Stephan Aier, Bettina Gleichauf

Application of Enterprise Models for Engineering Enterprise Transformation

Enterprise models constitute a valuable basis for enterprise transformation because they usually represent a widely accepted image of an enterprise. Practitioners often put a lot of effort in the creation and maintenance of such models that therefore represent a significant investment. However, so far the information contained in enterprise models is to a large extend 'dateless' which means it is hardly used to describe the transformation itself consistently. Therefore we propose a method to systematically derive an enterprise transformation model based on existing models representing enterprise structures at different points in time. The result of the method application is a set of project outlines derived from enterprise models. In order to generalise our approach to a multiperiod transformation model capable of coping with dynamic changes and plan deviations we propose a respective conceptual system. Our research artefact (the method) is finally demonstrated in a case study.

1 Introduction

The need for enterprises to constantly adapt to ever changing requirements of the environment is a continuing field of business and information systems engineering (BISE) research. The notion of engineering focuses on applying a systematic approach to enterprise transformation. The transformation of enterprises is engineered by the means of appropriate models and methods.

Enterprise modelling (EM) was early identified to be a core contribution to BISE (Fox and Gruninger 1998; Katz 1990) since it provides adequate means for the description of as-is and / or to-be states of enterprises. EM addresses the modelling of business processes, goals, strategy, information entities, business structure, support systems, skills, and people of an enterprise (Fox and Gruninger 1998). Thus, enterprise models integrate conceptual models of information systems as well as models of supported business functions and can provide the necessary transparency for a systematic support to enterprise engineering (Dietz 2006). By representing both organisational and technical structures, EM may also provide a broad and consolidated view of an entire corporation or government agency (Kirikova 2000), also known as enterprise architecture (EA) (Jonkers et al. 2006; Lankhorst 2005; Rohloff 2008; Tyler and Cathcart 2006; Winter et al. 2007).

However, so far EM and EA modelling have been restricted to modelling of (dateless) states of an enterprise by the majority of contributions (Aier et al. 2009b; Buckl et al. 2009). EM did not address the modelling of transformation and thus of the engineering process itself.

As we illustrate in the next two sections there is an actual need for model based enterprise engineering. Few contributions give first structure to the challenges of EM for transformation and engineering purposes (e.g., Aier et al. 2009b; Buckl et al. 2009). We claim that enterprise models can constitute the basis for engineering enterprise transformation by providing models of different points in time and an appropriate analysis upon them. Referring to the analyses mentioned above this paper aims at answering the following research question:

How can enterprise engineering and thus transformation tasks be derived from an analysis of EM snapshots of different points in time?

To answer this question the core contribution of this paper is a method that guides the engineering of enterprise transformation.

However, the paper does not consider the planning of future states of an enterprise by the means of models but focuses on the transformation of current states to planned states. Therefore it is presumed that at least a to-be model exists and that a preliminary decision about the planned implementation / transformation has already been made. Furthermore the representation of necessary temporal aspects is not in the focus of the paper.

The paper follows a design research (DR) process as described by Peffers et al. (2007, 89ff.): We identify the problem and define the objectives of our solution by analysing current practitioners' solutions (section 2) and by analysing related work (section 3). In section 4 we construct our DR artefact as a method for deriving transformation activities from enterprise model snapshots of different points in time. The method construction focuses on techniques, a process model, and design results. In section 5 we demonstrate our solution by presenting an industry case study reflecting our method application. Section 6 finally discusses the evaluation results, research limitations as well as further implications. The article closes with a conclusion in section 7.

2 Case Studies

The following case studies exemplify how the planning of transformation can be supported by enterprise models. They also show which requirements arise from current approaches and thus motivate the need for a method to systematically plan and engineer transformation.

2.1 Company A

Company A provides IT outsourcing services and banking solutions. The primary product is an integrated banking platform that is offered to private and universal banks. The company focuses on three main fields, namely application development, application management, and operations, and therefore offers an integrated portfolio to its customers. The application development division is responsible for the development of the integrated banking platform that is described by an application architecture model, an IT architecture model, and an operating model.

Further development activities are planned and controlled by the architecture team which designs and integrates development roadmaps for individual applications and establishes an architectural development plan that fits the banking platform's high level release plan. From there, appropriate development projects are identified and realised.

Major challenges within the architectural development plan are the coordination of the activities of the development teams and assurance that all dependencies are addressed and that milestones of the various integration and development activities are met simultaneously. If, for example, a component of an application needs an interface to a component of another application at a certain time for a certain milestone (e.g., test or release), it has to be assured that both components are available at that very point in time. This simple example grows very complex as the banking platform comprises over 200 applications, each consisting of a multitude of components that each have their own lifecycles as well as precursor and successor relationships.

The following questions are crucial to the architectural development process in company A:

- How can the necessary changes for achieving the desired to-be state be identified?
- How can transformation be decomposed into project activities?

- How can interdependencies between project activities be identified?
- How can the necessary development activities be bundled in order to be integrated into one release?

2.2 Company B

Company B is an internationally operating bank based in Switzerland. During recent decades, mergers led to an increasing complexity of its application landscape. Regarding architecture layers, business architecture, application and integration architecture, software and component architecture, and technical architecture are distinguished. Architecture management is carried out by more than 90 architects and comprises architecture governance, which is enforced in individual information system development projects. However, while IT architecture is strong in the bank's home country, the bank has to face challenges due to heterogeneous local solutions in almost every other country.

In order to enable a better management of the heterogeneous application landscape, an EA project is currently being conducted. The project focuses on an integrated view on the different solutions the IT departments offer to the company's operating departments and teams worldwide. Solutions, in this case, denote bundles of product components configured for a certain application scenario. The intended integrated view should also enable solution roadmap planning, i.e., the continuous development of the contained components. Therefore, the following questions need to be answered:

- Which projects should be shifted back or forward in order to meet the needs of a certain solution roadmap?
- Which projects affect which lifecycle planning of a certain solution?
- Does postponing of a project affect the lifecycle planning of a certain solution?

An EA approach aiming at these requirements must be capable of consolidating information of different projects affecting solution development, e.g., release planning, component development, and customer request management for customised solutions. This approach requires the inclusion of dynamic aspects such as solution and component lifecycles, but especially the support of multi project management. Generally speaking, there is a strong need for support of the transformation process from as-is (application) architecture to to-be (application) architecture.

2.3 Implications for method construction

Although the two case studies reveal different wordings, the central questions aim at similar challenges which need to be encountered. This leads to general requirements concerning the planning of enterprise transformation. Hence, a comprehensive method that supports enterprise transformation must respect the following requirements:

- Enterprise or Enterprise Architecture models represent the changes of enterprise artefacts required in the real world. Therefore, at least (temporal) successor relationships and lifecycles need to be captured.
- Actual project activities can be derived from the models.
- Enterprise or EA models reflect the interdependencies between the elements, i.e., the enterprise artefacts, as well as between their lifecycles.

3 Related Work

In this section we review related work from the field of EM, especially EA modelling, enterprise engineering as well as from the field of enterprise transformation since these fields of research are obviously related to our research question. We also analysed more general areas of research dealing with holism and dynamics such as systems theory or cybernetic. Especially systems theory provides a valuable basis to construct a sound terminology for systems, its components and their relations as a whole instead of a collection of components (e.g., Ashby 1956, or more specifically Hoogervorst 2009; Simon 1969). However, it hardly contributes specifically to the method construction we are aiming at here.

Up to now only a few approaches for engineering enterprise transformation based on and reflected in enterprise models exist. While there are various contributions dealing with related questions, none of the existing approaches addresses enterprise planning from business to IT covering artefact relationships in semi-formal models and / or addressing model dynamics.

Historically, the topic evolved from strategic IS planning which was firstly addressed in an MISQ contribution by King in 1978 (King 1978). This paper proposes a process to design a management information system (MIS) in accordance to the strategy of a corporation or government agency and thereby define a MIS strategy comprising MIS objectives and MIS constraints. As markets, organisational structures and system landscapes added more complexity to the matter of strategic planning and the alignment of business and IT, this approach as well as similar contributions were evolutionary refined. Strategic enterprise-wide information management (Targowski 1990) and more institutionalised IS planning processes became an issue in the 1990ies (Eliot 1991). A prominent example for IS planning methods is IBM's Business System Planning (BSP) (IBM 1984). BSP aims to (re-)group IT functionalities according to data use and thereby identify application candidates with high internal integration intensity, but limited external interfacing to other applications.

Especially the field of enterprise architecture (EA) is concerned with aggregated models covering a broad scope from business to IT. Therefore also in EA related approaches for planning and transformation were developed, e.g., by Spewak and Hill (1993); Spewak and Tiemann (2006) (the wedding cake model), Pulkkinen (2006); Pulkkinen and Hirvonen (2005), Land et al. (2009) and Niemann (2006). However, the majority of research results only focus on a unidirectional planning process that aims at improving the current structure, i.e., establishing a to-be architecture. The process of transforming the current architecture into the target architecture is only considered negligibly.

Recently the works of Buckl et al. (2008, 2009) and, Aier et al. (2009b) address a comprehensive modelling approach for planning purposes in an EA context. While Buckl et al. (2009) propose a set of meta model requirements for modelling temporal aspects, their proposal focuses on application landscapes. However, they take into account important temporal dimensions, e.g., the time a model is created and the time a model should be valid for the past, the present and the future, as well as different variants of future models (Buckl et al. 2009). Recommendations or a framework addressing the transformation process itself cannot be found in these approaches though.

While enterprise engineering has a long tradition and traditionally has a strong focus on enterprise modelling (Albani et al. 2009; Bernus and Nemes 1996; Kosanke et al. 1999; Liles and Presley 1996), there are only very few contributions dealing with modelling enterprise transformation. Among those few are Sousa et al. (2009) who take the dynamic evolution of enterprise models into account, providing a formal approach for generating blueprints of enterprise architecture from existing project management sources. By applying this bottom-up approach, the current and the planned state of the enterprise architecture as well as its evolution can be visualised. In doing so, this approach reconstructs EA models from existing project plans rather than considering the derivation of projects from model information.

Aier et al. (2009b) identify different complexity levels for representing dynamics in EA planning.

On the highest of their seven complexity levels, a comprehensive planning and transformation approach has to comply with the following requirements:

- Model as-is state
- Model to-be states
- Model transformation paths from as-is to to-be states
- Model alternatives for models and paths for a respective point in time
- Model an unlimited number of points in time as well as the respective transformation paths
- Model deviations from plans

These requirements set an environment in which transformation between as-is and to-be states can be planned. However, the work of Aier et al. does not elaborate on how transformation paths can be derived.

Finally there are a number of contributions dealing with enterprise transformation. Although these contributions also confirm the important role of models they mostly focus on the explanation of reasons for and directions of enterprise transformation rather than describing the HOW of enterprise transformation (e.g., Rouse 2008).

4 Method Construction

Methods are a widely used artefact for BISE purposes. The research field of method engineering addresses the construction of sound methods, techniques, and constructs (Brinkkemper 1996, p. 276). Therefore section 4.1 briefly introduces the concept of a method as it is used in this article. Before we present the construction of the method itself we outline the research framework for engineering enterprise transformation in section 4.2, introducing the main concepts of the different stages on which transformation tasks take place. The method which is focused on planning the actual transformation by the means of models is described in section 4.3. Subsequently, in section 4.4, we enhance our method to dynamic aspects that have to be considered by the method in order to fully respect the requirements derived by the case studies in section 2.

4.1 Method engineering

Design research aims at the design of problem-oriented artefacts like constructs, models, instantiations, and methods (Hevner et al. 2004, p. 77; March and Smith 1995, p. 253). A method is considered as a systematic and goal-oriented procedure to solve certain problems (Brinkkemper 1996, p. 275; Braun et al. 2005). Thus, methods provide the basis for effective and efficient procedures while engineering enterprise components like business processes, information systems, or IT infrastructure in a systemic way.

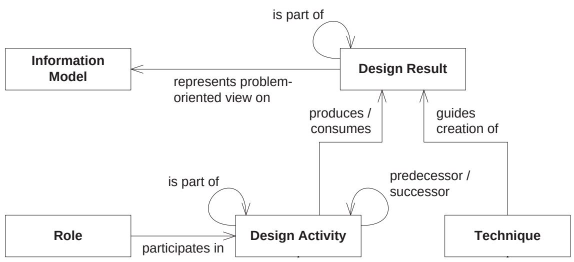

Figure 1: Meta model of a method (based on Gutzwiller 1994, p. 13)

A method consists of a sequence of design activities which are related to a procedure model. The produced design results of the design activities are represented by an information model. Additionally, a method consists of roles which describe the participants involved in the design activities. At the same time the inclusion of roles introduces various perspectives different stakeholders may have on design activities. Instructions on how the design results are produced and documented are provided by adequate techniques (Braun et al. 2005; Gutzwiller 1994, p. 13).

In this contribution we focus on design results, corresponding design activities as well as supporting techniques as the main components of a method. Therefore we start to identify and explicate functional and non-functional requirements of the method in order to specify the desired design results as the 'product' of the method. In doing so, we also elaborate on the interdependencies between design results. Subsequently, we derive design activities and techniques that are necessary to create the design results.

4.2 Research framework

In this paper we consider the engineering of enterprise transformation using enterprise models. Analysing models of as-is and desired to-be states can serve as a basis for conducting the transformation in reality in an engineered way. Therefore we distinguish different states of the concrete system (reality) and the conceptual system (model) (Dietz 2006, p. 64). Figure 2 illustrates the model and reality level as well as the different relationships between both (conceptualisation and implementation). When only these two states of a system are known, the transformation from one state into the other represents a black-box model (Dietz 2006, p. 67). However, the aim of this research is to describe this transformation as a white-box model. Therefore we differentiate three main parts of the transformation process, labelled TR1, TR2 and TR3.

During the transformation process the tasks of designing, analysing and engineering are used for specific purposes: The representation of the as-is state is achieved by an as-is model while the desired to-be state is specified by a to-be model. In terms of classical architecture the to-be model corresponds to a high level specification designed by the architect addressing the client's requirements. Based on this blueprint, the actual engineering can be initiated during which a transformation plan and subsequently a to-be state are established (Pavlak 2006). The engineering task needs to include more detail than the preliminary tasks of designing the to-be state. Therefore, a thorough planning of the transformation is necessary.

On the model level an analysis of the two models can identify the changes that are necessary to realise the specified to-be state coming from the as-is state (TR1). As a second step the transformation of reality can be planned in detail. The identified changes can be further analysed with regard to their interdependencies (TR2). As a result, a reasonable sequence of development activities can be derived. This serves as an input to the definition of activity bundles that can be turned into project definitions.

During TR1 as well as during TR2 findings of the analyses may indicate that there is no solution adhering to the requirements of the preceding step (design of to-be model respectively TR1). Also further knowledge about the possible future states of the as-is architecture may arise and thus require adjustments of TR1 or the design of the to-be model (cf. arrows labelled 'Feedback' in Figure 2). The same can occur while conducting the project and in program management (TR3): If, for example, concurrent use of resources is detected, this information needs to be propagated back, too.

Step TR3 includes the realisation of the transformation in reality, i.e., the changing of affected elements as well as the realisation of new structures where applicable. The transformation is most likely carried out by projects, based on the specifications made in TR2. Temporal or resource constraints and changing general conditions occurring during TR3 must also be controlled and monitored by TR2. The arrows in Figure 2 indicate the iterative control cycles performed during TR3.

4.3 A method for planning enterprise transformation

As described in section 4.2 the planning of the enterprise transformation can be divided in three parts. This division is also used for the method which is developed below. The method focuses on the use of EA models for planning purposes and therefore does not consider the project management tasks regarding costs, quality, human resources, communications, risks and procurement

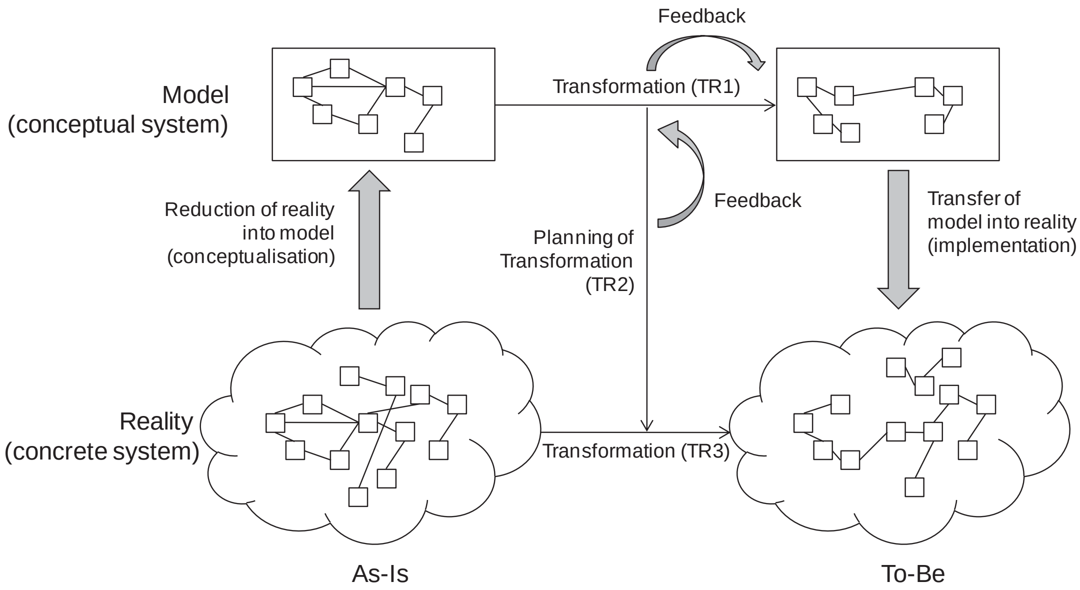

Figure 2: Different levels of transformation

(as identified by Project Management Institute 2000), which are subject to step TR3. Although relative project time dependencies are addressed because the main activities as well as resulting time constraints are derived from the enterprise models. Table 1 summarises the main concepts of the method.

The result of step TR1 is a list of model elements that need to be changed in order to reach the targeted to-be state (design result R1). Those model elements constitute the core subjects of the transformation activities. Additionally, as the model elements must not be considered as stand-alone artefacts, information about their relationships to the other model elements need to be captured as well (R2). The method must also produce a list of predecessor and successor relationships between the as-is and the to-be model because this is the basis for planning development activities (R3). The representation of the model elements and their relationships can be done by a transformation model, like Cîmpan et al. (2007) also propose to encapsulate transformation information in a separate model.

Step TR2 plans the transformation in detail. Therefore, the method's design result is a description of project proposals containing a list of preconditions, a list of the affected model elements, and a relative timeline that respects the individual lifecycles of as well as the temporal interdependencies between the model elements (R4). Based on these proposals actual projects can be defined in step TR3, taking current budgets and resources into account. Of course, the building of the project proposals depends on the results from step TR1 inasmuch as the interdependencies between the model elements within as well

Table 1: Method summary

	Method for planning enterprise transformation
Design Results	R1. List of model elements that need to be changed R2. Information about relationships to other model elements R3. List of predecessor and successor relationships R4. Description of project proposals containing a list of preconditions, a list of the affected model elements and a relative timeline
Information Model	I1. Transformation Model (for R1, R2, R3)
Design Activities	A1. Analyse differences between as-is and to-be A2. Separate segments, i.e. identify projects For each segment: A3. Identify general temporal interdependencies A4. Schedule an effective sequence of development activities, by refining temporal interdependencies A5. Draw network plan displaying all interdependencies
Techniques	T1. Graph comparison (for A1) T2. Establishing project network model/graph using Precedence Diagramming Method (PDM) and the Arrow Diagramming Method (ADM) (for A5) T3. Deriving a project timeline using Critical Path Method (CPM) (for A5)

as between the as-is and to-be models must be considered.

In order to produce the list of affected model elements and their interrelationships, the first design activity (A1) compares the graphs of the as-is and the to-be model. By this analysis of differences six possible relationships can be found. An example is shown in Figure 3 that depicts an as-is model labelled M0, a to-be model labelled M1, and the successor relationships between the model elements. The box labelled TM represents the transformation model in which the list of differences is captured. The central question is: Which elements in model M1 are successors of elements in model M0? We also assume that the elements on both sides of a successor relationship are of the same type or that they are refinements or aggregates of each other's type. For example, an application in model M0 can be subdivided into multiple services in model M1. In this case, a fine-grained meta model would identify two different element types, having a clear refinement relationship. However, a more coarse-grained meta model would possibly consider these artefacts as instantiations of the same meta model element, discriminated by different attributes only.

1. **Relationship 1:1**

One element in model M0 has exactly one successor in model M1. The example in Figure 3 shows element 11 in model M1 as a successor of element 4 in model M0. Element 1 represents a special case of explicit successorship, though: Here, the element in model M0 is identical to the element in model M1. This is emphasised in the picture by an identical label.

2. **Relationship 1:n**

One element in model M0 has more than one successor in model M1. This might be the case if a component is decomposed in the future model. In the example element 2 in model M0 is decomposed into elements 9 and 10.

3. **Relationship n:m**

Several elements in model M0 have multiple successors in model M1. This reflects complex restructuring of elements in the future model. In the example, elements 7 and 8 in model M0 are rearranged in elements 14 and 15 in model M1. In this case, the successor relationships cannot be expressed in 1:n or n:1 relationships, because different parts of the elements 7 and 8 might be

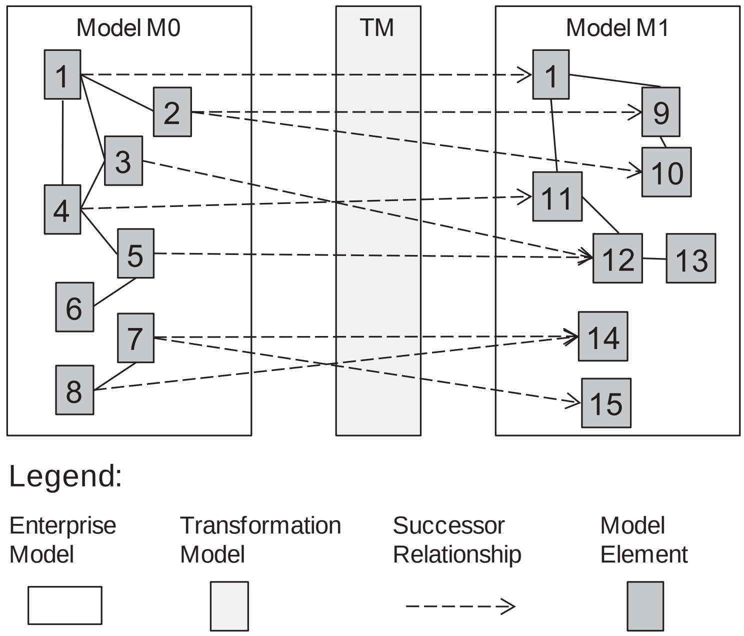

Figure 3: Analysis of differences

rearranged in both of the new elements 14 and 15.

4. Relationship n:1

Several elements in model M0 have exactly one successor in model M1. This is the case if an aggregation of elements is planned in the future model. The example shows elements 3 and 5 which are aggregated into element 12 in model M1.

In addition to cases 1 to 4, two special cases can be identified:

5. Relationship 1:0

One element in model M0 has no successor in model M1. This reflects a termination of an element in the future model like it is exemplified by the element 6 in Figure 3.

6. Relationship 0:1

One element in model M1 has no predecessor in model M0. This represents a new element in model M1. In the example, element 13, that does not replace an existing element in model M0, is added to element 12 in model M1.

Based on the relationships within the to-be model, general segments of model elements can be derived (A2). Such segments constitute the scope a project proposal (R4) will comprise. The following activities will then be executed for each segment. Referring to the example in Figure 2, the elements 7, 8, 14 and 15 can be grouped into one segment as they do not have any interdependencies with the other elements.

The analysis of differences (A1) identifies successor relationships between model elements which represent successor relationships of enterprise artefacts, more precisely of their lifecycle phase 'in production'. From there, the general dependencies of pre-production and decommissioning lifecycle phases can be derived for each segment (A3) (cf. Figure 4). The actual sub-division of such pre-production phases vary depending on the type of artefact considered and may comprise various development activities, including specification, testing etc.

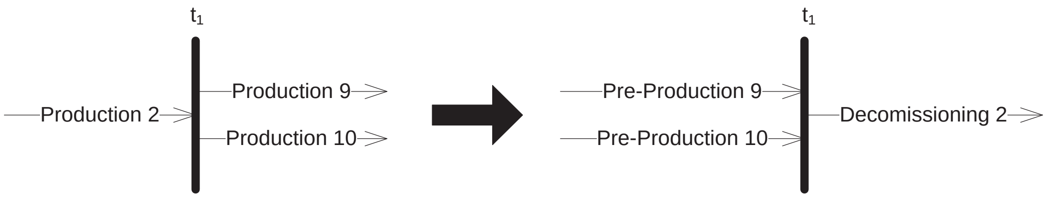

Figure 4: General temporal interdependencies of lifecycle phases

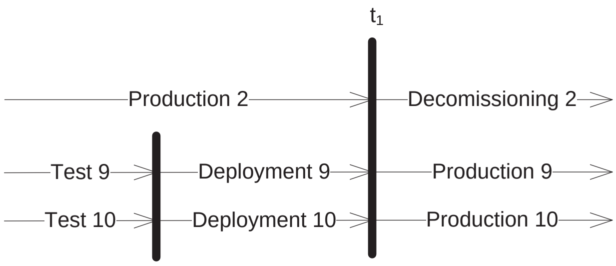

Figure 5: Refined temporal interdependencies of lifecycle phases

The next design activities aim at scheduling an effective sequence of development activities upon the affected artefacts (A4). Therefore, the inter-model relationships between the model elements must additionally be taken into account. In our example, elements 9 and 10 depend on each other, which might require an additional synchronisation point between the respective pre-production phases, e.g., in the test phase (cf. Figure 5). Another temporal restriction may arise if a new business process is to be introduced in the target state and the supporting applications must be developed before the business process can be activated. Additionally, individual model element lifecycles that might be affected by vendor support or the like must be respected and integrated.

Design activity A4 might result in several alternative sequences, so there is not necessarily a unique solution for a model specifying the development steps between two enterprise models. Moreover, no valid solution might be found at all, if defects in the models are detected that require an iteration of the process (cf. section 4.4 on dynamic planning).

Scheduling the development activities and establishing the transformation model can be supported by planning techniques taken from the related field of project time management (A5). Gantt charts or milestone charts are commonly used to represent schedules. However, these techniques lack in showing interdependencies between activities and events, which is provided by network techniques (Kerzner 2009, p. 496). Among such techniques to construct project network diagrams, the Project Management Institute mentions the Precedence Diagramming Method (PDM) and the Arrow Diagramming Method (ADM) that are able to display temporal dependencies between the individual activities (Project Management Institute 2000, 69 ff.). PDM and ADM differ in their representation of activities on nodes (PDM) or activities on arrows (ADM).

Based upon such project network models, mathematical analyses can be applied in order to determine actual dates for the project plan. The most common technique is the Critical Path Method (CPM) that calculates deterministic early and late start and finish dates for each activity, based on an activity-on-node network (Antill and Woodhead 1990, 8 ff.). The CPM technique is comprised of the following steps, applied to the use of enterprise models:

1. Describe and list unique feasible (model element change) activities

2. Specify the constraints between the activities, i.e., the inner-model and temporal interdependencies
3. Establish a network diagram, representing the activities and their interdependencies
4. Assign time (and cost) information for each activity on each arrow
5. Deriving the earliest possible start and latest possible finish dates for each activity

4.4 Conceptual system for multi-period dynamic planning

So far we have covered the planning process for two points in time only. However, if put into a more general framework of time (as we do in Figure 6), this process can be applied in every stage. However, during the analysis of model differences and the planning of the development steps, dynamic aspects need to be considered (Aier et al. 2009b): If external conditions for the realisation of the to-be model change, the planning of the transformation need to be adjusted respectively. Adjustments of the to-be model will then require adjustments or re-processing of the method activities.

In addition, during the process of the method described above, new information can arise that also trigger the adjustment of a to-be model. For example, findings about inconsistencies between the as-is and the to-be model during the analysis phase (TR1) will trigger a re-modelling of the to-be model. In order to enable dynamic planning of enterprise transformation such feedback flows will also have to be integrated on the described method.

The prerequisites for enabling dynamic transformation must be set within the underlying enterprise models. They must support modelling of different points in time and reflect changes in models over time. Our proposed framework for representing dynamic transformation in enterprise models are explained below (cf. Figure 6). This approach focuses on whole models, e.g., an application landscape model.

The main concept is the distinction of modelling time and valid time, based on the concept of bitemporal tables by Snodgrass (2000). In this respect, modelling time represents the time a certain enterprise model is created or updated. Hence, the modelling time dimension is used to capture versioning of enterprise models. On the other hand, valid time expresses the time a certain model is valid for. Figure 6 depicts the valid time on the horizontal axis and the modelling time on the vertical axis. While valid time is only sectioned in years, modelling times are shown in irregular intervals. A different granularity of both time dimensions reflects realistic conditions: While valid time intervals can be interpreted as release planning cycles, modelling time intervals might be subject to unplanned impacts and therefore occur irregularly. Such an impact is shown exemplary in Figure 6 for May 2009.

Following the explanations of Snodgrass, valid time and modelling time can be recorded by timestamps (Snodgrass 2000). For reasons of clarity, only the valid from date of each model is depicted in Figure 6. Although the information on the sequence of the different planned models is immanent in the valid time, the models are additionally named M0, M1 and so on in order to visualise their sequence. The sequence is valid for a certain point in modelling time, visualised as a perspective following the term used by Arnoldi et al. (2008).

Arrows between two successive models depict the transformation paths. As described above, the transformation between enterprise models can be represented in transformation models. Transformation models are depicted as black boxes in Figure 6.

With proceeding time, planned models become current models (visualised by dark grey boxes). At this point, we abstain from labelling the models as as-is or to-be models because this information is more formally expressed by the validity

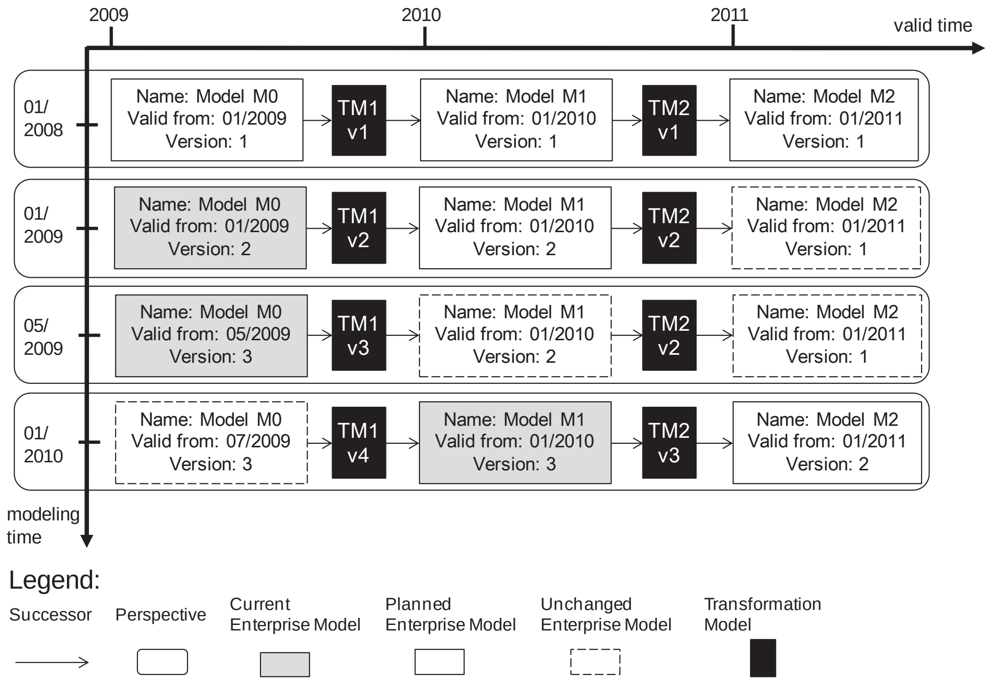

Figure 6: Representing transformation in enterprise models

timestamps in conjunction with actual time. Finally, new versions of the enterprise models are developed, which is visualised in Figure 6 along the vertical axis. In doing so, not necessarily all models are changed but some are simply taken over from the last point in modelling time (visualised by a dashed line). As models change, the transformation path between them and therefore the transformation model also changes. Regarding the transformation model's content, this means that the current state of planning the transformation also changes over time. Figure 6 depicts this by displaying versions of the transformation models.

5 Method Application

The described method has been developed in conjunction with practical experiences at company A that has already been presented in section 2. In doing so, experiences from existing best practices as well as ideas from related research fields have been combined. At the same time, the actual application of the method at the company served as an evaluation. This approach ensures the efficacy and utility of the constructed method, as it is required for a design research process (Peffers et al. 2007).

The planning activities at the company aim at the further development of their core product, a banking platform, which consists of various applications, interfaces and middleware components. As the company also provides the hosting of the platform, also hardware issues are important.

The design of to-be models takes place on different levels with different teams. High level guidelines for the general development vision of the platform are established by a superordinate planning team. Operating and IT architecture roadmaps are defined following these guidelines. These consist of models representing a snapshot of the desired architecture for up to three points in time in the future, taking into account existing vendor specific constraints if applicable. The planning of the development of the individual components is performed separately.

Based on the to-be architectures, the need for updating, replacing or developing new applications, middleware or hardware components is deduced. The information about predecessor and successor relationships is captured by versioning of applications, modules and interfaces. At the same time temporal restrictions, for example the time needed to develop a certain application, is modelled in special model types within an architecture modelling tool.

In order to determine a sequence of development activities the modelled relationship within the to-be models, the successor relationships as well as the determined time restrictions are consolidated and analysed. First, the general milestones and artefacts to be developed are defined. Afterwards, the development phases of the elements' lifecycles are planned in detail, i.e., specification and testing phases. Upon those specifications a rough project program schedule can be defined.

Finally, on the basis of the project outlines actual project planning can take place. For that purpose, the project proposals can be enriched with information about costs, quality metrics, staff, risks, and resources.

While conducting the method for planning transformation, the company found that the use of an appropriate modelling tool is most valuable, due to the large amount of models and model elements needed. The resulting number of interdependencies is not manually processable.

In contrast to the original method process model, during the application at the company the to-be model is continuously redefined. On the one hand, this is because time, budget and resource restrictions do not arise until the refinement of the project plans and are then reported back to the to-be architecture (cf. feedback loops described in section 4.4). On the other hand, the company encountered difficulties trying to define to-be architectures at a detailed level in advance. This leads to the assumption that an identical level of detail cannot be presupposed for both as-is and to-be models at the beginning of the transformation planning process. As a consequence, an iterative refinement of to-be models might be a useful amendment of the method as it is described above.

6 Discussion

The aim of our research has been to develop a method for enterprise engineering deriving transformation tasks from an analysis of EM snapshots of different points in time. Our proposed method focuses on supporting two (TR1, TR2) of three parts of enterprise transformation. Derived from the desired design results we have described the necessary design activities and techniques and anchored our approach in a framework capable of describing several points in time and their respective models. Thus, consistency between enterprise model snapshots of different points in time as well as transformation plan models is supported.

While our approach does not directly answer all of the questions raised, e.g., in the case studies in section 2 our approach enables its user to collect the necessary data to answer these questions. In this context the question of the business case for enterprise modelling emerges: The more information is put into these models, the more advanced the analyses may be and the more precisely a large number of decision problems can be addressed. However, the cost for modelling and maintaining the necessary information usually overcompensates the benefits (Aier et al. 2009a). Therefore the personal contribution of the 'enterprise engineer' still plays an essential role in planning enterprise transformation.

This leads to the question of situational methods (Harmsen et al. 1994): The potential complexity of enterprise models is very high and thus hardly manageable in a reasonable way. Introducing the dimensions of time raises the potential complexity exponentially. Therefore an element of sufficiency (Schelp and Aier 2009) needs to be introduced. However the question arises where self-restriction in modelling should be applied. The distinction of a limited number of well defined situations described by certain project types and contingency factors (Bucher et al. 2007) could guide this selection process. While some contributions (Aier et al. 2008; Leppänen et al. 2007) have made first steps in describing and identifying situations for EA management we did not yet comprehend the relationships of EA management goals, EA realisation approaches, enterprise engineering goals, and planning approaches and thus our method still lacks the concept of situation.

Finally complex systems like the ones described here usually require a high degree of division of labour for their management, design, and transformation (Shaw 1990). In such settings consistency is often a challenge. In our case for instance consistency between different levels of model detail as well as between the often independent development of very different model element types like products, processes, applications, and IT infrastructure is an issue. So far the method has only very inchoate techniques safeguarding consistency.

As sustainability literature indicates, addressing these four strategies *efficiency, sufficiency,* and *consistency* will foster a desirable enhancement of our method: a sustainable process for engineering enterprise transformation (Schelp and Aier 2009).

As a consequence we see the following implications for research:

For improving consistency, e.g., between model element types, we consider the definition of capabilities as a main topic. Capabilities describe business functions or services and will serve as a stable reference especially between business and IT units by providing meaning to related model elements like business processes or software components (Aier and Winter 2009; Albani and Zaha 2005; Albani et al. 2006).

Another field of future research should focus on an organisational concept defining (distributed) planning activities for organisational units, e.g., departments, divisions or entire enterprise. In this context, the issue of different (partial) to-be models—representing different functional areas or different modelling levels—for the same point in time will also become relevant.

Finally well-grounded definitions of situations of engineering enterprise transformation as well as the description of consequences for the design of the planning method are needed. Besides project types and contingency factors, such an approach should take the trade-off between detailed models and high costs of modelling as well as the manageable complexity into account.

7 Summary and Conclusion

With regard to planning purposes, this paper presents a method to derive preliminary project descriptions for enterprise transformation from enterprise models. The approach focuses on two of three parts of an enterprise transformation. In the first part the differences between two enterprise models of two points in time are analysed and a list of predecessor and successor relationships is derived. In the second part this list of differences is converted into logical blocks representing candidates for transformation projects. Subsequently, in the third part, the actual projects can be realised.

In order to address multi-period planning and dynamics, e.g., plan deviations, we have additionally proposed a concept for temporal dimensions that are relevant to represent model transformation. From this concept, valuable propositions for adapting enterprise meta models in order to enable transformation can be derived.

The long-term goal is to establish an information model that is able to capture all relevant aspects for planning while taking dynamic changes into account. This will require further research regarding meta model engineering for enterprise models and especially transformation models. Further enhancements of the method, e.g., including involved roles, will complement this approach.

Besides that, other aspects around the planning and transformation of enterprise models will be subject to further research. For example, strategic questions on where to start planning future states might be of interest.

References

Aier S., Winter R. (2009) Virtual Decoupling for IT/Business Alignment – Conceptual Foundations, Architecture Design and Implementation Example. In: Business & Information Systems Engineering 51(2) 2009, pp. 150–163

Aier S., Riege C., Winter R. (2008) Classification of Enterprise Architecture Scenarios – An Exploratory Analysis. In: Enterprise Modelling and Information Systems Architectures 3(1) 2008, pp. 14–23

Aier S., Buckl S., Franke U., Gleichauf B., Johnson P., Närman P., Schweda C. M., Ullberg J. (2009a) A Survival Analysis of Application Life Spans based on Enterprise Architecture Models. In: Mendling J., Rinderle-Ma S., Esswein W. (eds.) Enterprise Modelling and Information Systems Architectures, Proceedings of the 3rd Int'l Workshop EMISA. vol. LNI P-152 Gesellschaft für Informatik, Köllen, Ulm, pp. 141–154

Aier S., Gleichauf B., Saat J., Winter R. (2009b) Complexity Levels of Representing Dynamics in EA Planning. In: Albani A., Dietz J. (eds.) Proceedings of the 5th International Workshop on Cooperation & Interoperability – Architecture & Ontology (CIAO! 2009). vol. LNBIP 34 Springer, Amsterdam, pp. 55–69

Albani A., Zaha J. M. (2005) From Reference Model to Component Model. In: Desel J., Frank U. (eds.) Enterprise Modelling and Information Systems Architectures – Proceedings of the Workshop in Klagenfurt, October 24–25. vol. P-75 Enterprise Modelling and Information Systems Architectures Klagenfurt Gesellschaft für Informatik, Bonn, pp. 22–35

Albani A., Dietz J., Zaha J. M. (2006) Identifying Business Components on the Basis of an Enterprise Ontology. In: Konstantas D., Bourrières J.-P., Léonard M., Boudjlida N. (eds.) Interop-Esa 2005 – First International Conference on Interoperability of Enterprise Software and Applications, February 23–25, 2005. Springer, Geneva, pp. 335–347

Albani A., Barjis J., Dietz J. L. G. (2009) Advances in Enterprise Engineering III: 5th International Workshop, CIAO! 2009, and 5th International Workshop, EOMAS 2009, held at CAiSE 2009, Amsterdam, The Netherlands, June 8–9, vol. LNBIP 34. Springer, Berlin, Heidelberg

Antill J. M., Woodhead R. W. (1990) Critical Path Methods in Construction Practice 4th ed. John Wiley & Sons, New York

Arnoldi M., Beck K., Bieri M., Lange M. (2008) Time travel: A pattern language for values that change. http://www.lifeware.ch/upload/DMS/TimeTravel.pdf. Last Access: 2009-10-15

Ashby W. R. (1956) An Introduction to Cybernetics 2nd ed. Chapman & Hall, London

Bernus P., Nemes L. (1996) A framework to define a generic enterprise reference architecture and methodology. In: Computer Integrated Manufacturing Systems 9(3) 1996, pp. 179–191

Braun C., Wortmann F., Hafner M., Winter R. (2005) Method Construction – A Core Approach to Organizational Engineering. In: Haddad H., Liebrock L. M., Omicini A., Wainwright R. L. (eds.) Applied Computing 2005, Proceedings of the 2005 ACM Symposion on Applied Computing. vol. 2 ACM Press, New York, NY, pp. 1295–1299

Brinkkemper S. (1996) Method Engineering: Engineering of Information Systems Development Methods and Tools. In: Information and Software Technology 38(4), pp. 275–280

Bucher T., Klesse M., Kurpjuweit S., Winter R. (2007) Situational Method Engineering – On the Differentiation of 'Context' and 'Project Type'. In: Ralyté J., Brinkkemper

S., Henderson-Sellers B. (eds.) Situational Method Engineering – Fundamentals and Experiences. vol. 244 IFIP WG8.1 Working Conference on Situational Method Engineering – Fundamentals and Experiences (ME07) Geneva Springer, Boston, pp. 33–48

Buckl S., Ernst A. M., Matthes F., Schweda C. M. (2008) An Information Model for Landscape Management – Discussing Temporality Aspects. In: Aier S., Johnson P., Schelp J. (eds.) Proceedings of the 3rd Workshop on Trends in Enterprise Architecture Research (TEAR 2008). Sydney, pp. 363–374

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

Cîmpan S., Verjus H., Alloui I. (2007) Dynamic Architecture Based Evolution of Enterprise Information Systems. In: Proceedings of the International Conference on Enterprise Information Systems (ICEIS'07). Madeira, pp. 221–229

Dietz J. (2006) Enterprise Ontology – Theory and Methodology. Springer, Berlin

Eliot L. B. (1991) Information systems strategic planning. Computer Technology Research Corporation, Charleston, SC

Fox M. S., Gruninger M. (1998) Enterprise Modeling. In: AI Magazine 19(3), pp. 109–121

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

Harmsen A. F., Brinkkemper S., Oei H. (1994) Situational Method Engineering for Information System Project Approaches. In: Verrijn-Stuart A. A., Olle T. W. (eds.) Methods and Associated Tools for the Information Systems Life Cycle. Proceedings of the IFIP 8.1 Working Conference on Methods and Associated Tools for the Information Systems Life Cycle, Masstricht, The Netherlands, 26–28 September, 1994 Maastricht, The Netherlands North-Holland, Amsterdam, pp. 169–194

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

Hoogervorst J. A. P. (2009) Enterprise Governance and Enterprise Engineering. Springer, Berlin

IBM (1984) Business Systems Planning: Information Systems Planning Guide 4th vol. 1. IBM, Atlanta

Jonkers H., Lankhorst M. M., Doest H. W. L., Arbab F., Bosma H., Wieringa R. J. (2006) Enterprise architecture: Management tool and blueprint for the organisation. In: Information Systems Frontiers 8, pp. 63–66

Katz R. L. (1990) Business/enterprise modeling. In: IBM Systems Journal 29(4) 1990, pp. 509–525

Kerzner H. (2009) Project Management – A Systems Approach to Planning, Scheduling, and Controlling 10th. John Wiley & Sons, New York

King W. (1978) Strategic Planning for Management Information Systems. In: MIS Quarterly 2(1), pp. 27–37

Kirikova M. (2000) Explanatory capability of enterprise models. In: Data and Knowledge Engineering 33(2), pp. 119–136

Kosanke K., Vernadat F., Zelm M. (1999) CIMOSA: enterprise engineering and integration. In: Computers In Industry 40(2-3) 1999, pp. 83–97

Op 't Land M., Proper E., Waage M., Cloo J., Steghuis C. (2009) Enterprise Architecture – Creating Value by Informed Governance. Springer, Berlin

Lankhorst M. (2005) Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Berlin

Leppänen M., Valtonen K., Pulkkinen M. (2007) Towards a Contingency Framework for Engineering an Enterprise Architecture Planning Method. In: Proceedings of the 30th Information Systems Research Seminar in Scandinavia (IRIS 2007). Tampere, Finland

Liles D. H., Presley A. R. (1996) Enterprise modeling within an enterprise engineering framework. In: Proceedings of the 28th conference on Winter simulation. Winter Simulation Conference Coronado, California, United States ACM Press, New York, NY, pp. 993–999

March S. T., Smith G. F. (1995) Design and Natural Science Research on Information Technology. In: Decision Support Systems 15(4) 1995, pp. 251–266

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

Pavlak A. (2006) Enterprise Architecture: Lessons from Classical Architecture. In: Journal of Enterprise Architecture 2(2), pp. 20–28

Peffers K., Tuunanen T., Rothenberger M. A., Chatterjee S. (2007) A Design Science Research Methodology for Information Systems Research. In: Journal of Management Information Systems 24(3), pp. 45–77

Project Management Institute (2000) A Guide to the Project Management Body of Knowledge. Project Management Institute, Newtown Square

Pulkkinen M. (2006) Systemic Management of Architectural Decisions in Enterprise Architecture Planning. Four Dimensions and Three Abstraction Levels. In: Sprague Jr. R. H. (ed.) Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS-39). vol. 8 IEEE Computer Society, Honolulu, Hawaii, 179a (1–9)

Pulkkinen M., Hirvonen A. (2005) EA Planning, Development and Management Process for Agile Enterprise Development. In: Sprague R. H. J. (ed.) Proceedings of the 38th Annual Hawaii International Conference on Systems Sciences (HICSS-38). IEEE Computer Society, Los Alamitos, CA, 223.3 (1–10)

Rohloff M. (2008) Framework and Reference for Architecture Design. In: Proceedings of the 14th Americas Conference on Information Systems (AMCIS 2008). Toronto, pp. 1–14

Rouse W. B. (2008) A Theory of Enterprise Transformation. In: Systems Engineering 8(4), pp. 279–295

Schelp J., Aier S. (2009) SOA and EA – Sustainable Contributions for Increasing Corporate Agility. In: Sprague R. H. (ed.) Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS-42). IEEE Computer Society, Los Alamitos, pp. 1–8

Shaw M. (1990) Prospects for an Engineering Discipline of Software. In: Software, IEEE 7(6) 1990, pp. 15–24

Simon H. A. (1969) The Sciences of the Artificial. MIT Press, Cambridge, MA

Snodgrass R. T. (2000) Developing Time-Oriented Database Applications in SQL. Morgan Kaufmann Publishers

Sousa P., Lima J., Sampaio A., Pereira C. (2009) An Approach for Creating and Managing Enterprise Blueprints: A Case for IT Blueprints. In: Albani A., Barijs J., Dietz J. L. G. (eds.) Advances in Enterprise Engineering III, Proceedings of the 5th International Workshop CIAO! 2009 and 5th International Workshop EOMAS. vol. LNBIP 34 Springer, Amsterdam, pp. 70–84

Spewak S. H., Hill S. C. (1993) Enterprise Architecture Planning – Developing a Blueprint for Data, Applications and Technology. John Wiley & Sons, New York

Spewak S. H., Tiemann M. (2006) Updating the Enterprise Architecture Planning Model. In: Journal of Enterprise Architecture 2(2), pp. 11–19

Targowski A. (1990) The architecture and planning of enterprise-wide information management systems. Idea Group Publishing, Harrisburg

Tyler D. F., Cathcart T. P. (2006) A structured Method for Developing Agile Enterprise Architectures. In: Verma A. K. (ed.) Proceedings of the International Conference on Agile Manufacturing (ICAM 2006). Norfolk, Virginia, USA, pp. 1–8

Winter R., Bucher T., Fischer R., Kurpjuweit S. (2007) Analysis and Application Scenarios of Enterprise Architecture – An Exploratory Study. In: Journal of Enterprise Architecture 3(3), pp. 33–43

Stephan Aier, Bettina Gleichauf
Institute of Information Management
University of St. Gallen
Mueller-Friedberg-Strasse 8
CH-9000 St.Gallen
Switzerland
{stephan.aier | bettina.gleichauf}@unisg.ch