot, A.; Speitkamp, B. et al., Ed.): *Multikonferenz Wirtschaftsinformatik.* Berlin, Germany. 2008. GITO-Verlag.

[Lu03] Luftman, J. N.: *Competing in the Information Age – Align in the Sand.* Oxford University Press. New York, NY, USA. 2^{nd} edition. 2003.

[Lu09] Luijpers, J.: *White Paper: Project Start Architecture – Version 1.0.* Technical report. Sogeti Nederland B.V. Diemen, The Netherlands. 2009.

[LVP07] 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).* pages 1–20. 2007.

[LW04a] 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.

[LW04b] Lê, L.-S.; Wegmann, A.: *Meta-model for Object-Oriented Hierarchical Systems.* Technical report. Laboratory of Systemic Modeling, Ecole Polytechnique Fédérale de Lausanne. Lausanne, Switzerland. 2004.

[LW05] Lê, L.-S.; Wegmann, A.: *Definition of an Object-Oriented Modeling Language for Enterprise Architecture.* In *38^{th} Hawaii International Conference on System Sciences (HICSS 2005).* pages 179–188. 2005.

[LW06] Lê, L.-S.; Wegmann, A.: *SeamCAD: Object-Oriented Modeling Tool for Hierarchical Systems in Enterprise Architecture.* In *39^{th} Hawaii International Conference on System Sciences (HICSS 2006).* 2006.

[Ma08] 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.

[MaZT07] Magalhães, R.; Zacharias, M.; Tribolet, J. M.: *Making Sense of Enterprise Architectures as Tools of Organizational Self-Awareness (OSA).* In *2^{nd} Workshop on Trends in Enterprise Architecture Research (TEAR 2007).* 2007.

[Mi95] Miller, J.: *Living Systems.* University of Colorado Press. 1995.

[MW04] Matthes, F.; Wittenburg, A.: *Softwarekartographie: Visualisierung von Anwendungslandschaften und ihrer Schnittstellen.* In (Dadam, P.; Reichert, M., Ed.): *GI Jahrestagung (2).* pages 71–75. Ulm, Germany. 2004.

[MWF08] Murer, S.; Worms, C.; Furrer, F. J.: *Managed Evolution. Informatik Spektrum.* 31(6):537–547. 2008.

[My11] Mykhashchuk, M.; Buckl, S.; Dierl, T.; Schweda, C. M.: *Charting the landscape of enterprise architecture management – An extensive literature analysis.* In *Wirtschaftsinformatik.* Zürich, Swiss. 2011.

[MZT10] Marques, J. a.; Zacarias, M.; Tribolet, J.: *A Bottom-Up Competency Modeling Approach.* In (Aalst, W.; Mylopoulos, J.; Sadeh, N. M.; Shaw, M. J.; Szyperski, C.; Albani, A.; Dietz, J. L. G., Ed.): *Advances in Enterprise Engineering IV.* volume 49 of *Lecture Notes in Business Information Processing.* pages 50–64. Springer Berlin Heidelberg. 2010.

[NA07] NATO: *NATO Architecture Framework Version 3.* `http://www.nhqc3s.nato.int/ARCHITECTURE/_docs/NAF_v3/ANNEX1.pdf` (cited 2010-02-25). 2007.

[Nä08] Närman, P.; Schönherr, M.; Johnson, P.; Ekstedt, M.; Chenine, M.: *Using Enterprise Architecture Models for System Quality Analysis.* In *12^{th} IEEE International EDOC Conference (EDOC 2008).* pages 14–23. Washington, DC, USA. 2008. IEEE Computer Society.

[Ne00] Neilson, R.: *Strategic Scenario Planning at CA International. Knowledge Management Review.* 2000.

[Ni06] Niemann, K. D.: *From Enterprise Architecture to IT Governance – Elements of Effective IT Management.* Vieweg+Teubner. Wiesbaden, Germany. 2006.

[No03] Noran, O.: *A Mapping of Individual Architecture Frameworks (GRAI, PERA, C4ISR, CIMOSA, Zachman, ARIS) onto GERAM.* In (Bernus, P.; Nemes, L.; Schmidt, G., Ed.): *Handbook on Enterprise Architecture.* pages 65–210. Berlin, Heidelberg, Germany. 2003. Springer.

[Ob10a] Object Management Group (OMG): *Business Process Model and Notation (BPMN) – Version 2.0.* 2010.

[Ob10b] Object Management Group (OMG): *OMG Unified Modeling Language^{TM}(OMG UML), Infrastructure – Version 2.3 (formal/2010-05-03).* 2010.

[Ob10c] Object Management Group (OMG): *OMG Unified Modeling Language^{TM}(OMG UML), Superstructure – Version 2.3 (formal/2010-05-05).* 2010.

[Of09] Office of Government Commerce (OGC): *Managing Successful Projects with PRINCE2.* The Stationery Office. Norwich, UK. 2009.

[Ös07] Österle, H.; Winter, R.; Hoening, F.; Kurpjuweit, S.; Osl, P.: *Business Engineering: Core-Business-Metamodell. Wisu – Das Wirtschaftsstudium.* 36(2):191–194. 2007.

[ÖW03] Österle, H.; Winter, R.: *Business Engineering – Auf dem Weg zum Unternehmen des Informationszeitalters.* Springer. Berlin, Germany. 2nd edition. 2003.

[PH05] Pulkkinen, M.; Hirvonen, A. P.: *EA Planning, Development and Management Process for Agile Enterprise Development.* In *38th Hawaii International Conference on System Sciences.* Big Island, HI, USA. 2005. IEEE Computer Society.

[PNL07] Pulkkinen, M.; Naumenko, A.; Luostarinen, K.: *Managing information security in a business network of machinery maintenance services business - Enterprise architecture as a coordination tool. J. Syst. Softw.* 80(10):1607–1620. 2007.

[Pu06] Pulkkinen, M.: *Systemic Management of Architectural Decisions in Enterprise Architecture Planning. Four Dimensions and Three Abstraction Levels.* In *39th Hawaii International Conference on System Sciences (HICSS 2006).* volume 8. page 179c. 2006.

[QEJ10] Quartel, D.; Engelsman, W.; Jonkers, H.: *ArchiMate Extension for Modeling and Managing Motivation, Principles and Requirements in TOGAF.* `http://www.bizzdesign.com/index.php/component/docman/doc_download/16-extending-ea-modelling-with-business-goals-and-requirements`. 2010.

[RA09] Riege, C.; Aier, S.: *A Contingency Approach to Enterprise Architecture Method Engineering.* In *Service-Oriented Computing – ICSOC 2008 Workshops.* pages 388–399. 2009.

[RB06] Ross, J. W.; Beath, C. M.: *Sustainable IT Outsourcing Success: Let Enterprise Architecture be your Guide. MIS Quarterly Executive.* 5(4):181–192. 12 2006.

[RBW03] Rychkova, I.; Balabko, P.; Wegmann, A.: *Operational ASM Semantics behind Graphical SEAM Notation.* In *DAIS/FMOODS Ph.D. workshop.* pages 10–19. 2003.

[Re09] Regev, G.; Hayard, O.; Gause, D. C.; Wegmann, A.: *Modeling Service-Level Requirements: A Constancy Perspective.* In *RE 2009, 17th IEEE International Requirements Engineering Conference.* pages 231–236. 2009.

[RNE09] Raderius, J.; Närman, P.; Ekstedt, M.: *Assessing System Availability Using an Enterprise Architecture Analysis Approach.* In *Service-Oriented Computing — ICSOC 2008 Workshops: ICSOC 2008 International Workshops, Sydney, Australia, December 1st, 2008, Revised Selected Papers.* pages 351–362. Berlin, Heidelberg. 2009. Springer-Verlag.

[Ro03] Ross, J. W.: *Creating a Strategic IT Architecture Competency: Learning in Stages. MIS Quarterly Executive.* 2(1). 2003.

[RV08] van Raadt, B. d.; van Vliet, H.: *Designing the Enterprise Architecture Function.* In (Steffen Becker, F. P.; Reussner, R., Ed.): *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.

[RW06] Rychkova, I.; Wegmann, A.: *A Method for Functional Alignment Verification in Hierarchical Enterprise Models*. In *BUSITAL*. 2006.

[RW07] Rychkova, I.; Wegmann, A.: *Formal Semantics for Property-Property Relations in SEAM Visual Language: Towards Simulation and Analysis of Visual Specifications*. In *MSVVEIS*. pages 138–147. 2007.

[RWR06] Ross, J. W.; Weill, P.; Robertson, D. C.: *Enterprise Architecture as Strategy*. Harvard Business School Press. Boston, MA, USA. 2006.

[SB08] Steenbergen, M. v.; Brinkkemper, S.: *Modeling the Contribution of Enterprise Architecture Practice to the Achievement of Business Goals*. In (Papadopoulos, G.; Wojtkowski, W.; Wojtkowski, W.; Wrycza, S.; Zupancici, J., Ed.): *Proceedings of the 17^{th} International Conference on Information Systems Development (ISD 2008)*. pages 609–618. New York, USA. 2008. Springer.

[SB09] Steenbergen, M. v.; Brinkkemper, S.: *The Architectural Dilemma: Division of Work vs. Knowledge Integration*. In *Proceedings of the 4^{th} International Workshop on Business/IT Alignment and Interoperability (BUSITAL'09)*. Amsterdam, The Netherlands. 2009. Springer.

[SBB07a] Steenbergen, M. v.; Berg, M. v. d.; Brinkkemper, S.: *An Instrument for the Development of the Enterprise Architecture Practice*. In (Cardoso, J.; Cordeiro, J.; Filipe, J., Ed.): *ICEIS 2007 - Proceedings of the Ninth International Conference on Enterprise Information Systems, Volume EIS, Funchal, Madeira, Portugal, June 12-16, 2007*. pages 14–22. 2007.

[SBB07b] van Steenbergen, M.; van den Berg, M.; Brinkkemper, S.: *An Instrument for the Development of the Enterprise Architecture Practice*. In (Cardoso, J.; Cordeiro, J.; Filipe, J., Ed.): *ICEIS 2007 – Proceedings of the Ninth International Conference on Enterprise Information Systems, Volume EIS, Funchal, Madeira, Portugal, June 12-16, 2007 (3)*. pages 14–22. 2007.

[SBB08] Steenbergen, M. v.; Berg, M. v. d.; Brinkkemper, S.: *A Balanced Approach to Developing the Enterprise Architecture Practice*. In (Filipe, J.; Cordeiro, J.; Cardoso, J., Ed.): *ICEIS 2007*. pages 240–253. Berlin, Heidelberg, Germany. 2008. Springer.

[Sc01] Scheer, A.-W.: *ARIS – Modellierungsmethoden, Metamodelle, Anwendungen*. Springer. Berlin, Germany. 4^{th} edition. 2001.

[Sc02] Scheer, A.-W.: *ARIS – Vom Geschäftsprozess zum Anwendungssystem*. Springer. Berlin, Germany. 4^{th} edition. 2002.

[Sc04] Schekkerman, J.: *Enterprise Architecture Score Card*. Technical report. Institute For Enterprise Architecture Development. 2004.

[Sc06a] Schekkerman, J.: *Enterprise Architecture Deliverables Guide*. Technical report. Institute For Enterprise Architecture Development. 2006.

[Sc06b] Schekkerman, J.: *Extended Enterprise Architecture Framework Essential Guide*. Technical report. Institute For Enterprise Architecture Development. 2006.

[Sc06c] Schekkerman, J.: *Extended Enterprise Architecture Maturity Model Suppot Guide.* Technical report. Institute For Enterprise Architecture Development. 2006.

[Sc06d] Schekkerman, J.: *How to Survive in the Jungle of Enterprise Architecture Frameworks: Creating or Choosing an Enterprise Architecture Framework.* Trafford Publishing. Victoria, BC, Canada. 2006.

[Sc08a] Schekkerman, J.: *Enterprise Architecture Good Practices Guide – How to Manage the Enterprise Architecture Practice.* Trafford Publishing. Victoria, BC, Canada. 2008.

[Sc08b] Schönherr, M.: *Towards a common terminology in the discipline of Enterprise Architecture.* In (Aier, S.; Johnson, P.; Schelp, J., Ed.): *Pre-Proceedings of the 3rd Workshop on Trends in Enterprise Architecture Research.* pages 107–123. Sydney, Australia. 2008.

[Sc10] Schekkerman, J.: *STREAM – A Successful and Pragmatic 'Managed Diversity' Enterprise Architecture Approach.* 2010.

[se05] sebis: *Enterprise Architecture Management Tool Survey 2005.* Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2005.

[SEJ08] Sommestad, T.; Ekstedt, M.; Johnson, P.: *Combining Defense Graphs and Enterprise Architecture Models for Security Analysis.* In *12th International IEEE Enterprise Distributed Object Computing Conference.* pages 349–355. Munich, Germany. 2008. IEEE Computer Society.

[Sh86] Shewhart, W. A.: *Statistical Method from the Viewpoint of Quality Control.* Dover Publication. New York, NY, USA. 1986.

[Sh90] Shaw, M.: *Prospects for an Engineering Discipline of Software.* *IEEE Softw.* 7(6):15–24. 1990.

[SH93] Spewak, S. H.; Hill, S. C.: *Enterprise Architecture Planning – Developing a Blueprint for Data, Applications, and Technology.* John Wiley & Sons. New York, USA. 1993.

[SHL09] Seppänen, V.; Heikkilä, J.; Liimatainen, K.: *Key Issues in EA-Implementation: Case Study of Two Finnish Government Agencies.* In (Hofreiter, B.; Werthner, H., Ed.): *2009 IEEE Conference on Commerce and Enterprise Computing.* pages 114–120. Vienna, Austria. 2009. IEEE Computer Society.

[Si96] Simon, H. A.: *The Sciences of the Artificial.* MIT Press. Cambridge, Massachusetts, USA. 3rd edition. 1996.

[SK04] Schwarzer, B.; Krcmar, H.: *Wirtschaftsinformatik. Grundzüge der betrieblichen Datenverarbeitung.* Schäffer - Poeschel Verlag Stuttgart. 3rd edition. 2004. 3791021710.

[SS07] Schelp, J.; Stutz, M.: *A Balanced Scorecard Approach to Measure the Value of Enterprise Architecture.* In (Lankhorst, M. M.; Johnson, P., Ed.): *Second Workshop on Trends in Enterprise Architecture Research.* pages 5–11. 2007.

[St73] Stachowiak, H.: *Allgemeine Modelltheorie.* Springer-Verlag. Wien, Austria. 1973.

[St05] Steenbergen, M. v.: *Architecture Maturity Matrix DYA – Version 1.0.* Technical report. Sogeti Nederland B.V. Diemen, The Netherlands. 2005.

[St09] Stelzer, D.: *Enterprise Architecture Principles: Literature Review and Research Directions.* In *4th Workshop on Trends in Enterprise Architecture Research (TEAR 2009).* Stockholm, Sweden. 2009.

[St10a] Steenbergen, M. v.; Bos, R.; Brinkkemper, S.; Weerd, I. v. d.; Bekkers, W.: *The Design of Focus Area Maturity Models.* In (Winter, R.; Zhao, J.; Aier, S., Ed.): *Global Perspectives on Design Science Research.* volume 6105 of *Lecture Notes in Computer Science.* pages 317–332. Springer Berlin / Heidelberg. 2010.

[St10b] Struck, V.; Buckl, S.; Matthes, F.; Schweda, C. M.: *Enterprise Architecture Management from a knowledge management perspective – Results from an empirical study.* In *5th Mediterranean Conference on Information Systems (MCIS2010), Tel Aviv.* 2010.

[SW08] Schelp, J.; Winter, R.: *On the Interplay of Design Research and Behavioral Research – A Language Community Perspective.* In (Vaishanvi, V.; Baskerville, R., Ed.): *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.

[SW09] Schelp, J.; Winter, R.: *Language communities in enterprise architecture research.* In *DESRIST '09: Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology.* pages 1–10. New York, NY, USA. 2009. ACM.

[SZ92] Sowa, J. F.; Zachman, J. A.: *Extending and Formalizing the Framework for Information Systems Architecture. IBM Systems Journal.* 31(3):590–616. 1992.

[Sz09] Szyszka, B.: *Analysis and Classification of Maturity Models in Enterprise Architecture Management.* Bachelor's thesis. Fakultät für Informatik, Technische Universität München. 2009.

[Th09a] The Open Group: *TOGAF "Enterprise Edition" Version 9.* `http://www.togaf.org` (cited 2010-02-25). 2009.

[Th09b] The Open Group: *TOGAF Version 9 – A Manual.* Van Haren Publishing. 9th edition. 2009. 978-90-8753-230-7.

[Va01] Vasconcelos, A.; Caetano, A.; Neves, J.; Sinogas, P.; Mendes, R.; Tribolet, J. M.: *A Framework for Modeling Strategy, Business Processes and Information Systems.* In *5th IEEE International EDOC Conference (EDOC 2001).* pages 69–81. IEEE Computer Society. 2001.

[Va04] Vasconcelos, A.; Pereira, C. M.; Sousa, P.; Tribolet, J. M.: *Open Issues On Information System Architecture.* In *Proceedings of the 6th International Conference on Enterprise Information Systems (ICEIS 2004).* 2004.

[VS08] Valtonen, K.; Seppänen, V.: *Adaptation and adoption of Finnish government EA method – Preliminary guidelines for method users.* Ministry of Finance. 2008.

[VSL09] Valtonen, K.; Seppänen, V.; Leppänen, M.: *Government Enterprise Architecture Grid Adaptation in Finnland.* In *Hawaii International Conference on System Sciences.* Los Alamitos, CA, USA. 2009. IEEE Computer Society.

[VST03] Vasconcelos, A.; Sousa, P.; Tribolet, J. M.: *Information System Architectures: Representation, Planning and Evaluation. Journal of Systemics, Cybernetics and Informatics.* 1(6):78–84. 2003.

[VST05] Vasconcelos, A.; Sousa, P.; Tribolet, J. M.: *Information System Architecture Evaluation: From Software to Enterprise Level Approaches.* In *12th European Conference On Information Technology Evaluation (ECITE 2005).* 2005.

[VST07] Vasconcelos, A.; Sousa, P.; Tribolet, J. M.: *Information System Architecture Metrics: an Enterprise Engineering Evaluation Approach. The Electronic Journal Information Systems Evaluation.* 10(1):91–122. 2007.

[VST08] Vasconcelos, A.; Sousa, P.; Tribolet, J. M.: *Enterprise Architecture Analysis - An Information System Evaluation Approach. Enterprise Modelling and Information Systems Architectures.* 3(2):31–53. 2008.

[Wa05] Wagter, R.; van den Berg, M.; Luijpers, J.; van Steenbergen, M.: *Dynamic Enterprise Architecture: How to Make IT Work.* John Wiley. 2005.

[We02] Wegmann, A.: *The Systemic Enterprise Architecture Methodology (SEAM).* Technical report. EPFL. 2002.

[We03] Wegmann, A.: *On the Systemic Enterprise Architecture Methodology (SEAM).* In *SEAM). Published at the International Conference on Enterprise Information Systems 2003 (ICEIS 2003.* pages 483–490. 2003.

[We05] Wegmann, A.; Balabko, P.; Lê; Regev, G.; Rychkova, I.: *A Method and Tool for Business-IT Alignment in Enterprise Architecture.* In (Belo, O.; Eder, J.; ao e Cunha, F.; Pastor, O., Ed.): *Proceedings of the CAiSE'05 Forum.* pages 113–118. Porto, Portugal. 2005.

[We07a] Wegmann, A.; Lê, L.-S.; Regev, G.; Wood, B.: *Enterprise modeling using the foundation concepts of the RM-ODP ISO/ITU standard. Inf. Syst. E-Business Management.* 5(4):397–413. 2007.

[We07b] Wegmann, A.; Regev, G.; Rychkova, I.; Lê, L.-S.; Julia, P.: *Business and IT Alignment with SEAM for Enterprise Architecture.* In *11th IEEE International EDOC Conference (EDOC 2007).* pages 111–121. 2007.

[We08] Wegmann, A.; Kotsalainen, A.; Matthey, L.; Regev, G.; Giannattasio, A.: *Augmenting the Zachman Enterprise Architecture Framework with a Systemic Conceptualization.* In *12th IEEE International EDOC Conference (EDOC 2008).* pages 3–13. 2008.

[WF06] Winter, R.; Fischer, R.: *Essential Layers, Artifacts, and Dependencies of Enterprise Architecture.* In *1st Workshop on Trends in Enterprise Architecture Research (TEAR 2006).* page 30. Washington, DC, USA. 2006. IEEE Computer Society.

[Wi07] Wittenburg, A.: *Softwarekartographie: Modelle und Methoden zur systematischen Visualisierung von Anwendungslandschaften.* PhD thesis. Fakultät für Informatik, Technische Universität München, Germany. 2007.

[Wo10] Wout, J. v.; Waage, M.; Hartman, H.; Stahlecker, M.; Hofman, A.: *The Integrated Architecture Framework Explained.* Springer. 2010.

[WR04] Weill, P.; Ross, J. W.: *IT Governance – How Top Performers Manage IT Decision Rights for Superior Results.* Harvard Business School Press. Boston, Massachusetts, USA. 2004.

[WRL05] Wegmann, A.; Regev, G.; Loison, B.: *Business and IT Alignment with SEAM.* In *REBNITA Workshop at 14th International Requirements Engineering Conference.* 2005.

[WS08] Winter, R.; Schelp, J.: *Enterprise architecture governance: the need for a business-to-IT approach.* In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing.* pages 548–552. New York, NY, USA. 2008. ACM.

[WW02] Webster, J.; Watson, R. T.: *Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly.* 26(2):xiii—-xxiii. 2002.

[Za87] Zachman, J. A.: *A framework for information systems architecture. IBM Syst. J.* 26(3):276–292. 1987.

[Za10] Zacarias, M.; Pinto, H. S.; Magalhães, R.; Tribolet, J.: *A 'context-aware' and agent-centric perspective for the alignment between individuals and organizations. Inf. Syst.* 35:441–466. 2010.