a|EA JOURNAL **Article**

An Information Model for Managed Application Landscape Evolution

By Sabine Buckl, Alexander M. Ernst, Florian Matthes, and Christian M. Schweda

Abstract

Planning, managing, and maintaining the evolution of the application landscape is a focal point of enterprise architecture (EA) management. Whereas, planning the evolution of business support provided by the business applications is understood as one challenge to be addressed in landscape management, another challenge arises in the context of traceability of management decisions. This article discusses the requirements regarding support for landscape management as risen by practitioners from industry, gathered in an extensive survey during which the tool support for EA management was analyzed. Thereby, a lack of support for this management discipline was discovered, which is caused by the way, application landscapes are modeled in tools. We subsequently discuss how to incorporate these requirements into an information model.

Keywords

enterprise architecture management, management tools, modeling, temporality, historization, traceability

INTRODUCTION

Over the last years enterprise architecture (EA) management has become an important management area, many companies are currently executing or planning to introduce in the nearby future. As a consequence of the increased attention, a multitude of methods for EA management has been developed by academic communities (Buckl et al, 2008; Lankhorst, 2005; Winter et al, 2007), standardization bodies (e.g., The Open Group, 2007), or practitioners (Dern, 2006; Keller 2007). Although these methods differ substantially concerning the quantity, abstractness, and granularity of the EA documentation, which is needed for performing EA management, the need for a documentation of the body of management is common. As a consequence, different methods for creating such a documentation as well as for maintaining its timeliness have been subjected to research, commonly attributing this documentation as a model of the EA (Fischer et al, 2007).

The methods and models have to cope with a set of challenges arising in the context of EA management, especially when the management of the application landscape as a central task is concerned. The term *application landscape* in this context refers to the entirety of the business applications and their relationships to other elements, e.g. business processes in a company. We abstain from using the term *application portfolio*, which we regard to have a narrower focus. During information gathering not only information about the *as-is* situation of the landscape has to be collected, but also information about *future* aspects, e.g. projects changing the application landscape, or business support provided by a newly introduced business application, has to be maintained. In order to get an overview on the relationships and dependencies of the various elements of the enterprise, different kind of visualizations, which we refer to as *software maps*, are typically used.

Figure 1 shows an exemplary software map, a so-called *process support map*, utilizing positioning of symbols to show, which business processes are supported by which business applications at which organizational units. Thereby, chevrons representing a sequence of processes make up the x-axis. The y-axis is made up of labels representing organizational units. The rectangles symbolize business applications, and their positioning indicates which business process is supported at which organizational unit.

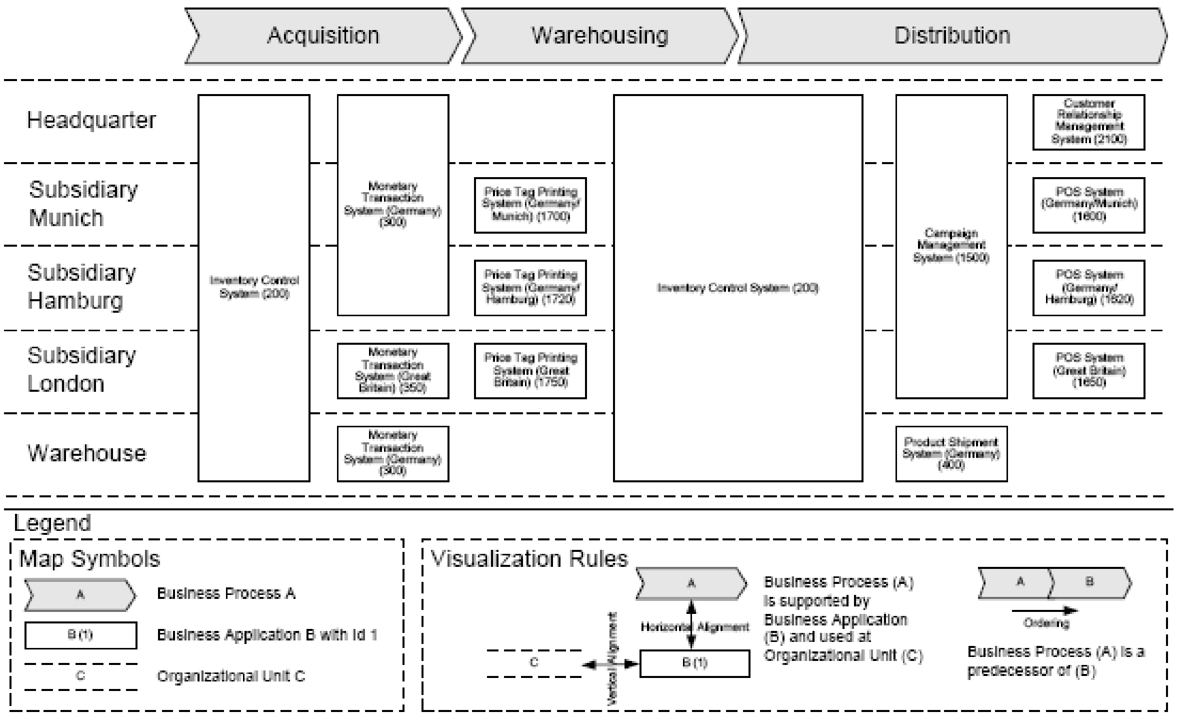

Figure 1. Exemplary Process Support Map

Different versions of process support maps are commonly used to document the evolution of the application landscape, illustrating either the as-is or future business support at a certain point in time (*planned for*). In order to create these documentations, the respective data has to be stored in a repository corresponding to an information model, which defines the respective elements to be modeled.

Furthermore, landscape management is closely connected to project portfolio management, as the selected project portfolio determines the future development of the application landscape in the next planning cycle. Regarding the state of the art in the context of project portfolio management, most decisions about project portfolios are currently based on *gut feel*, not on information, which is derived from a comparison of different *variants* of the landscape regarding quantitative or qualitative aspects (cf. Lankes and Schwenda, 2008). The landscape variants thus indicate the outcome of different project portfolios and can therefore be used to provide decision support, for example, in project portfolio management.

EA management and especially landscape management are understood to be endeavors following a typical management cycle consisting of the phases: Plan - Do - Check – Act (Deming, 1982; Shewart, 1986). Thereby, the traceability of management decisions taken in the *Plan* phase and implemented in the *Do* phase, must be ensured to control the achievement of objectives. In the context of a management cycle traceability of decisions can be achieved by storing previous states of the managed objects. The respective technique is sometimes referred to as *historization* An exemplary question in this context could be: Is the status of the planned landscape reached within the planned time frame or has the plan been changed? This information is subsequently used to prepare the next management cycle (*Act*). Consequently, a third type of information has to be stored in an information model for landscape management besides the *planned for* and the *variant* information as mentioned before: the moment in time the landscape was actually modeled (*modeled at*). From this discussion the following research question can be derived:

> *How should an information model for landscape management be designed to incorporate both business and technical aspects, and to support future planning and traceability of management decisions?*

This question must especially take the aspects of *temporality* as connected to landscape management into account. Therein, different versions of the landscape are of importance: the *current*, *planned*, and *target* version. The current landscape represents the status quo of the landscape *as is*, modeled at a certain time. The planned landscape represents a future state of the landscape as *to be* at a specific time in the future. In some publications on landscape management (e.g., Keller, 2007; Engles, 2008), the terms *as-is* and *to-be* are used to indicate the respective landscape version. We abstain from re-using this terminology, as especially the term *to*-be is often used ambiguously for both planned and target landscapes.

This state is modeled by an architect at a certain time, emphasizing e.g. the changes performed by projects up to that specific future date. The various projects and their impacts transforming the EA from current to a planned state, can be made explicit using roadmaps (Buckl et al, 2009). As a long term perspective the target landscape shows the architecture of the application landscape as envisioned at a certain time following the strategies and goals of the enterprise. There is no need to have projects defined transforming the current or planned landscape into the target one. Furthermore, the target landscape does not necessarily specify deployed business applications but refers to envisioned future support providers.

Summarizing, the traceability aspects of landscape management lead to three different *time-related* dimensions:

- firstly, a landscape is *planned* for a specific time,
- secondly, a landscape has been *modeled at* a certain time, and
- thirdly, different *variants* of a planned landscape may exist.

Figure 2 below illustrates the relationships between current, planned, and target landscape as well as the different dimensions relevant for landscape management.

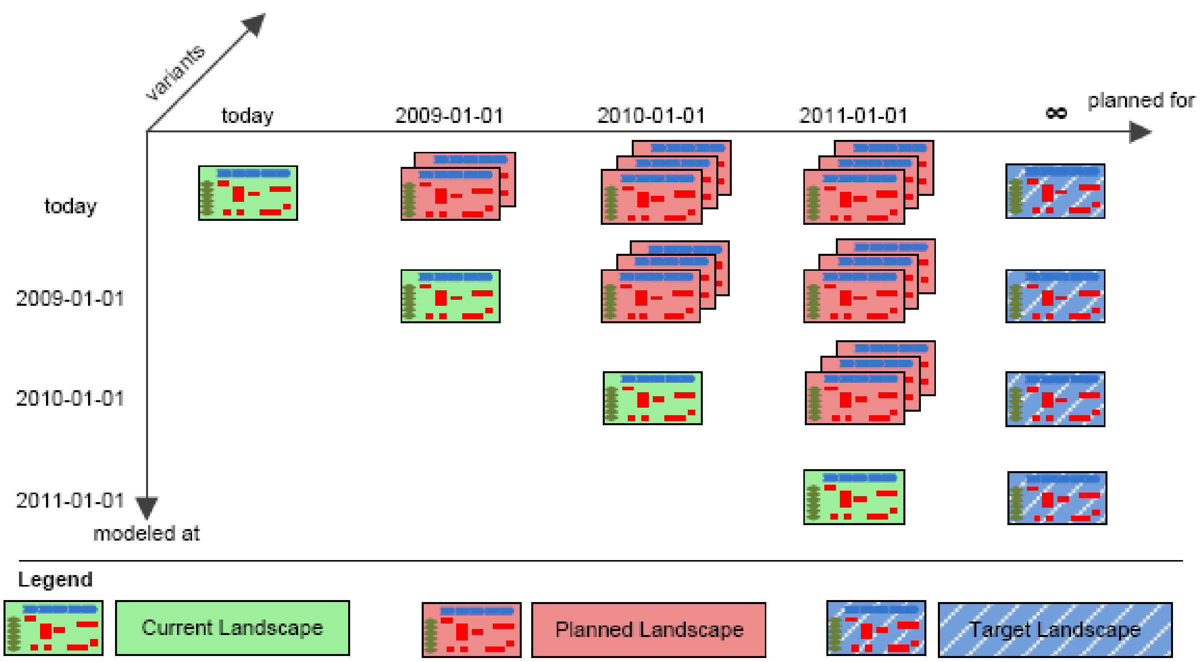

Figure 2. Current, Planned, and Target Landscape

The aforementioned research question is addressed in this article as follows: the next section gives an overview on current approaches to landscape management as described by researchers and practitioners in this field. Further, requirements - especially time-related ones - for an information model for landscape management are introduced. Thereby, a framework for the analysis of the support for landscape management is established. Alongside this framework an analysis of the current tool support for landscape management is performed in subsequent sections of this article and discusses ideas, which can be used to create an information model for landscape management fulfilling the aforementioned requirements. Therein, especially solutions originating from related modeling disciplines are taken into account to develop an information model suitable for documenting and planning application landscapes. The final section of this article hints at further areas of research in the context of EA management and in particular landscape management.

REQUIREMENTS FOR AND CURRENT APPROACHES TO LANDSCAPE MANAGEMENT

Due to the importance of managing the application landscape as a constituent of EA management, a number of different ways to approach this task have been proposed both in practice and academia. Subsequently, we give an overview on these approaches with an emphasis on the aspect of temporality.

In Braun and Winter (2005) the application landscape is referred to as a concept specifying the enterprise's business applications and their interdependencies. This information is rejected in the information model of via *interfaces* utilized to interconnect the *applications* and/or their inner *components*. References from these application level concepts (on the *application layer* as expressed by Braun and Winter (2005) to business level entities, e.g. the different types of *business processes* (on the *organizational layer* of the model) are present and can be used to explicate the way, how business support is provided. More sophisticated considerations are not directly supported, e.g. the question at which organizational unit which business process is supported, by which business application, cannot be answered based on the information model. The aspect of temporality is also only partially addressed, while the models contain ways to store life cycle states of applications, it does neither support planning transitions between life cycle states nor does it take projects into account.

In van der Torre et al (2006) the business applications as well as their relationships to other constituents of the EA are considered an important information asset, which should be presented to managers in an appropriate way to provide decision support. As presentation form of choice, they introduce a type of visualizations, called *landscape maps*, in which the business applications are related to business functions and products. This relationship is referred to as a ternary one, which could also be established between applications and two other concepts, although such considerations are not detailed in the article. Temporal aspects are not part of the approach, while ways to use the landscape map visualizations for interacting and changing the data in the underlying models are explicitly stated. Additionally, the focal point of the work of van der Torre et al (2006) is on the application landscape, not on the EA as a whole, i.e. projects are not considered in the approach.

A slightly different focus on managing the application landscape is taken in the work of Garg et al (2006). Therein, especially the aspect of the interfaces connecting the business applications is put under research. The number of interfaces associated to a business application is considered an important impact factor, e.g. when changes to the application landscape are considered. In this context, Garg et al (2006) put special emphasis on documenting and analyzing the current application landscape. This information is used as input to coordinate potential change processes affecting the landscape - especially concerning risks associated to these processes. While Garg et al (2006) take a rather detailed look on the business applications and their interconnections, relationships to business related concepts of the EA are not presented in their paper. Whereas, the topic of the evolution of the application landscape is indicated, actual planning of future states or transformation projects is not in the focus of this article.

In the work of Jonkers et al (2005) a language for enterprise modeling is presented, in which they target the three layers of *business*, *application*, and *technology*. The concepts introduced on the different layers can be used for modeling the current application landscape, especially for explicating the business support provided by applications (components) via offered interfaces. Further, the approach refines the description of the business support by adding the supplemental concepts of *business*- and *application-services* respectively. These concepts can be used to describe the existence of a support without having to specify, which actual application is responsible for the support. Thereby, target landscape planning could be facilitated. Nevertheless, planned landscapes are not in the scope of the model, which also contains no concept for modeling projects or explicating project dependencies.

The approach of *multi-perspective enterprise modeling* (MEMO) as discussed for example in the work of Frank (2002) that explicitly accounts for the modeling of IT concepts, as business applications, in an organizational and business context, described as *organizational units* and *roles* as well as *business processes* and *services*. The respective modeling language concerned with IT aspects, the *IT modeling language* (ITML) (see Kirchner, 2008), introduces the respective concepts, as e.g. the *information system*. According to the reference process described as complementing the language, these concepts should not only be used for documentation, but also for landscape planning. Nevertheless, projects are not part of the model, which also does not explicitly account for issues of time-dependence.

Beside the academic community, as alluded to above, also practitioners address the field of landscape management. In the work of Engles et al (2008) the application landscape is presented as a management subject embedded in the context of business and technical concepts, ranging from business processes to technical platform modules. The current landscape should, accordingly, be documented with references to these aspects, especially the technical ones. Complementing the current landscape, a so called *ideal landscape (*Target landscape in the terms used throughout this article) should be defined as part of a landscape management endeavor, incorporating technical visions of the landscape. Mediating between current and ideal, different *to-be landscapes* (in this article these landscapes are called *planned* ones) should be developed, each of these landscapes is assigned to a set of projects, which must be executed to actually realize the respective to-be landscape. Here, a strong relationship between the projects and the to-be landscapes should be maintained in an underlying model, nevertheless means for tracing back the evolution of a to-be landscape are not incorporated.

Another approach originating from practical experience is given in Kirchner (2008), which also emphasizes on the importance of a managed evolution of the application landscape in the context of EA management. Thereby, a map similar to the process support map is used, focusing on the support provided by business applications for products instead of organizational units. Following the product-centered approach, different states, current and planned, of the application landscape are modeled in order to support the planning process. Thereby, projects are linked to the constituents of the EA in order to support the deduction of planning variants, called *planning scenarios*. In order to support the evaluation of these variants, (Neimann, 2006) discusses the usage of historization to support traceability of management decisions. Although, the documentation aspect of EA management is addressed, no integrated information model supporting the future planning of the application landscape is presented.

Subsuming the state-of-the-art in managing application landscapes as presented in literature, many common aspects can be seen, although different approaches are employed especially concerning the aspect of temporality. Nevertheless, creating an information model of the application landscape is a widely accepted prerequisite employed in landscape management. In some of the papers, presented above, information models are provided, which introduce the concepts necessary for performing landscape management. These information models differ widely regarding the concepts and relationships introduced as well as regarding their complexity, because, among others, no common terminology for the concepts employed has yet been established. We regard, notwithstanding, such a model to be mandatory to approach landscape management as a whole and the important aspect of temporality in special.

Due to great interest from industry partners in information about EA management tools and especially their capabilities to address the concerns arising in the context of landscape management, an extensive survey - the *Enterprise Architecture Management Tool Survey 2008* - was conducted (Matthes et al, 2008).

The survey pursued a threefold evaluation approach, relying on two distinct sets of scenarios together with an online questionnaire. The survey was developed in cooperation with 30 industry partners (among others Allianz Group IT; Siemens IT Solutions and Services; Munich Re; O2 Germany; BMW Group; and Nokia Siemens Networks). Thereby, the first set of scenarios focuses on specific functionality, an EA management tool should provide, without connecting these functionalities to the execution of a typical EA management task, e.g. 1) *flexibility of the information model*, 2) *creating visualizations*, or 3) *impact analysis and reporting*. The EA management tools were further evaluated by the scenarios of the second set, which reflect tasks that have been identified as essential constituents of many EA management endeavors, for example: 1) *business object management*, 2) *IT architecture management*, or 3) *SOA transformation management*. One of the most prominent scenarios of the second part is the scenario *landscape management*, which is concerned with the managed evolution of the application landscape (Aier and Schönherr, 2007). The concern of the scenario was described by the industry partners as follows:

> *Information about the application landscape should be stored in a tool. Starting with the information about the current landscape potential development variants should be modeled. The information about the current application landscape and future states should be historicized to enable comparisons.* (Matthes et al, 2008)

Subsequently, a catalog of typical questions in the context of landscape management as raised by the industry partners is given:

- What does the current application landscape look like today? Which business applications currently support which business process at which organizational unit?
- How is, according to the current plan, the application landscape going to look like in January 2010? Which future support providers support which business process at which organizational unit?
- What was, according to the plan of 01-01-2008, the application landscape going to look like in January 2010?
- How does the target application landscape look like?
- What are the differences between the current landscape and the planned landscape, according to the current plan? What are the differences' reasons?
- What are the differences between the planned landscape according to the plan of 01-01-2008 and the current plan?
- What projects have to be initiated in order to change from the planned landscape (according to the current plan) to the target landscape? What planning scenarios can be envisioned and how do they look like?

Based on the questions from the industry partners and the different dimensions relevant for landscape management, the following requirements regarding an information model can be derived. An information model suitable for landscape management must:

- (R1) contain a ternary relationship in order to support analyses regarding current and future business support (which business processes are supported by which business applications at which organizational units),
- (R2) provide the possibility to specify envisioned business support in order to facilitate target landscape planning without having to specify implementation details of the business support,
- (R3) support the deduction of future landscapes from the project tasks, which execute the transition from the current to the future business support,
- (R4) ensure the traceability of management decisions by storing historic information of past planning states, which may be interesting especially if complemented with information on the rationale for the decisions, and
- (R5) foster the creation of landscape variants based on distinct project portfolios

in order to tightly integrate project portfolio management activities.

- From these requirements, we subsequently evaluate the support for landscape management as provided in the approaches from literature (see Table 1). A detailed discussion of the landscape management support provided by tool 1, tool 2, and tool 3 is given in the following section of this article.

Thereby, the support provided by the different approaches is indicated by different symbols ranging from complete fulfillment of the requirement (●) via partial fulfillment (◑) to approaches, which completely lack support for the analyzed requirement (○). In addition, an overview on the support provided by exemplary tools, which were analyzed during an extensive survey (Matthes et al, 2008), is shown in Table 1. A detailed discussion of the used information models shipped with the respective tools is given in the following section.

	[4]	[26]	[15]	[16]	[14]	[11]	[22]	Tool 1	Tool 2	Tool 3
R1	○	●	○	◑	●	●	●	○	●	●
R2	◑	◑	◑	○	◑	●	○	●	●	●
R3	○	○	○	○	○	●	●	○	◑	●
R4	○	○	○	○	○	○	◑	○	○	○
R5	○	○	◑	○	○	◑	◑	◑	○	●

Table 1. Existing Approaches and Tools and their Fulfillment of the Requirements

TOOL SUPPORT FOR LANDSCAPE MANAGEMENT

The solutions of nine major players in the market of EA management tools were analyzed regarding the information models, which they come shipped within (Matthes et al, 2008). Three different exemplary models as employed by the different tools are subsequently explicated to provide an overview about the current operationalizations of landscape management. The attributes are thereby not shown to improve readability but are mentioned in the description, if necessary for understanding. Due to reasons of confidentiality the names of the tools analyzed are omitted.

Prior to discussing the different approaches taken by the tools, the core concepts relevant in application landscape management, are introduced and defined in an informal way. The definitions are taken from the glossary as presented in Buckl et al (2008), although minor adaptation have been applied to suite the specific stetting of the article:

- **Business application.** A business application refers to an actual deployment of a software system in a certain version at a distinct location and hardware. Thus, business applications maintain versioning information in addition to the relationships to the business processes, they support at specific organizational units. In landscape management, the business applications are limited to those software systems, which support at least one business process. Further, the business applications are the objects, which are transformed by the projects considered in application landscape management.
- **Business process.** A business process is defined as a sequence of logical, individual functions with connections in between. A process here should not be identified with a single process step, as found e.g. in an *event driven process chain (EPC)*. It should be considered a coarse grained process at a level similar to the one used in value chains, i.e. partially ordered, linear sequences of processes. Additionally, a process maintains relationships to the business applications, which support him at the different organizational units. As in application landscape management, the business processes are considered to be *fixed*, i.e. they are not transformed by projects.

- **Business support provider.** A business support provider is a constituent of an application landscape, used to indicate that a related business process is supported at a distinct organizational unit, without giving a specification, which business application is likely to provide this support, if any. In spite of the similarities to the business application, the envisioned support provider is not affected by projects but has nevertheless a period of validity associated. Thereby, it references the point in time it has been modeled at and (optional) the point in time, the provider became invalid.
- **Organizational unit.** An organizational unit represents a subdivision of the organization according to its internal structure. An organizational unit is a node of a hierarchical organization structure, e.g. a department or a branch. In application landscape management, organizational units are considered *fixed* - thus, they are not transformed by projects.
- **Project.** Projects are drivers of organizational change. Therefore, adaptations of the application landscape are the result of a project being completed. Projects are scheduled activities and thus hold different types of temporal attributes, their *startDate* and *endDate* on the one hand. On the other hand, projects are *plannedAt* respectively *removedAt* certain points in time referring to the time of their creation or deletion. This effectively results in a period of validity, which is assigned to each project. In application landscape management, projects are considered to only affect business applications in general and their business support provided, in special. Projects do not affect business processes or organizational units in this model.

Starting with a basic approach to landscape management Tool 1 presents an information model containing landscape management related concepts, as shown in Figure 3. Here, the business process is connected with the organizational unit via a business support provider, which can be used to support target landscape planning (cf. R2). Whereas data gathered according to this information model can be used to generally analyze the business support for a business process, the relationship to the organizational unit, where the support takes place, is not derivable unambiguously (cf. R1).

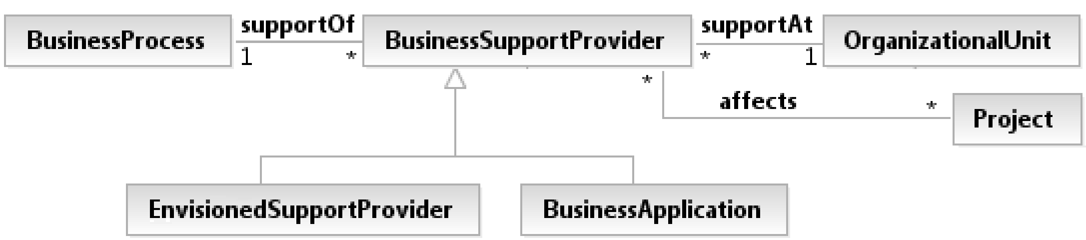

Figure 3. Information Model of Tool 1

Figure 4 shows exemplary data instantiating the information model from Figure 3. Analyzing this data, a statement, which business process is supported by the *Inventory Control System* at the *Subsidiary Munich* cannot be made.

Besides the missing ternary relationship between business process, organizational unit, and support provider, the only concept carrying temporal information - the project - is connected to the support provider via the relationship *affects*. Due to the missing ternary relationship no time information for the business support provided can be stored (cf. R3). In addition, planning variants of the landscape can only be built based on the business support providers instead of the business support provided (cf. R5). Consequently, Tool 1 only rudimentarily supports the management of current, planned, and target landscapes. While such information might be sufficient for future planning in a one dimensional manner, the requirements as risen by the industry partners concerning traceability and versioning cannot be addressed (cf. R4).

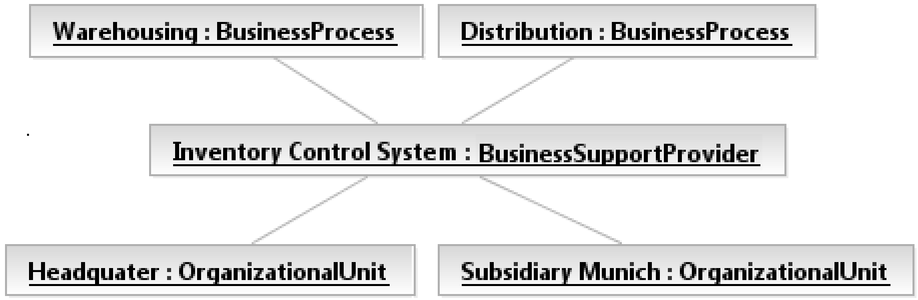

Figure 4. Instance Data Corresponding to Information Model of Tool 1

Business Support

Business Support represents the support of a specific business process by a specific business support provider at a specific organizational unit.

The information model of Tool 2 (see Figure 5) incorporates the ternary relationship between the business processes, the organizational units, and the business support providers by introducing a dedicated class and respective associations (cf. R1). The association *supportBy* is further assigned life cycle parameters using a mechanism similar to an association class in UML. Thus, it is possible to indicate that the business support provided by a specific instance of class *BusinessSupportProvider* is at a certain point in time in a specific life cycle phase, e.g. *planned* or *active* (cf. R2). This notion of life cycle is nevertheless disconnected from the concept of the project, which is independently associated to the class realizing the ternary relationship. While this association allows to model, that the support for a specific business process executed at a specific organizational unit is affected by a project, no mechanism to indicate, which *BusinessSupportProvider* actually is changed by the project, is present (cf. R3 and R5). This fact is caused by the * multiplicity on the *BusinessSupportProvider* end of the *supportBy* association. Therefore, an unambiguous mapping from projects to affected support providers is not possible. Further, the model does not support the creation of different landscape scenarios, as it is not possible to make projects or providers of business support belong together in one scenario. A mechanism for marking a *BusinessSupportProvider* an element of a target landscape is nevertheless provided via a flag attribute *target* in the association class *supportBy*. Historization of planned application landscapes is not supported (cf. R4) as no means for versioning instances corresponding to the model are given.

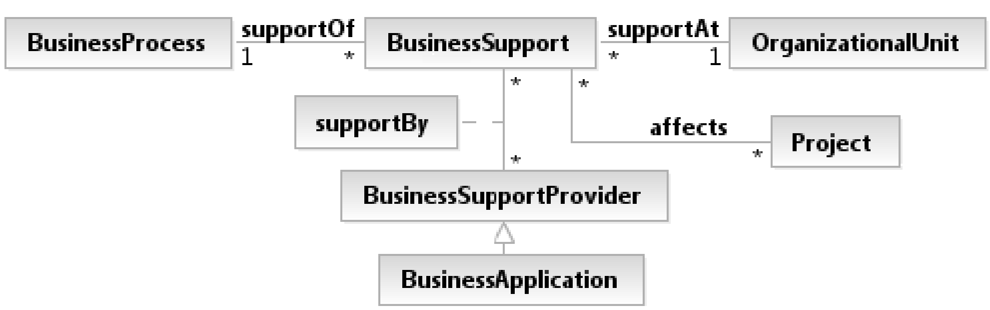

Figure 5. Information Model of Tool 2

Finally, the information model of Tool 3 (cf. Figure 6), which is only slightly different from the model of Tool 2, provides additional support for application landscape management - future state considerations are supported similarly as in Tool 2 (cf. R2). The information model contains the business support concept (cf. R1) and also implements temporality in a one dimensional manner by the project concept (cf. R3 and R5), which affects the business support and contains temporal information, e.g. start and end dates. Such information might be sufficient for planning the evolution of the EA, but is somewhat limited concerning traceability of changes to the plans (cf. R4), which would demand support for *bitemporal* modeling. As an example, one might think of a plan for the EA regarding the year 2010, which might look different as-of begin 2008 respectively begin 2009.

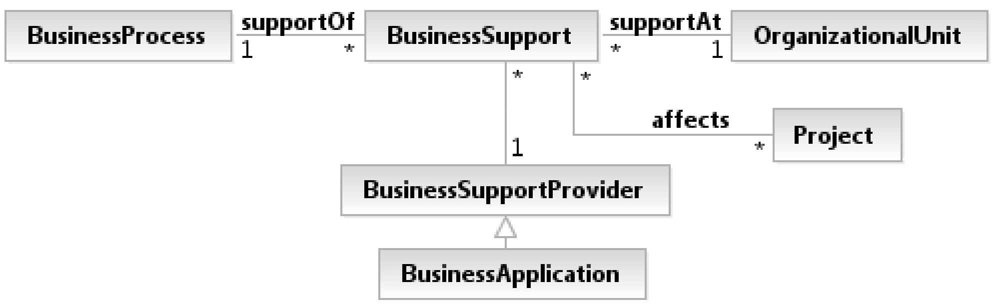

Figure 6. Information Model of Tool 3

Refer to Table 1 for an overview about the evaluation results of the current tool support regarding landscape management in general and temporality aspects in special.

DEVELOPING A TEMPORAL INFORMATION MODEL

This section presents an information model meeting the requirements as introduced above and thus also addressing the research question as stated previously. To give a convenient and well understandable presentation of the model, the section starts with introducing the idea of temporal modeling. This introduction accounts for methods and techniques supporting the creation of time-related models - it has thereby a special emphasis on *temporal patterns*, i.e. on *patterns for things that change over time* (Carlson et al, 1999). Prior, temporal databases are briefly alluded to, as an early means for explicating time dependency of data. Finally, the core facets of an information model capable to address the aforementioned requirements are explicated.

Introduction to Temporal Modeling

The question of how *to incorporate temporal dependencies into a model* has been repeatedly discussed in computer science. A very prominent approach to this question originates from the field of database research, where *temporal databases* were designed as means for *bitemporal modeling*, i.e. modeling of entities that change over time but have to maintain previous states accessible. Subsequently, we sketch the core principle of bitemporal modeling. For a more comprehensive treatment of the field in the context of databases (see for example the work of Date, 2000). In the simplest case, a timestamp would be added as a further column to a table, which should be enriched with temporal information. Thereby, it becomes possible to determine, since which point in time the respective row is valid. This simple solution has the drawback that it is not directly possible to specify, that a table row is valid for a certain period of time. Nevertheless this drawback can be resolved by adding another column for storing the end of the period of validity. If traceability of changes should further be considered, an additional temporal attribute has to be specified in addition to the *period of validity*. According to Date (2000), this can be achieved by introducing two more temporal attributes defining a respective time interval and thus capturing the sequence of states of a changing table. Such a table is than called a *bitemporal table*.

In the area of object-oriented modeling, similar discussions on how to incorporate time-dependencies have been undertaken. From these discussions, a set of *temporal (design) patterns* (e.g., Arnoldi et al, 2008) has emerged. The publication by Anderson (1999) gives a good overview. Prior to this overview, the article introduces basic time-related concepts as commonly encountered in temporal object-oriented models:

- **Event.** An event triggers a change of state in a system, i.e. a model. An event has a timestamp associated, responsible to record the time of the occurrence. According to [2], events may not always be (natural) first class objects in the respective modeling domain.
- **Time Interval.** A time interval has a specific *start* and *end* event, which allows the derivation of a duration of an interval.

These basic concepts are utilized by the temporal patterns, which address specific time-related design issues

One widely used pattern is the *temporal property* as discussed in Carlson et al (1999) and Fowler (2008). This pattern is also known as "'historical mapping" or "time-value pairs." If the need to track how this property has changed (or is expected to change) exists, the temporal property pattern can be used. The pattern therefore assigns a period of validity to the respective property value, to reflect that the value is only valid for a discrete interval of time.

For achieving this, the property, if not a priori modeled in a first class concept, is converted to a value class. This class not only contains the respective property value but is further augmented with two more properties indicating the *start*- respectively *end-time* of the validity period. Nevertheless, using this pattern to address issues of time-dependency for properties does not come without costs - the introduction of an additional class adds further complexity to the respective model, while lowering modeling clarity. The later becomes obvious, when considering multiplicities in the model. A property owner may have exactly one value for a property assigned at a specific point in time. However, there may be multiple instances of the respective value class assigned to the same owner, as they represent the history of property values over time. To address this loss of modeling clarity and make the model structure more concise, the utilization of a UML stereotype (the Object Management Group, 2005) <<*temporal*>> is recommended. Figures 7 and 8 exemplify the issue as raised above, by modeling the last name of a person as temporal property without and with the stereotype respectively.

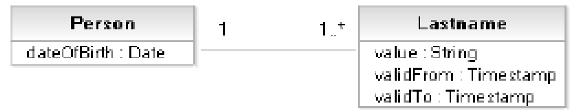

Figure 3. Temporal Property Model without Stereotype

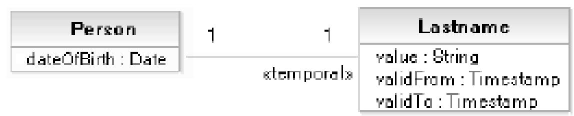

Figure 8. Temporal Property Model with Stereotype

In order to fulfill the requirements as mentioned previously, especially R4 and R5, the pattern *temporal association* (see Carlson et al, 1999) can be used on the *business support* concept, as this concept actually explicates a (ternary) association. This pattern introduces additional attributes similar to the temporal property to supply a period of validity for the business support.

If landscape plans for the same point in time created at different times should be compared to each other (cf. R5), the information concerning the point, when the project has been planned at, had to be considered. Consistently, the temporal pattern *edition* (Carlson et al, 1999) could be used to implemented this mechanism.

An Information Model for Landscape Management

The information models from Tools 2 and 3 as presented in Section 3 both form good starting points for a compulsory information model for landscape management fully satisfactory also in respect to the time-related requirements. Nevertheless, the incorporation of the project concept in both models is not completely satisfying in two ways:

- The *affects* relationship does not distinguish clearly between the different types of influence, a project can have, namely *introduction*, *migration*, and *retirement*.

- Projects do not only change the business support, but also influences business applications. In fact, projects (IT projects) mostly perform changes on the business application leading to a change in the business support provided thereby.

The first limitation, mentioned above, can be resolved easily, e.g. by introducing two relationships effectively replacing the *affects* relationship. These relationships can be labeled *introduces* and *retires* respectively, a migration is thus indicating by using both relationships. In contrast, the second limitation is not that easy to release; projects or parts thereof (*project tasks*) must consequently be associated to any *affectable* concept. This can actually be achieved in many different ways, e.g. via distinct project (task) types that affect only business applications or business supports. A maximum of genericity can be reached by introducing a basic concept for any concept, which can be affected by a project or a part thereof and to use respective inheritance in the information model.

We further pursue this approach and introduce the respective basic concept and its associations to *project tasks*, which are used to model distinct activities within a project. The model incorporating this idea is shown in Figure 9. In this information model, any *project affectable* can derive its period of validity from the *start* and *end dates* of the transitively associated projects. Thereby, inheriting from *project affectable* makes it possible to assign a project dependency to a concept in the information model. Nevertheless, using the standard UML-notation for inheritance would make the model less easy to perceive, as many classes are likely to inherit from *project affectable*. To make the resulting model more concise, we introduce an additional stereotype <<*projectDependency*>>, which can be assigned to a class in order to indicate, that this class is actually a subclass of *project affectable* (cf. Figure 10).

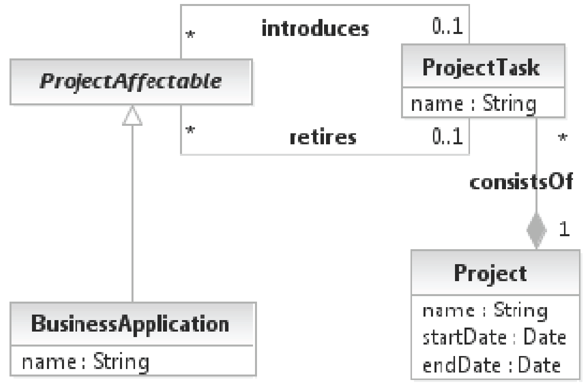

Figure 9. Project Affectable and Project with Exemplary Child Class

Figure 10. Exemplary Child Class with Stereotype

The information model presented in Figure 9 is complemented with an OCL constraint:

context ProjectTask
inv: introduces.type==retires.type

With this abbreviating notation at hand, an information model for landscape management satisfying the requirements (R1)-(R4) and explicating the stereotype <<projectDependency>> can be provided in Figure 11 on the next page.

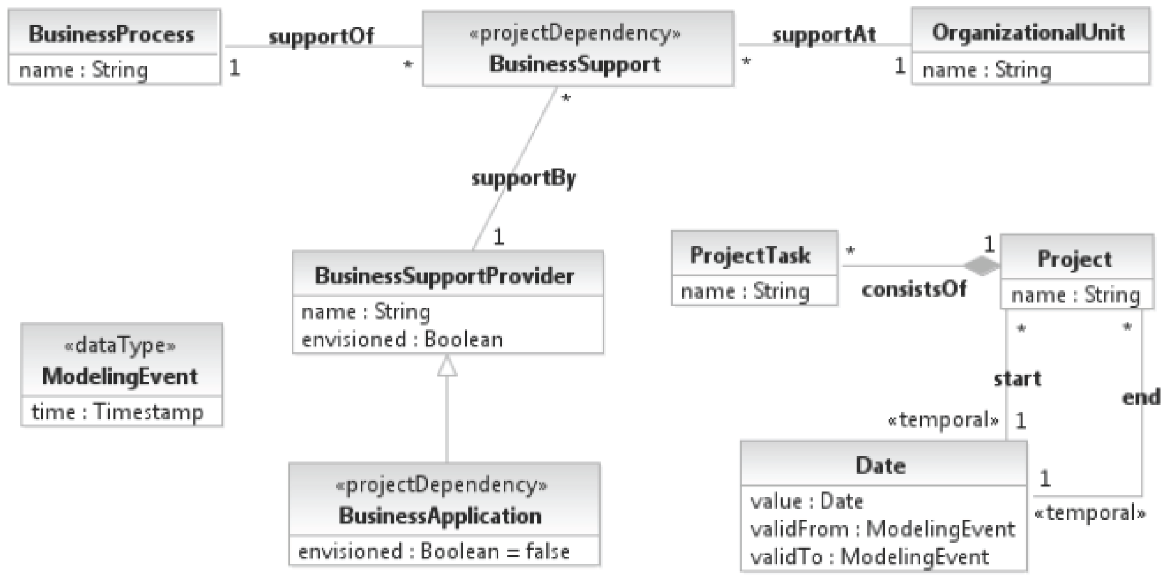

Figure 4. Time-Aware Information Model for Landscape Management

The model also fulfills the requirement (R5) to a certain extent. Landscape variants, based on certain project selections, i.e. planned project portfolios, can be derived from the model at any point in time. Nevertheless, these variants are not historized, as the model does not contain a concept for storing different portfolio selections. We do not regard this a major issue, because the project selections are most commonly used in a discussion process for project portfolio management, which leads to a certain selection to be approved. Additionally, making it possible to store different selections or, even more sophisticated, different timelines for the projects in a long-term project planning would require a number of additional concepts. This seems to us especially cumbersome, as the consequential complexity in creating model instances, might not relate to the benefits earned from this additional instrument of future planning. Furthermore, the practitioners, which have raised the requirements (R1-R5) (Matthes et al, 2008), did not state such medium-term multi-project portfolio variants as a topic of interest.

OUTLOOK

In this article, we motivated the importance of modeling temporal aspects of EAs in general and application landscapes more specifically. In Section 2 we considered selected state-of-the-art approaches to EA management, having a special emphasis on their support for explicating time-dependencies in their information models via eliciting requirements for temporal modeling support in landscape management, which we have collected from EA management practitioners. Subsequently, we evaluated existing tool support for landscape management. Based on the state-of-the-art analysis, we created and presented an information model capable of fulfilling the requirements and, therein, we applied temporal patterns, which had been previously shorthanded at the beginning of each respective section of the article.

The information model developed in this article can be considered quite complex, which might impede its implementation in a EA management tool and its use in practical EA management endeavors. Nevertheless, might an appropriate user interface be able to hide large parts of the complexity, thus making a convenient modeling experience possible. It has furthermore to be mentioned, that the information model presented still is somehow surprisingly simple in its structure, as the occasional introduction of the *business support* class actually representing an association simplified the creation of the model. If none such class existed, the inheritance mechanism to introduce project dependencies would not have worked, demanding the introduction of further *non-domain* concepts. As *non-domain* concepts, we regard concepts, which are not immediate elements of the domain of application landscape management. This aspect should be investigated in more detail in future research to find simplifying modeling techniques.

Another interesting direction of research is concerned with the integration of the information model into a comprehensive information model for EA management. Questions, on how to contain the temporal modeling aspects in just this cutout of the holistic model, would have to be discussed. An promising approach is the idea of EA management patterns as presented in Buckl et al (2008). Future research could transform the information model, presented in this article, into EA management patterns for time-dependent application landscape management. Thus, the information model could be reused in an organization specific EA management approach.

AUTHOR BIOGRAPHIES

Sabine Buckl is research assistant at the chair for Software Engineering of Business Information Systems at the Technische Universität München. Her research interests center around methods, models, and tools for the management of enterprise architectures and application landscapes as part thereof. In this area she is especially interested in methods and tools, which can be used to provide decision support regarding EA evolution and transformation. She can be reached at buckls@in.tum.de, or at the Technische Universität München, Institute for Informatics, Boltzmannstr. 3, 85748 Garching, Germany. http://www.systemcartography.info

Alexander M. Ernst is a research assistant at the chair for informatics 19 (sebis) at Technische Universität München. His research interests center around methods, models and tools for the management of enterprise architectures and application landscapes as part thereof. In this area, he is especially interested in a pattern based approach for enterprise architecture management, based on methodologies, viewpoints and information models. He can be reached at ernst@in.tum.de.

Florian Matthes holds the chair Software Engineering for Business Information Systems at the Technische Universität München. The current focus of his research is on enterprise architecture management, social software and model-driven web application engineering. He is co-founder of four high-tech software and service providers (CoreMedia, infoAsset, 20six, 21publish) with more than 160 employees. Since 2003 he has served as Dean of Studies at the Faculty for Informatics of TU München. Florian Matthes received his diploma degree in computer science 1988 from the Johann Wolfgang-Goethe University, Frankfurt am Main. After graduating with a doctorate in computer science from the University of Hamburg in 1993, he went to the Digital Systems Research Center (now HP SRC Classic) in Palo Alto, USA, with a post-doc fellowship. He then was assistant professor at the University of Hamburg where he contributed to a number of European and transatlantic research projects with a focus on persistent and mobile programming. From 1997 to 2002 he worked as a professor for computer science at the Technical University Hamburg-Harburg in the area of software architectures and programming models for cooperative information systems. He can be reached at matthes@in.tum.de.

Christian M. Schweda is research assistant at the chair for informatics 19 (sebis) at Technische Universität München. His research interests center around methods, models and tools for the management of enterprise architectures and application landscapes as part thereof. In this area, he is especially interested in quantitative methods and models, which can be used to operationalize and measure properties of the respective management body. As part of this research, Christian M. Schweda is developing models for simulating the behavior of the EA and the application landscape, to broaden the understanding of dynamic complexity arising in these complex structures. He can be reached at schweda@in.tum.de.

REFERENCES

Aier, S. and M. Schönherr (2007). *Enterprise Application Integration - Flexibilisierung komplexer Unternehmensarchitekturen*. Gito, Berlin. (in German).

Anderson, F (1999). A collection of history patterns. In P*attern Languages of Program Design*, Boston, Addison Wesley.

Arnoldi, M., K. Beck, M. Bieri, and M. Lange (2008). Time travel: A pattern language for values that change. http://www.manfredlange.com/publications/TimeTravel.pdf (cited 2008-11-25).

Braun, C. and R. Winter (2005). A comprehensive enterprise architecture metamodel. In J. Desel and U. Frank, editors, *Enterprise Modelling and Information Systems Architectures 2005*, volume 75 of LNI, pages 64-79. GI.

Buckl, S., A. Ernst, J. Lankes, and F. Matthes (2008). Enterprise architecture management pattern catalog (version 1.0, February 2008). Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Munich.

Buckl, S., A. Ernst, F. Matthes, and C. Schweda (2009). Visual roadmaps for enterprise architecture evolution. In The International Workshop on Enterprise Architecture Challenges and Responses (WEACR 2009), Daegu, South Korea, 2009

Carlson, A., S. Estepp, and M. Fowler (1999). Temporal patterns. In Pa*ttern Languages of Program Design*, Boston, Addison Wesley.

Date, C. (2000). *An introduction to database systems*. Addison-Wesley-Longman, Reading, Menlo Park, New York, 7 edition.

Deming, E. (2006). Out of the crisis, 1982. 10. G. Dern. Management von IT-Architekturen (Edition CIO). Vieweg, Wiesbaden.

Dern, G. (2006). *Management von IT-Architekturen (Edition CIO)*. Vieweg, Wiesbaden.

Engels, G., A. Hess, B. Humm, O. Juwig, M. Lohmann, and J.-P. Richter (2008). *Quasar Enterprise - Anwendungslandschaften serviceorientiert gestalten*. dpunkt.verlag, Heidelberg.

Fischer, R., S. Aier, and R. Winter (2007). A federated approach to enterprise architecture model maintenance. In *EMISA*, pages 9-22.

Fowler, M. (2008). Analysis patterns: Temporal property http://martinfowler.com/ap2/temporal Property. html. (cited 2008-11-25).

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

Garg, A., R. Kazman, and H.-M. Chen (2006). Interface descriptions for enterprise architecture. *Science of Computer Programming*, 61(1):4-15.

Jonkers, H., L. Goenewegen, M. Bonsangue, and R. van Buuren (2005). A language for enterprise modelling. In M. Lankhorst, editor, *Enterprise Architecture at Work*. Springer, Berlin, Heidelberg, New York.

Keller, W. (2007). *IT-Unternehmensarchitektur*. dpunkt.verlag, Heidelberg.

Kirchner, L. (2008). *Eine Methode zur Unterstützung des IT-Managements im Rahmen der Unternehmensmodellierung*. PhD thesis, Universität Duisburg-Essen, Berlin.

Lankes, J. and C. M. Schweda (2008). Using metrics to evaluate failure propagation and failure impacts in application landscapes. In *Multikonferenz Wirtschaftsinformatik*. GITO-Verlag, Berlin.

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

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

Niemann, K (2006). *From Enterprise Architecture to IT Governance - Elements of Effective IT Management*. Vieweg+Teubner.

Object Management Group – OMG (2005). Unified modeling language: Superstructure, version 2.0, formal/05-07-04.

Shewart, W. (1986). *Statistical Method from the Viewpoint of Quality Control*. Dover Publication, New York.

The Open Group – TOGAF (2007). "enterprise edition" version 8.1.1.

van der Torre, L., M. Lankhorst, H. ter Doest, J. Campschroer, and F. Arbab (2006). Landscape maps for enterprise architectures. In E. Dubois and K. Pohl, editors, CAiSE, volume 4001 of *Lecture Notes in Computer Science*, pages 351-366. Springer.

Winter, R., T. Bucher, R. Fischer, and S. Kurpjuweit (2007). Analysis and application scenarios of enterprise architecture - an exploratory study. *Journal of Enterprise Architecture*, 3(3):33-43.