ICIS 2008 Proceedings

International Conference on Information Systems (ICIS)

2008

Situational Architecture Engineering (SAE) - Improving Strategic Change Through Architecture Methods

Susanne Leist
University of Regensburg, susanne.leist@wiwi.uni-regensburg.de

Gregor Zellner
University of Regensburg, gregor.zellner@wiwi.uni-regensburg.de

Situational Architecture Engineering (SAE) – Improving Strategic Change through Architecture Methods

Ingénierie d'architecture en situation – Améliorer le changement stratégique au moyen de méthodes d'architecture

Leist Susanne
University of Regensburg
Universitätsstraße 31, 93053 Regensburg
susanne.leist@wiwi.uni-regensburg.de

Zellner Gregor
University of Regensburg
Universitätsstraße 31, 93053 Regensburg
gregor.zellner@wiwi.uni-regensburg.de

Abstract

Market and environmental requirements call for constant changes in enterprises. To be able to record these changes in a structured way and to manage them it is helpful to use enterprise architectures as stable regulation frameworks. To support the development and the adaptation of the enterprise architectures there are numerous architecture methods (e.g. Zachman Framework, ARIS (Architecture of Integrated Information Systems), TOGAF (The Open Group Architecture Framework), DoDAF (Department of Defense Architecture Framework), BEN (Business Engineering Navigator), Semantic Object Model (SOM)), however, they often lack the necessary flexibility to enable a construction process adaptable to the given situation. This article presents a first approach towards making architecture methods flexible: meta models of the architecture frameworks of selected methods are generated and integrated into a joint meta model. The latter supports the situational adaptability of the enterprise architecture as the architecture method applied has been adapted.

Keywords: Framework, Meta Model, Integration, Situational Architecture Engineering

Résumé

Les méthodes d´architecture d'entreprise (par exemple TOGAF, DoDAF) constituent des moyens pour soutenir le changement stratégique auquel les entreprises sont confrontées. Cependant, il leur manque souvent la flexibilité nécessaire pour permettre un processus de construction adaptable à la situation donnée. Cet article présente une première approche pour rendre flexible les méthodes d'architecture à l'aide de méta-modèles d'architecture.

Abstract in Native language

Unternehmensarchitekturmethoden (z.B. TOGAF, DoDAF, ARIS, Buisness Engineering) unterstützen die Anpassung an den strategischen Wandel, dem die Unternehmen ausgesetzt sind. Jedoch fehlt es diesen Methoden meist an der nötigen Flexibilität, die Anpassungen situativ vornehmen zu können, da sie in ihren Ausprägungen hinsichtlich der Betrachtungsebenen des Unternehmens (Framework der Methode) meist begrenzt sind. Dieser Artikel liefert einen ersten Ansatz, Architekturmethoden flexibler zu gestalten, indem an Hand ausgewählter Architekturmethoden ein integriertes Metamodell aus deren Frameworks erstellt wird, welches alle Betrachtungsebenen und Sichten vereint. Somit können eine Vielzahl von Architekturmodellen situationsspezifisch abgeleitet und ausgearbeitet werden, was eine flexible Anpassung an unterschiedliche Veränderungsszenarien ermöglicht.

Introduction

Business systems are open, goal-oriented, socio-technical systems (Ferstl and Sinz 2006) – like enterprises – which continuously have to adapt to the changes in market and environment. Strategic decisions are necessary to deal with that change. Therefore a flexible adaption of the enterprise is needed to quickly implement the changes in strategy throughout all levels of the organization. Enterprise architectures are indispensable instruments to meet these changes in a holistic way (Niemann 2006). Obviously, a change in strategy is the trigger for an extensive change within an organization. Changes in strategy only get real if changes in business processes and information technology (IT) were derived also.

To manage the complexity of the change and the design of enterprises, it can be helpful to break the business system into different components. The coactions of the different components are ensured by means of a logic construct. Architectures constitute such a construct (Zachman 1987). They enable the structuring of the enterprise by describing different architecture levels (e.g. processes of the enterprise) or taking different views on these levels, as is the case e.g. with the Zachman Framework (cf. (Zachman 1987)).

Even though different approaches exist as regards the elaboration and representation of enterprise architectures (e.g. Zachman, TOGAF, DoDAF), there often is – nonetheless – a lack of support concerning the further development or alteration of existing enterprise architectures. Thus it is a great challenge to flexibly adapt archtitectures to be able to react to the change. In particular, in fields of application which stand out for a high degree of novelty and for their specific tasks one is depending on a flexible construction process which is adaptable to the given situation (Baumöl 2005). That is why the situational adaptation of architectures – we call it "Situational Architecture Engineering" (SAE) – presents a promising approach.

The concept of the situational adaptation is taken from Method Engineering. Method Engineering denotes the discipline in which methods, techniques, and tools are designed, constructed, and adapted to support the development of application systems (Brinkkemper 1996; Tolvanen et al. 1996). In addition, influence factors of the construction process are examined, precisely characteristics of the project and development environment, of the problem domain, and the goal set in order to integrate them into the construction process (cf. (Brinkkemper 1996; Karlsson 2004)). More recent publications extend the focus of study and examine the application of Method Engineering in different domains (e.g. organizational change (Baumöl 2005)). Situation reference results, on the one hand, from finding general starting points when developing the methods which allow for a flexible design. On the other hand, project situations as well as the development environment are examined for indications of the correct construction of the methods.

Our research deals with the first challenge (flexibility) and looks for starting points to make architecture methods flexible to be able to adapt the enterprise architecture, in anticipation of different situations. The objective of our research is to support Situational Architecture Engineering (SAE) as an approach for improving strategic change. Therefore we aim to construct an architecture method for situational adaption including all relevant models, meta models, and procedure models. This article contains first results of this research and focuses the "core" of such a method which is the meta model of the architecture framework.

The research process follows the principles of Design Science Research (Hevner et al. 2004; Peffers et al. 2006). The aim of this approach is to elaborate knowledge of a problem field and the respective solution by means of artefacts (e.g. models and methods) (Hevner et al. 2004). To better handle this research process there is a guideline (Hevner et al. 2004) comprising the two major phases "Develop/Build" and "Justify/Evaluate" (Hevner et al. 2004). These phases can be found in further publications on Design Science, however with different emphases (March and Smith 1995; Takeda et al. 1990; Vaishnavi and Kuechler 2004).

Our paper focuses the phase "Develop/Build" and considers the meta models of the architecture frameworks as our artifacts. The paper is organized as follows: after a short description of the terms, the elements, and the benefits of enterprise architectures the flexibility of frameworks and the connection between strategy and architecture is explained. Subsequently, the requirements and the realization of the situational adaptation of architectures as the basis for their further development are discussed. As an important fundament for the situational adaptation of enterprise architectures an integrated framework meta model is derived – based upon the framework meta models (artifacts) of selected architecture methods. Since the integrated framework meta model is complex, it is only illustrated by examples. The summary of important results as well as an outlook conclude the article. The evaluation ("Justify/Evaluate") of our approach will have to be done in further research projects.

Conceptual Basis

To provide a basis for the following article, we first discuss the term and elements, and the benefits of architectures. Then we address the flexibility of frameworks and explain the connection between strategy and architectures.

Term and Elements of an Architecture

In this article, the term architecture refers to a business system comprising the specification of its components and their relationships as well as construction rules (procedure, linguistic conventions & rules) for the generation and the design of the system (IEEE 2000).

The business system as a reference point of an architecture is generally specified in more detail. If we speak, for example, of an enterprise architecture, the term business system refers to an enterprise while the information system (IS) architecture relates to the information system of the enterprise. At the same time, the IS architecture can be considered as part of an enterprise architecture (Figure 1).

Architectures are usually described by means of models (Ferstl and Sinz 2006). The simplest form of a model provides a specific perspective of the architecture. For instance, application systems (the components of an applications architecture) as well as their relationships can be represented in a *(simple) model*. By definition, construction rules have to be laid down which can be described by means of *meta models* and *procedure models*. Meta models specify the available types of components as well as the types of relationships whereas the procedure model defines activities for the development and for the adaptation of the architecture descriptions. In summary, there are three different types of models (Figure 1) which, by definition, have to be specified when describing an architecture. *(Simple) models* which represent the different perspectives on the architecture, *meta models* which specify the elements of the architecture and their relationships, and *procedure models* which determine activities for the development and the adaptation. These different model types represent different levels of abstraction: meta models represent the highest level of abstraction whereas the procedure models and (simple) models represent a lower level (object level).

To support a systematic development and a consistent and continuous adaptation of the enterprise architecture and the IS architecture, numerous architecture methods (e.g. TOGAF, ARIS) with different focuses have been generated. Even if the term method is not normally used in this context – usually terms such as (architecture) framework (e.g. Zachman Framework) or architecture (e.g. architecture of integrated information systems) are utilized -, it is unambiguous, what cannot be said of the above examples. At the same time, the term method implies a certain standard as regards the quality of the support provided since important elements of a method are defined (cf. (Brinkkemper 1996)). A method embodies a set of concepts that determine what is perceived (techniques), a set of linguistic conventions and rules (meta model) which govern how the perception is presented and communicated, and a set of procedural guidelines which state in what order and how representations are derived or transformed (procedure model) ((Smolander et al. 1991) see also (Brinkkemper 2000; Tolvanen 1996)). To specify a method, it additionally needs a description of participating roles (for the development and management of the architecture models) and the specification of the result documents (Gutzwiller 1994; Heym 1993). It should be mentioned that roles and procedure are related to each other. If there is no procedure provided by a method, the definition of roles for the development process does not make any sense. To summarize, a method consists of the five constitutive elements: *linguistic conventions & rules* (*meta model), procedure, techniques, roles* and *result document*. Apart from this, further important elements for the development and adaptation of architectures are determined (Leist and Zellner 2006). Especially the architecture framework (see Figure 1) which determines the different types of description for different perspectives (Zachman 1987) and models of the architecture elements is of particular importance. A framework is a logical structure for classifying and organizing complex information (The CIO Council 1999).

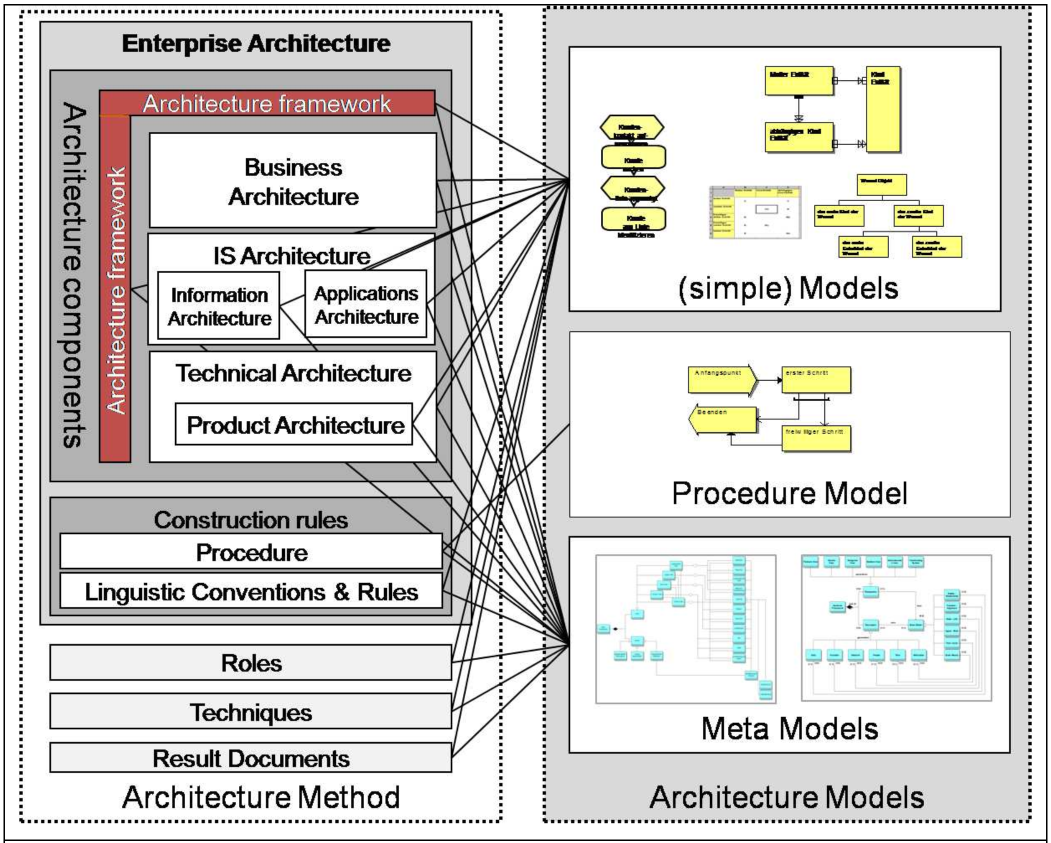

Figure 1. Elements of an Architecture

Benefits of Architectures

Numerous literature sources provide evidence of the benefits of architectures. The most obvious benefit lies in their contribution to the structuring which can reduce complexity and create transparency (a.o. (Dickson and Wetherbe 1985; Goikoetxea 2007; Sowa and Zachman 1992)). The architecture models take different perspectives on the enterprise or the information system, focus selected aspects (e.g. data in data models or applications in the application landscape), and ignore non-relevant aspects. In addition, the architecture structures the different perspectives in a framework. The structuring approach of an architecture does thus provide an essential contribution to increasing both flexibility and adaptability (a.o. (Allen and Boynton 1991)).

However, the significance of an enterprise or IS architecture, as its definition suggests, goes considerably beyond its contribution to structuring (see as well (Goethals et al. 2004; Pereira and Sousa 2004). Other benefits of an enterprise architecture are that e.g. economies of scale can be achieved by providing shared services for the organization or that consistency, accuracy, and timeliness of the IT-driven resources can be improved (Goikoetxea 2007). Organizational bottlenecks or inconsistencies, too, are detected and can be eliminated. At the same time, the transparency makes it possible to improve the information logistics, i.e. the architecture ensures that the information infrastructure efficiently supports the decision makers in the enterprise (the right information at the right time in the right format etc.).

In addition, the architecture supports the realization of the IS strategy (The Open Group 2006b). The architecture descriptions serve as a buffer between strategic business objectives (and thus the strategic IT objectives) and the physical components of the information infrastructure (the implemented information systems and the basic infrastructure). The planning by means of architectures facilitates the alteration of strategic business objectives without modifying the information systems.

Without logical structures – and that is what architectures are – it would not be possible to illustrate the complexity of enterprises. They enable the segmentation of enterprises into comprehensible units without losing sight of the overall context. Architectures thus contribute to developing an enterprise infrastructure which makes it possible to adapt flexibly to changes. (Sowa and Zachman 1992)

Flexibility of Frameworks and Architecture Methods

The framework, as the core element of an architecture method (Figure 1), structures the different dimensions (e.g. perspectives, views, modeling layers) of an architecture and determines the components of the architecture. A framework defines the relevant business requirements that are necessary to evolve the architecture (Whitman et al. 2001). As the different architecture methods have different purposes (e.g. the Zachman Framework for defining and controlling the integration of the elements of an information system (Zachman 1987); DoDAF provides guidance and rules for developing architectures across the department of defense (DoD) (DoDAF 2007b)), each method must be regarded according to its specific purpose. A good possibility to integrate different methods with regard to their specific characteristics is meta model integration. Meta model integration supports the search of similarities (see section on „integration of the framework meta models“) and highlights similar aspects. Since specific characteristics of the method are identified as well, meta model integration provides an integrated method in terms of a ‘set union’ (as in the ‘set theory’).

It is the aim of developing an architecture to provide a picture of the entire enterprise infrastructure by taking into account all possible views that may help to achieve the goals of the enterprise (Whitman et al. 2001). Models (see Figure 1) describe that picture and need to be changed if strategic change happens. The more flexible the models can be modified or generated, the better the infrastructure of the enterprise can be adapted and the better the change can be supported or controlled. That is important in a “world of change” because “the infrastructure can become a firm´s capability if management is able to deploy it in a way that is unique and strategic for the firm” (Ciborra et al. 2000). We assume that the more dimensions the framework contains, the more possibilities exist to represent the infrastructure of a firm´s architecture by its underlying models. The size of the framework thus partially determines the degree of the flexibility of an architecture method because the more extensive the framework is, the more different model types (e.g. process model, data model) can be assorted and used to support the change.

Hence, due to these characteristics, architecture is an essential instrument for the strategic planning and the development of the enterprise.

Strategy and Architecture

Strategic decisions cause organizational change and affect all levels of an enterprise like e.g. business processes or information systems. Because of their interrelationship IS strategy affects and is affected by changes in organizational strategies (Pearlson and Saunders 2006). In contrast to changes in IS (-infrastructure), strategic decisions are made in shorter time intervals. Therefore it is important to adapt to the strategic changes in both a situational and holistic way. Enterprise architecture methods can help achieving this goal (Niemann 2006). Their frameworks, procedures, and guidelines support a structured implementation of new strategies at almost all levels of an enterprise. Even if different enterprise architecture methods offer different approaches to structure enterprise architecture models (e.g. DoDAF: operational view, systems view, technology view, all view (DoDAF 2007b)), all structures include a business and application perspective (Niemann 2006). It is the business architecture that deals with the visions and strategies and aligns the IT to the implementation of the strategies (Niemann 2006) which determine the new "to-be" enterprise architecture (Goikoetxea 2007). Business success significantly depends on the harmony of business strategy, IT strategy, organizational infrastructure, and processes (Luftman et al. 1993). This relation addresses the Strategic Alignment Model (Henderson and Venkatraman 1999). It describes the dependencies of business and IT and presents four different perspectives to align both (Henderson and Venkatraman 1999). Enterprise architecture methods are a means to support the alignment of business strategy and IT, a tool to implement changes in strategy. Architectures determine an effective strategy which forms the baseline of any successful business strategy (Whitman et al. 2001). According to (Pearlson and Saunders 2006) "architecture translates strategy into infrastructure" (Figure 2). This is an outstanding connection between strategy and architecture.

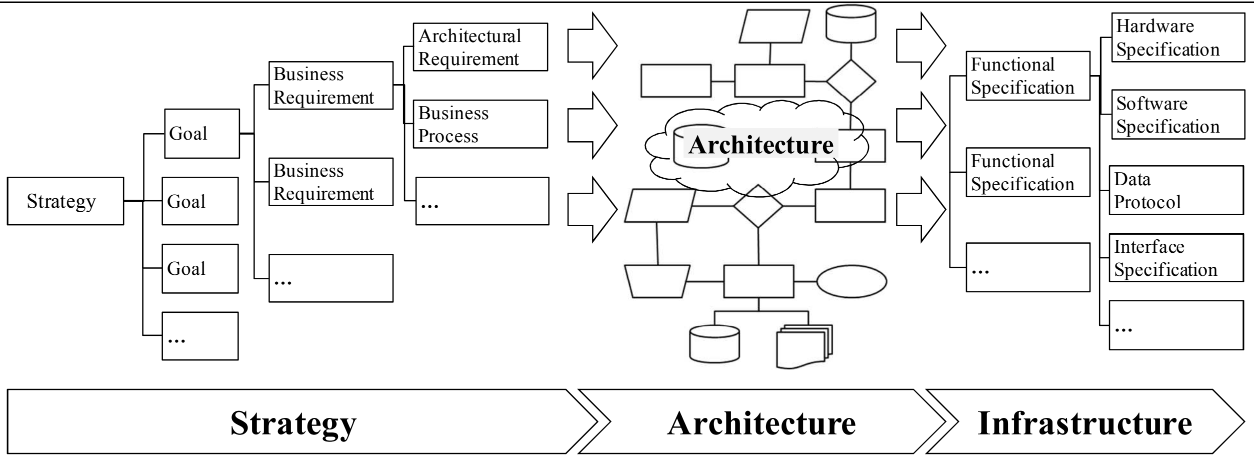

Figure 2. Strategy and Architecture (according to (Pearlson and Saunders 2006))

Situational Adaptability of Architectures as a Basis for their Development

The increase of the flexibility and strategic adaptability of the enterprise is evidenced by the possibilities to further develop enterprise architectures. In the following, the starting points will be examined, and it will be shown how methods can provide support.

Measures to Develop Architectures

In particular, the further development of the architecture is carried out for the three above-mentioned model types (Figure 1). For instance, the implementation of new processes or software components requires extensions of the process models or the application models. In this case, the changes only refer to individual, existing models. These changes have to be carried out either within the framework of the given modeling language or by extending the modeling language. For example, the process model can be extended by functions of the new process. If, additionally, types of functions are supposed to be differentiated (e.g. functions adding value and functions not adding value), the modeling language (meta model) must be extended.

Strategic changes can, however, also necessitate the implementation of a new perspective on the architecture. For instance, evaluations as to the degree of automation are of importance to the further development of the IT infrastructure and require the integration of software systems and operational functions. If the modeling of such an integrated perspective had not yet been envisaged, it would at least be necessary to define a new model by means of the meta model of the architecture framework to represent this perspective. Adding or deleting a perspective change the structure of the architecture.

Adaptations of the architecture can also refer to activities described in the procedure model of the architecture. Hence, for instance sequences of alignment processes for the determination of standards can be changed or new activities (e.g. for the collection of requirements of the stakeholders) implemented.

Thus an architecture provides at least three starting points to execute further developments in a structured way. To increase the flexibility of an enterprise it has to be stipulated that such adaptations do not cause inconsistencies with the existing architectures. Accordingly, architecture methods should make visible emerging inconsistencies or even prevent them. According to the above-mentioned starting points the methods have to meet the following requirements:

- A meta model should exist for each of the models that has to be developed for the application of the method so that extensions of the models can be executed consistently.
- Apart from that, a meta meta model should have been developed that ensures the consistency when extending the modeling language (of the meta model) of a model.
- To apply changes to the structure of the architecture a meta model of the architecture framework should exist.

State of the Art Regarding the Flexibility of Architecture Methods

There are only few contributions which deal with architecture methods (a.o. (Schekkerman 2004; The Open Group 2006a). Most of them (e.g. (Goethals 2003; Schekkerman 2004; The Open Group 2006a) provide similarly structured descriptions of the architecture methods and do not dwell on the comparison or the further development of the methods. In a contribution by (Leist and Zellner 2006), selected architecture methods were examined in regard to their suitability to provide methodic support, and gaps in these methods were identified. Table 1 shows the results of this examination in the field „Methodic Support".

Table 1. Gaps of Existing Architecture Methods

		ARIS	C4ISR/DoDAF	FEAF	MDA	TEAF	TOGAF	Zachman Framework	Business Engineering
Methodic Support	Simple models (result documents)	●	●	●	●	●	⊙	●	●
	Meta model (linguistic conventions & rules)	●	⊙	○	●	⊙	○	⊙	●
	Role model	○	⊙	●	⊙	●	⊙	○	○
	Techniques	●	●	○	⊙	○	⊙	○	●
	Procedure model	○	●	●	●	●	●	⊙	⊙
Framework	Views (stakeholders)	○	○	/	/	/	⊙	●	●
	Perspectives (data, functions, etc.)	●	⊙	/	/	/	●	●	○
	Phase concepts (requirements definition etc.)	●	●	/	/	/	○	○	○
	Meta model of the architecture framework	○	○	○	○	○	○	○	○

Legend: ● fully fulfilled ⊙ partly fulfilled ○ not fulfilled / not examined

Source: Extended representation according to (Leist and Zellner 2006)

One result of the examination is the fact that none of the selected architecture methods offer an integrated, methodic support to represent (partial) enterprise architectures. The extension of this summary table (Table 1) by the category "framework" shows further gaps of the methods. Even though most of the meta models are defined by parts of the architecture methods thus offering the possibility of changes to the architecture models, there is a lack of meta meta models as well as of meta models for the architecture frameworks themselves so that, when for example implementing a new perspective, consistency breaches cannot be reliably prevented.

Integrated Framework Meta Model

In the following, a first approach is presented to develop the framework meta models of six selected architecture methods. Those frameworks are helpful to develop and adapt the architecture. Subsequently, the derived frameworks are integrated into one framework meta model. The choice of the architecture methods is based on how widely they are known in science and practice. The sources for the examination of the methods are listed in Table 2.

Table 2. Selected Architecture Methods

Method	**Source**
ARIS	(Scheer 2001; Scheer and Schneider 2006)
BEN	(Braun and Winter 2005; Winter and Fischer 2007)
DoDAF	(DoDAF 2007a ; DoDAF 2007b)
TOGAF	(The Open Group 2006b)
SOM	(Ferstl and Sinz 2001; Ferstl and Sinz 2006)
Zachman Framework	(Sowa and Zachman 1992; Zachman 1987)

With the exception of TOGAF, all methods describe their frameworks in detail. TOGAF claims to be applicable for different architecture methods thus dispensing with specifying concrete dimensions of a framework. Still, the architecture models which were developed by means of TOGAF are assigned to certain domains corresponding to the dimensions of a framework. Even if, in principle, TOGAF is intended to enable the user to carry out a different allocation thus focusing on different architecture methods, a clear structure can be derived from the method. This clear structure is taken as the basis of the examination of TOGAF.

As an example, in the following the meta model of the framework of the method by Zachman will be described. Further down, the integration procedure is explained, and, finally, the “integrated framework meta model” is introduced.

Meta Model of Selected Architecture Frameworks: The Zachman Framework

The Zachman Framework for Information System Architectures (Sowa and Zachman 1992; Zachman 1987) enables to examine systems from different perspectives and to describe the connection between the perspectives (Sowa and Zachman 1992). The framework itself is a taxonomy consisting of 30 cells distributed to six columns and 5 lines (Sowa and Zachman 1992). The columns represent different levels of abstraction or different possibilities to describe the real world (“descriptions”) (Sowa and Zachman 1992). In so doing, the different possibilities of description result from the different aspects of an object (e.g. the material or the function of an object, etc.) (Zachman 1987). In the Zachmann Framework, these different aspects are highlighted by the questions “what“, “how“, “where“, “when“, and “why“, and are answered with the help of the column variables “data“, “function“, “network“, “people“, “time“, and “motivation“ (Sowa and Zachman 1992).

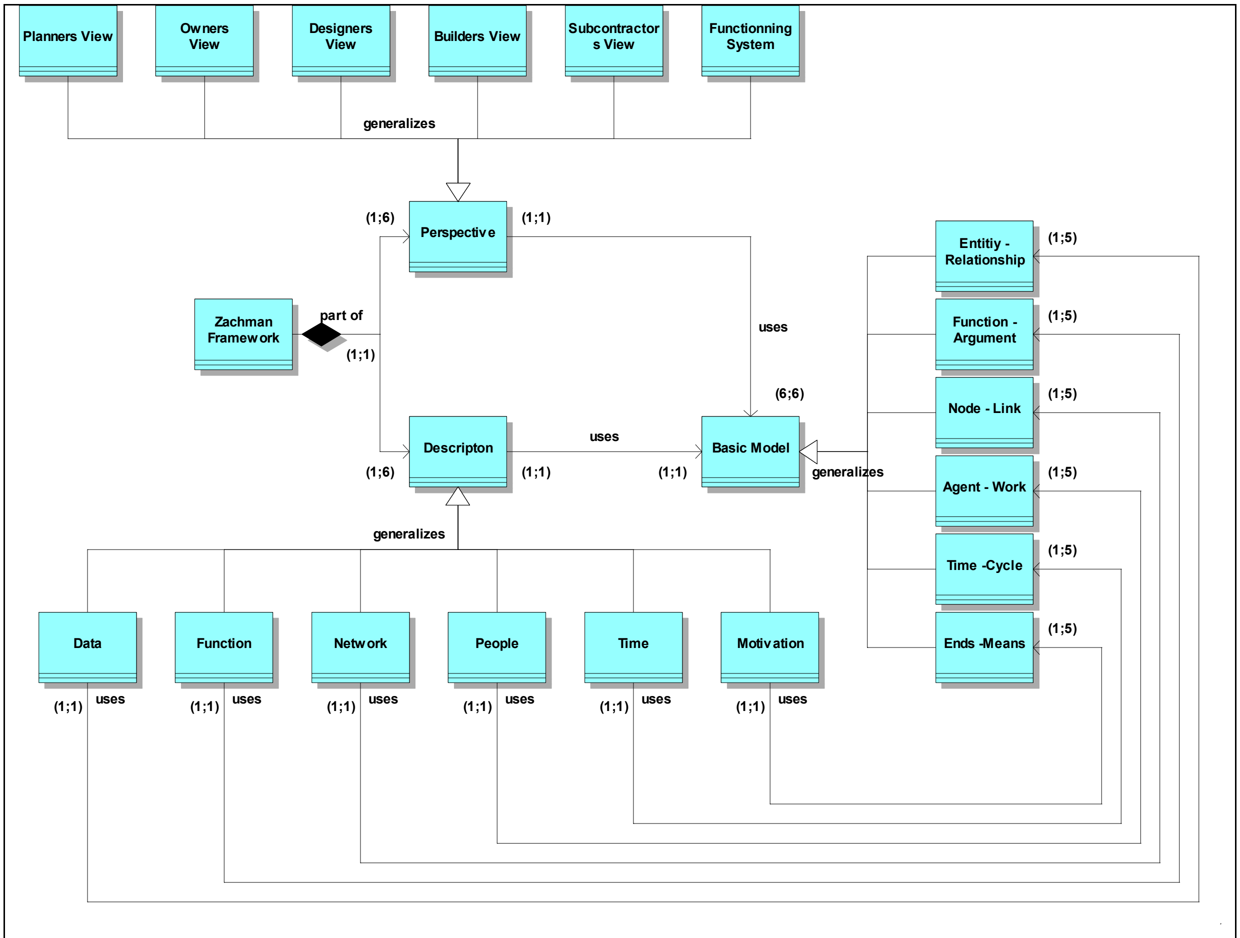

Figure 3. Zachman Framework Meta Model

The lines in the framework mirror the different perspectives of the people taking part in the design. The six perspectives (planner, owner, designer, builder, subcontractor, and functioning system) represent all critical models which are necessary to develop the system and complement one another in that the respective subordinate perspective determines the restrictions for the superordinate one. Linking the cells in a line thus provides a complete description of this perspective. The columns represent different levels of abstraction so that a reduction of complexity is made possible for each individual model.

In the Zachman Framework, meta models for the business model (owner´s view), the system model (designer´s view), and the technology model (builder´s view) are described (Sowa and Zachman 1992). In so doing, the relative meta models only represent the connections of the individual cells within the perspectives for the description "what", "how" , and "where". Thus there is no complete meta model, neither as regards the individual result documents nor in regard to the connection of the relative result documents in the individual cells (Leist and Zellner 2006). Moreover, there is no meta model of the framework. Figure 3 shows the Zachman Framework meta model by means of the perspectives and the description which form the regulation framework.

The essential features of the Zachman Framework lie in the division into perspectives and descriptions and the existence of base models on the description levels.

Integration of the Framework Meta Models

In literature, there are numerous approaches dealing with the integration of different methods or models (cf. (Batini et al. 1992; Zivkovic et al. 2007). In many approaches, the integration is carried out by means of meta models: in so doing, the central problem is the determination of homonymous and synonymous objects of the models or the methods which should be integrated. An additional problem arises from the use of different modeling languages:

during integration the question of equivalence of the modeling concepts of the languages has to be resolved. In this article, the individual meta models are modeled in one and the same language so that the said problem does not arise. Therefore approaches from schema integration are especially appropriate (cf. (Batini et al. 1992; Larson et al. 1989)) with which different, locally developed, data models should be subsequently integrated into a global model. Accordingly, the following procedure has to be carried out:

1. Selection of the comparative strategy: the meta models are examined in pairs and are integrated.
2. A similarity analysis is carried out and conflicts are interpreted: To identify homonymous or synonymous meta model objects types, first object types with similar denotations are looked for by means of the lexical analysis (cf. (Fantoni et al. 2007)) and then object types with similar characteristics (e.g. attributes, relations with other object types) are looked for (cf. (Batini and Lenzerini 1984; Larson et al. 1989)). On this basis, conflicts (synonymous or homonymous terms) are identified and also, if helpful, interpreted by means of rules (cf. (Batini and Lenzerini 1984; Larson et al. 1989)). Finally, synonymous designations for an object type are substituted by one term and homonymous designations by different terms. Additionally, transformation rules according to (Zivkovic et al. 2007) can be used.
3. Determination of the integrated framework meta model

In the following, the procedure is exemplified by means of the integration of parts of the meta models of the Zachman Framework and of ARIS (Figure 4).

At first sight, those object types of the meta models stand out which, at least in parts, have similar denotations (”data“, ”data view“, and ”function“, “function view”). Each of them is integrated in a similar structure, i.e. it is part of a generalization relationship. Other object types attract attention because they are placed in the center of their meta model. They do not show similarities as regards their denotations but as regards their characteristics. ARIS has two object types (“view“ and ”phase“), and the Zachman Framework has three (”description“, ”basic model“ and ”perspective“) which present the origin of all further object types of the meta models. Since they have specialization relationships to all other object types, they can thus be interpreted as the dimensions of the framework. The comparison of their specialization relationships gives indications as to whether these object types have further similarities. In addition, we compare the object types (subtypes) which belong to the specialization relationships. In this context, these subtypes with similar denotations and related meanings suggest the proximity of the object types “view“ (ARIS) and ”description“ (Zachman Framework) as well as ”phase“ (ARIS) and ”perspective“ (Zachman Framework). For instance, apart from subtypes with similar denotations (“data“/”data view“ and ”function“/”function view“), the definitions of the same subtypes overlap. Likewise, there are subtypes with different denotations but similar characteristics (“organization view“ and ”people“). The subtypes “function view“ and “motivation“, too, have one similar attribute since they both describe objectives of functions. Yet the similarity of a single attribute, in comparison with differences regarding denotation, definition, and structure, is less important.

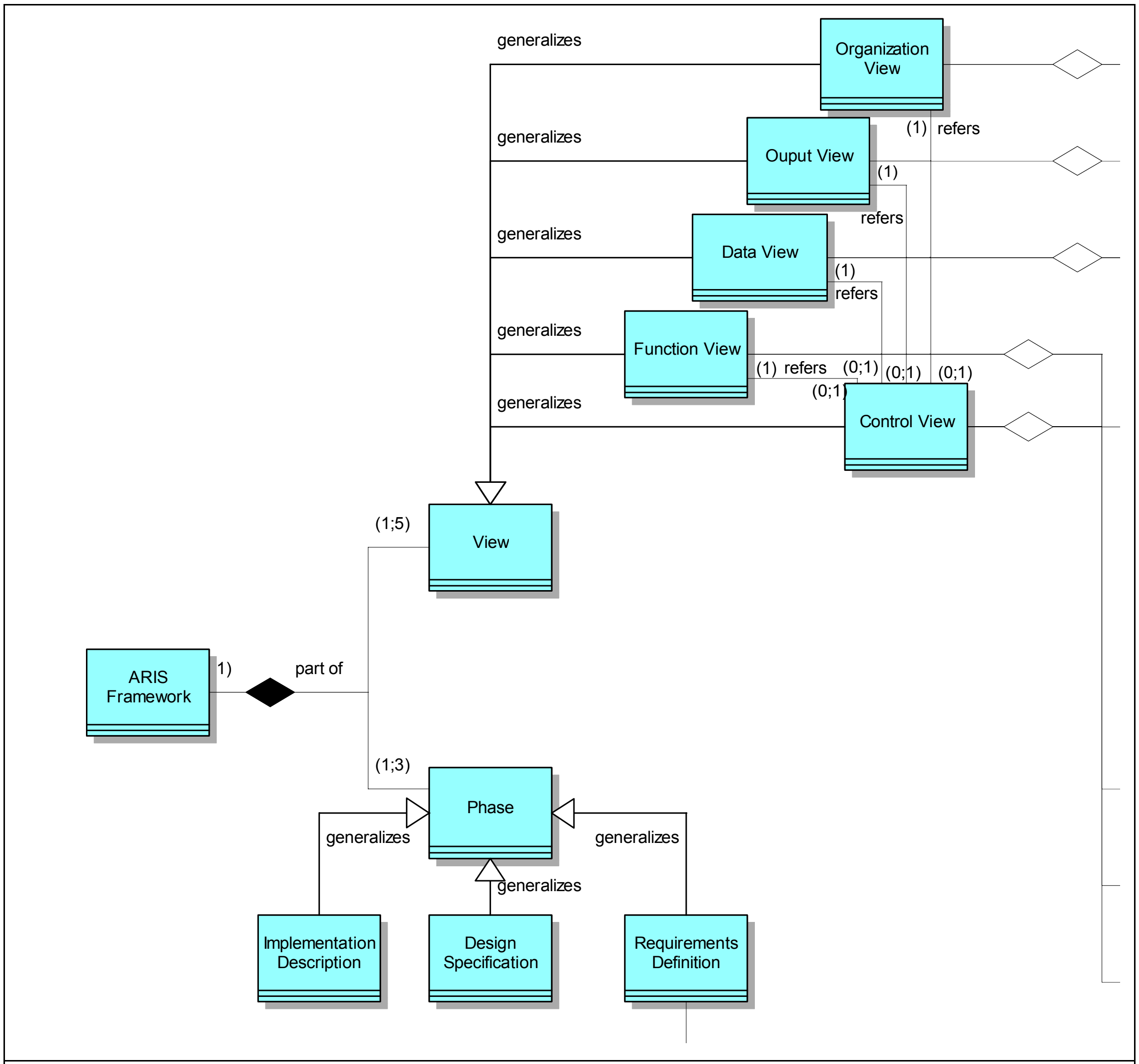

Figure 4. Part of the ARIS Framework Meta Model

As regards the above examination, the integration can be carried out by equalizing the object types "description" and "view" as well as the above-mentioned subtypes.

Despite some formal rules which are described in many of the approaches developed for the schema integration the above comments show that the integration cannot follow a deterministic procedure. Last but not least, this can be accounted for by the fact that the semantics of a model cannot be fully specified.

Integrated Framework Meta Model

Not only the Zachman Framework and ARIS, but also the meta models of the other frameworks contain one same object type which generalizes the different perspectives on the components of the architecture (e.g. processes, data). This object type is referred to as "view" (ARIS), "layer" (BEN), "view" (DoDAF), "modeling level" (SOM), "architecture domain" (TOGAF), or "description" (Zachman). In the following, we will refer to it as the "modeling level". Two of the methods, namely ARIS and Zachman, term the modeling level like the components which they comprise (e.g. for Zachman: "data", "function", "people"). In contrast, the remaining four methods prefer to choose generic designations and delimit domains which comprise several components (e.g. for DoDAF: "operational view", "systems view", "technical view"). An equivalence of the individual modeling levels can be carried out by means of the procedure described in the previous section.

A very important part of the procedure in this case is comparing the components of each modeling level. Sometimes the latters are referred to as core artifacts (BEN) and, mostly, are listed precisely. The modeling levels can be clearly delimited by means of the components which are part of the characteristics of the mentioned object types. This does, however, not exclude that individual components are assigned to several levels. In particular, this is valid for the integration levels which – as in ARIS ("control view") – comprise all components, or – as in BEN ("integration architecture") – only refer to the components of the process and software architectures. This leads to the results as in Table 3.

Table 3. Equivalence of Modeling Levels (framework perspectives)

ARIS	BEN	DoDAF	SOM	TOGAF	Zachman	Integrated Framework
	Business Architecture		Enterprise Plan			Business View
Organizational View, Output View, Function View, Data View		Operational View		Business Architecture	Motivation, People Function, Data Time, Network	
	Process Architecture		Business Process			Organizational View
	Software Architecture	Systems View	Resources	IS-Architecture		System View
-					-	
-	Technology Architecture	Technology View	-	Technology Architecture	-	Technical View
Control view (integrates all views)	Integration (integrates Process and Software Architecture)	All Views (integrates all views)	-	-	-	Integration View

DoDAF and SOM, on the other hand, rather focus the models (especially simple models) which are defined for each modeling level and in which the components are specified. The models are as well itemized as products of the architecture, and are, by some of the methods (especially ARIS, TOGAF, Zachman), only sporadically explained. Thus the architectures offer the user more possibilities to design the specific modeling level.

Moreover, the individual methods recommend a large number of techniques to model the components. Most frequently, the UML techniques are advocated (e.g. DoDAF, TOGAF, Zachman).

A second same object type which, similar to the "modeling level", was also interpreted as a dimension of the framework is only contained in two methods (ARIS, Zachman). The second dimension looks at the components of the architecture by means of the different stages of the software design and will be referred to further down as the "development level". We define the development levels of the integrated framework meta model (see Table 4) according to the ARIS framework "requirements definition" (equivalent to Zachmans "planners view" and "owners view"), "design specification" (equivalent to Zachmans "designers view") and "implementation description" (equivalent to Zachmans "builders view" and "subcontractors view") and according to the Zachman Framework "functioning system" (there is no equivalent within the ARIS framework).

Table 4. Equivalence of Development Levels

ARIS	Zachman	Integrated Framework
Requirements Definition	Planners, Owners view	Requirements Definition
Design Specification	Designers view	Design Specification
Implementation Description	Builders, Subcontractors view	Implementation Description
-	Functioning System	Functioning System

As to the integrated model, only a section is represented. It does not comprise all of the identified components (e.g. components of design specification, components of implementation description, and components of functioning system), architecture models (e.g. process model, data model), and techniques (e.g. UML class diagram) for the development of the diagrams. Quickly it becomes obvious that the assignment of the components to, firstly, the techniques, and, secondly, the architecture models is confusing, even for a limited number of components. Figure 5 represents the integrated framework meta model which consists of the different *modeling levels* (Table 3) and *development levels* (Table 4) as the major parts. A short explanation of the integrated framework meta model is given: the object type *development level* is the generalization of the object types requirements definition, design specification, implementation description, and functioning system. Business view, organizational view, system view, technical view, and integration view are specializations of the object type *modeling level*. The object type *business view,* for example, is related to the object types business goal, business location, organizational unit (all from the Zachman Framework), and the integration view (from the ARIS framework). Figure 5 represents only an incomplete overview of the overall "integrated framework meta model" and gives a first idea of the complexity of the result.

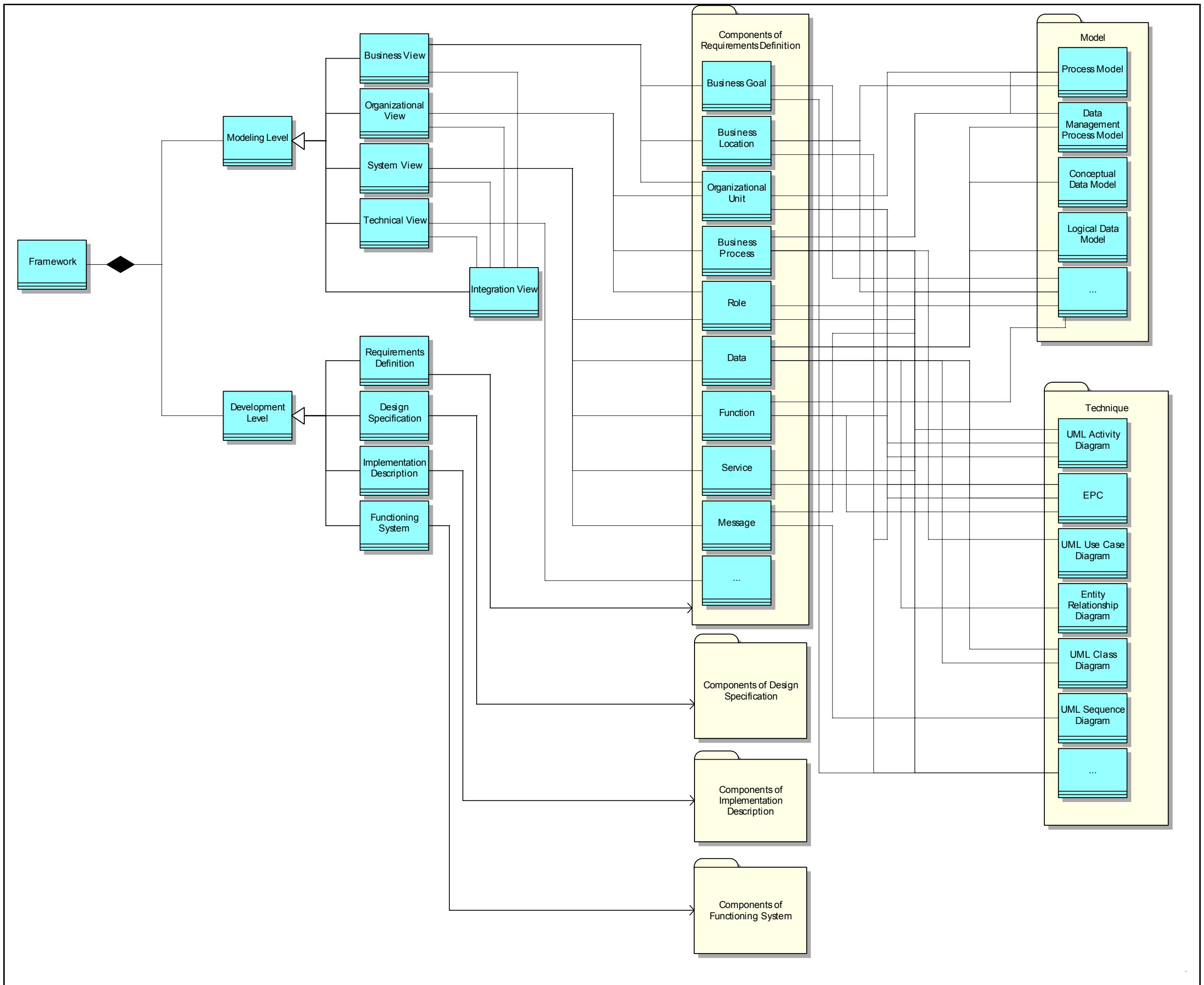

Figure 5. Part of the integrated framework meta model

Conclusion – Evaluation of the Approach and Outlook

This article explains the development of the integrated framework meta model that is a central building block for the architecture method which supports situational adaption. Based on the meta model, several frameworks can be derived in order to meet the situational requirements. The architect has to select the relevant object types of the integrated framework meta model. Existing frameworks based on the integrated framework meta model or based on one of the analyzed architecture methods can easily be adapted to special situational requirements through adding relevant or deleting irrelevant object types of their framework meta model. Valid adoptions of the framework meta models are given through the integrated framework meta model.

On the contrary, a disadvantage of the integrated framework meta model when examining a growing number of methods is the quickly increasing complexity which is difficult to handle. This is particularly disadvantageous for the practical application. Yet, at the same time, the integrated framework meta model offers a basis for a best practice framework. Since, by means of integration, same object types of the meta models were identified, all object types can be derived from the integrated framework meta model which will, for instance, be used in at least three frameworks. The integrated framework meta model can be reduced to only these object types. Therefore, the article provides a first approach to derive a best practice framework and enables the architects to focus common domains.

In this context, the integrated framework is not to be seen exclusively as the only basis for the development and the adaptation of an architecture, but, like in Method Engineering, different types of application can be supported. For instance, the modeling levels of a so-called base method (e.g. TOGAF) can be extended ("configuration") against the background of a specific development or project situation (e.g. TOGAF by means of the integration view). Further on, the assembly of an architecture method from fragments of different methods ("assembling") would be possible. For example, adequate architecture models from DoDAF and ARIS can represent the components identified in BEN in diagrams.

Against the background of constant change and especially changes in strategy, enterprise architecture methods offer a stable regulation framework. It can be adapted according to the situation or strategy and extended at the same time and will be of great importance in future, too. In order to exploit the full amount of flexibility which is offered by Method Engineering, further elements of the architecture methods (especially the procedure model) have to be integrated, and the modules have to be designed in such a way that individual modules can be assembled according to the situation or intended strategy. Moreover, tools have to be developed which reduce the complexity resulting from the integration and help to assemble an appropriate architecture method. At the same time, they should provide interfaces so that the application of the method can be carried out tool-based. And, finally, investigations to validate the results are necessary.

References

Allen, B.R., and Boynton, A.C. "Information Architecture: In Search of Efficient Flexibility," *MIS Quarterly* (15:4) 1991, pp 435-445.

Batini, C., Ceri, S., and Navathe, S.B. *Conceptual Database Design: An Entity-Relationship Approach*, The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1992.

Batini, C., and Lenzerini, M. "A Methodology for Data Schema Integration in the Entity Relationship Model," *IEEE Transaction on Software Engineering* (SE-10:6), November 1984, pp 650-664.

Baumöl, U. "Strategic agility through situational method construction," in *Proceedings of the European Academy of Management (EURAM) Annual Conference*, Munich, 2005.

Braun, C., and Winter, R. "A Comprehensive Enterprise Architecture Metamodel and Its Implementation Using a Metamodeling Platform," in *Proceedings of the GI Workshop on Enterprise Modelling and Information System Architectures*, Klagenfurt, 2005.

Brinkkemper, S. "Method engineering: engineering of information systems development methods and tools," *Information and Software Technology* (38:4) 1996, pp 275-280.

Brinkkemper, S. "Method Engineering with Web-enabled Methods," in *Information Systems Engineering: State of the Art and Research Themes,* S. Brinkkemper, E. Lindencrona and A. Solvberg (Eds.), Springer, London, 2000, pp. 123-133.

Ciborra, C.U., Braa, K., Cordella, A., Dahlbom, B., Failla, A., and Hanseth, O. *From Control to Drift: The Dynamics of Corporate Information Infrastructures*, Oxford University Press, 2000.

Dickson, G.W., and Wetherbe, J.C. *The Management of Information Systems*, NewYork, 1985.

DoDAF "DoD Architecture Framework Version 1.5 - Volume I: Definitions and Guidelines." Retrieved 29.3.2008, from http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_I.pdf, 2007a.

DoDAF "DoD Architecture Framework Version 1.5 - Volume II: Product Descriptions." Retrieved 29.3.2008, from http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_II.pdf, 2007b.

Fantoni, G., Taviani, C., and Santoro, R. "Design by functional synonyms and antonyms: a structured creative technique based on functional analysis," in *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, Professional Engineering Publishing, 2007, pp. 673-683.

Ferstl, O.K., and Sinz, E.J. *Grundlagen der Wirtschaftsinformatik*, (4. ed.), Oldenburg, München/Wien, 2001.

Ferstl, O.K., and Sinz, E.J. "Modeling of Business Systems Using SOM," in *Handbook on Architectures of Information Systems,* P. Bernus, K. Mertins and G. Schmidt (Eds.), Springer, Berlin et al., 2006, pp. 347-367.

Goethals, F. "An Overview of Enterprise Architecture Framework Deliverables." Retrieved 7.8.2008, from http://www.econ.kuleuven.be/leerstoel/sap/downloads/Goethals%20Overview%20existing%20frameworks.pdf, 2003.

Goethals, F., Vandenbulcke, J., and Lemahieu, W. "Developing the Extended Enterprise with the FADEE," in *Proceedings of the 2004 ACM symposium on Applied computing, SESSION: Organizational engineering (OE)*, ACM Press, New York, NY, Nicosia, Cyprus, 2004, pp. 1372-1379.

Goikoetxea, A. *Enterprise Architectures and Digital Administration - Planning, Design and Assessment*, World Scientific, New Jersey et al., 2007.

Gutzwiller, T.A. *Das CC RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen*, Physica-Verlag, Heidelberg, 1994.

Henderson, J.C., and Venkatraman, N. "Strategic alignment: Leveraging information technology for transforming organizations," *IBM Systems Journal (Reprint Vol. 32, No. 1, 1993)* (38:2&3) 1999, pp 472-484.

Hevner, A.R., March, S.T., Park, J., and Ram, S. "Design Science in Information Systems Research," *MIS Quarterly* (28:1) 2004, pp 75-105.

Heym, M. *Methoden-Engineering*, Rosch-Buchbinderei, Hallstadt, 1993.

IEEE *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, Software Engineering Standards Committee of the IEEE Computer Society, The Institute of Electrical and Electronics Engineers Inc., New York, 2000.

Karlsson, F., Ågerfalk, P.J. "Method Configuration: Adapting to Situational Characteristics While Creating Reusable Assets," *Journal of Information and Software Technology* (46:9) 2004, pp 619-633.

Larson, J.A., Navathe, S.B., and Elmasri, R. "A Theory of Attribute Equivalence in Database with Application to Schema Integration," *IEEE Transaction on Software Engineering* (15:4), April 1989, pp 449-463.

Leist, S., and Zellner, G. "Evaluation of Current Architecture Frameworks," in *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC)*, Bourgogne University, Dijon, France, 2006, pp. 1546-1553.

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

March, S.T., and Smith, G.F. "Design and natural science research on information technology," *Decision Support Systems* (15:4) 1995, pp 251-266.

Niemann, K.D. *From Enterprise Architecture to IT Governance - Elements of Effective IT Management*, Vieweg, Wiesbaden, 2006.

Pearlson, K.E., and Saunders, C.S. *Managing and Using Informaton Systems - A Strategic Approach*, (Third ed.), Wiley, Hoboken, NJ, 2006.

Peffers, K., Tuunanen, T., Gengler, C., Rossi, M., Hui, W., Virtanen, V., and Bragge, J. "The Design Science Research Process: A Model for Producing and Presenting Information System Research," in *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology*, Claremont, Carlifornia, 2006.

Pereira, C.M., and Sousa, P. "A Methode to Define an Enterprise Architecture using the Zachman Framework," in *Proceedings of the 2004 ACM Symposium on Applied Computing (SAC)*, ACM Press New York NY, Nicosia Cyprus, 2004, pp. 1366-1371.

Scheer, A.-W. *ARIS - Modellierungsmethoden, Metamodelle, Anwendungen*, (4. ed.), Springer, Berlin et al., 2001.

Scheer, A.-W., and Schneider, K. "ARIS - Architecture of Integrated Information Systems," in *Handbook on Architectures of Information Systems,* P. Bernus, K. Mertins and G. Schmidt (Eds.), Springer, Berlin et al., 2006, pp. 605-623.

Schekkerman, J. *How to survive in the jungle of Enterprise Architecture Frameworks - Creating or choosing an Enterprise Architecture Framework*, Trafford, Canada, 2004.

Smolander, K., Lyytinen, K., Tahvanainen, V.-P., and Marttiin, P. "MetaEdit: a flexible graphical environment for methodology modelling," in *Proceedings of the CAiSE '91: Proceedings of the third international conference on Advanced information Systems Engineering*, Springer-Verlag New York, Trondheim, Norway, 1991, pp. 168-193.

Sowa, F.J., and Zachman, J.A. "Extending and formalizing the framework for information systems architecture," *IBM Systems Journal* (31:3) 1992, pp 590-616.

Takeda, H., Veerkamp, P., Tomiyama, T., and Yoshikawam, H. "Modeling Design Processes," *AI Magazine* (11:4) 1990, pp 37-48.

The CIO Council "Federal Enterprise Architecture Framework Version 1.1." Retrieved 13.8.2008, from http://www.cio.gov/Documents/fedarch1.pdf, 1999.

The Open Group "Other Architectures and Frameworks." Retrieved 29.3.2008, from http://www.opengroup.org/architecture/togaf8-doc/arch/chap37.html#tag_38, 2006a.

The Open Group "TOGAF (The Open Group Architecture Framework) Version 8.1.1 "Enterprise Edition"." Retrieved 26.3.2008, from http://www.opengroup.org/architecture/togaf8-doc/arch/toc.html, 2006b.

Tolvanen, J.-P. "Method engineering: current research directions and implications for future research," in *Proceedings of the Method Engineering - Principles of Method Construction and Tool Support, Proceedings of the IFIP TC8 WG8.1/8.2 Working Conference on Method Engineering*, Chapmann Hall, Atlanta, USA, 1996, pp. 296-317.

Tolvanen, J.-P., Rossi, M., and Liu, H. "Method engineering: current research directions and implications for future research," in *Proceedings of the Method Engineering - Principles of Method Construction and Tool Support, Proceedings of the IFIP TC8 WG8.1/8.2 Working Conference on Method Engineering*, Chapman Hall, Atlanta, USA, 1996, pp. 296-317.

Vaishnavi, V., and Kuechler, W. "Design Research in Information Systems," 2004.

Whitman, L., Ramachandran, K., and Ketkar, V. "A taxonomy of a living model of the enterprise," in *Proceedings of the Winter Simulation Conference (WSC)*, ACM, Arlington, VA, USA, 2001.

Winter, R., and Fischer, R. "Essential Layers, Artifacts, and Dependencies of Enterprise Architecture," *Journal of Enterprise Architecture* (3:2) 2007, pp 7-18.

Zachman, J.A. "A framework for information systems architecture," *IBM Systems Journal* (26:3) 1987, pp 276-295.

Zivkovic, S., Kühn, H., and Karagiannis, D. "Facilitate Modelling Using Method Integration - An Approach Using Mapping and Integration Rules," in *Proceedings of the 15th European Conference on Information Systems (ECIS) "Relevant rigor - Rigorous relevance"*, St. Gallen, Switzerland, 2007, pp. 2038-2050.