An Information Model Capturing the Managed Evolution of Application Landscapes

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

Technische Universität München
Boltzmannstr. 3
85748 Garching
{buckls,ernst,matthes,schweda}@in.tum.de

Abstract. Projects are the executors of organizational change and hence in charge of the managed evolution of the application landscape in the context of enterprise architecture (EA) management. Although the aforementioned fact is widely agreed upon, no generally accepted information model addressing the challenges arising in the context of future planning and historization of management decisions concerning projects yet exists. This paper addresses this challenge by identifying requirements regarding an information model for linking projects and application landscape management concepts from an extensive survey, during which the demands from practitioners and the existing tool support for EA management were analyzed. Furthermore, we discuss the shortcomings of existing approaches to temporal landscape management in literature and propose an information model capable of addressing the identified requirements by taking related modeling techniques from nearby disciplines into account.

Keywords: Enterprise Architecture Management, Project Portfolio Management, Temporal Modeling.

1 Introduction

The need for an alignment between business and IT in an organization has been an important topic for both practitioners and researchers ever since the 90's of the last century [15]. Nevertheless, enterprise architecture (EA) management as a means to achieve this alignment has only recently become an important topic, many companies are currently addressing or planning to address in the nearby future. As a consequence of the greater demand from practice, a multitude of approaches to EA management has been proposed in academia [7,19], by standardization bodies [11,26], or practitioners [12,22]. These approaches differ widely concerning the coverage of different aspects of EA management, as e. g. infrastructure or landscape planning. While documentation, analysis, and planning of the EA are common tasks throughout all approaches, the level of abstractness and granularity of information needed to perform EA management differs – for a comprehensive comparison see e. g. [1]. As a consequence, different

A. Albani, J. Barjis, and J.L.G. Dietz (Eds.): CIAO!/EOMAS 2009, LNBIP 34, pp. 85–99, 2009.

information models[1] defining the structure of the respective EA documentation are used in the approaches.

Notwithstanding, certain similarities among the EA management approaches exist, literally all approaches agree on the *application layer* being an important management asset [1,24]. To holistically manage the *application landscape*[2], as the entirety of the business applications and their relationships to each other as well as to other elements, e. g. business processes, of an enterprise, is therefore a widely accepted central task of EA management.

EA management in general and landscape management more specifically can be considered *typical* management processes thus, adhering to the management cycle containing the phases *Plan*, *Do*, *Check*, and *Act* [10,25]. Following the periodic phases, the importance of traceability concerning management decisions increases. This means, that the realization of decisions taken in the *Plan* and executed in the *Do* phase is evaluated during the *Check* phase to determine potential process improvements, which are subsequently applied in the *Act* phase. To achieve this type of *self-improving* process, it is necessary to make past decision explicit and accessible during evaluation. The respective technique of archiving management decisions is usually called *historization.* A typical EA management question, which needs historic information for answering, could be: Is the status of a planned application landscape reached within the planned time frame or has the plan been changed? The type of time-dependence employed in making that information available is different from the dependence as alluded to above, such that it has to be incorporated into an information model for EA management separately. From this discussion the following research question guiding the remainder of the article 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 historization of management decisions?

This question especially alludes to the aspects of *time-dependency* as connected to application landscape management. Therein, different types of landscapes are of importance [3]:

– the *current* landscape, reflecting the actual landscape state at a given point in time,
– *planned* landscapes, which are derived from planned projects for transforming the landscape until a certain point in time, and
– the *target* landscape, envisioning an ideal landscape state to be pursued.

[1] Consistent to the terminology as used e. g. in [8], we call the meta model for EA documentation an *information model.*

[2] We do not use the term *application portfolio*, which is widely used in this area, as we regard it to have a narrower focus on just the business applications without considering related artifacts, as e.g. business processes.

[3] The different types are sometimes also called *as-is* and *to-be* landscapes, thereby abstaining from the distinction between planned and target landscape.

This information has to be maintained consistently in an integrated information model for the EA, describing the three types of landscapes adequately. Therein, especially the relationship between application landscape and project portfolio management, concerning the planned landscapes, has to be considered. To ensure consistency in between these management tasks, the planned landscape should be derived from the transforming projects selected for execution in the next planning cycle. The main purpose of the project portfolio management process is to identify those projects, which should be accomplished [29]. Thereby, a distinction has to be made between projects that have to be performed, e.g. due to end-of-support – *run the enterprise* – and projects that should be performed, e.g. due to strategic reasons – *change the enterprise* [3].

The close linkage between application landscape and project portfolio management should be used to create different planned landscapes for the next planning period based on distinct project portfolio selections. Thereby, analyses of the application landscape can be used to provide decision support for project portfolio management. In order to identify the appropriate portfolio, analyses regarding the dependencies between projects have to be conducted. Therefore, not only the required resources, e.g. persons, tools, etc, need to be considered but also dependencies between projects regarding affected artifacts, e.g. business applications, interconnections, etc. These analyses support the identification of potential conflicts regarding time-dependencies if a project in the realization phase is delayed. The aspect of time, as related to projects, must be considered a highly complex one, which is also not well reflected in current information models for landscape management (cf. Section 2 and [6]).

For the planned and target landscape different (historic) states may exist and evolve over time, i. e. a planned landscape is planned for a certain point in time and is modeled at a certain (different) point in time. The latter is also true for the target landscape, which might be modeled differently at different points in time. Additionally, the idea of *variants* for planned landscapes resulting from the selection of different project portfolios has to be taken into account. Summarizing, three different *time-related* dimensions exist:

- firstly, a landscape is *planned for* for a specific time[4],
- secondly, a landscape is *modeled at* a certain point in time, and
- thirdly, different landscape *variants* of a planned landscape may exist.

Figure 1 illustrates the relationships between current, planned, and target landscapes as well as the different dimensions relevant for landscape management.

In the remainder of the article, the research question, as alluded to above, is approached as follows: Section 2 gives an overview about current approaches to landscape management as described by researchers and practitioners in this field. In Section 3, requirements – especially time-related ones – for an information model for landscape management are elicited. These requirements are subsequently incorporated in a temporal information model for documenting and

[4] Therein, the current landscape can be regarded to be planned for the current time, while the target landscape can be regarded to be planned for an infinite future point in time.

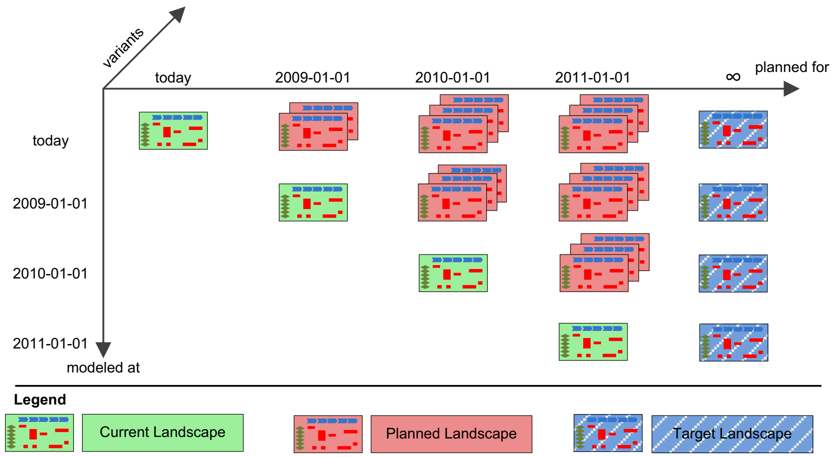

Fig. 1. Current, planned, and target landscape

planning application landscapes, presented in Section 4. Final Section 5 concludes the paper and briefly sketches interesting directions for future research.

2 Related Work

Application landscape management is, as outlined before, widely considered an important task in EA management. Consequently, a number of different ways to approach this task have been proposed both in practice and academia. Subsequently, we briefly introduce selected approaches and especially emphasize on the coverage of time-related aspects.

The EA management approach presented by van der Torre et al. in [27,28] considers the business applications as well as their relationships to other elements of the EA an important information asset. As part of the approach, ways to display this information to managers in an appropriate way to provide decision support are discussed. The recommended type of visualization introduced in the paper, is the so-called *landscape map* detailing the *business applications* in relationship to the provided *business functions* and the respective *products* of the company. This relationship is referred to as a ternary one. Aspects of time-dependency are not discussed, also projects are not alluded to in the article, as the approach mainly focuses on application landscapes.

In [4], Braun and Winter present the application landscape as the set of an enterprise's business applications and their interdependencies. Respectively, the information model contains the classes *application* and *interface* to describe the web of dependencies. These concepts from the *application layer* can be connected to elements from the *organizational layer*, i.e. the *business processes* in order to explicate the business support relationships. Nevertheless, the model does not account for analyses in multi-organizational unit environments, as no concept for relating the organizational unit, where the business support takes place, is provided. Further, the concept of time-dependence is only partially addressed in the information model – means for planning landscape transitions via projects are not included therein.

The interfaces connecting the business applications are also focused by Garg et al. in their approach presented in [14]. Therein, especially the total number of interfaces associated with an application is regarded an important quantitative indicator, which should be taken into consideration, e.g. if changes are applied to the application landscape. Consequently, the approach emphasizes on analyses regarding the current landscape, while planned landscapes in special and time-related concepts in general are not part of the approach. Additionally, due to the strong application focus of the model, business aspects, i.e. business processes are omitted in the model. Also, transformation projects changing the application landscape are not alluded to in the approach.

As part of the *systemic enterprise architecture methodology (SEAM)* Le and Wegmann [20] discuss a way to model enterprise architectures for reasons of documentation and analysis. Prior to presenting the model, some requirements are introduced, especially focusing on *multi-level-modeling* from company level down to component level. Furthermore, the importance of traceability is emphasized, although not targeting temporal traceability but inter-level traceability of relationships. The approach further introduces the more abstract concept of the *computational object*, effectively replacing the business application. Nevertheless, time-related aspects are not discussed in the approach; projects as executors of organizational change, which correspond to the external drivers such as new competitors, laws, changing markets, etc., are also not alluded to.

Jonkers et al. present in [17] a language for enterprise modeling, 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 modelling (MEMO)* as discussed e.g. in [13] 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 is the *IT modeling language (ITML)* [18] 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, also practitioners address the field of EA management and landscape management. A representative approach developed by a consulting company is the *QUASAR Enterprise* approach [12]. In this approach the application landscape is presented as management subject related with business and technical concepts, ranging from *business processes* to *technical platform modules*. The current landscape is consequently documented with references to these concepts. Complementing, a so-called *ideal landscape*[5] should be defined in the application landscape management process. Different *to-be* (planned) landscapes are created describing the transition roadmap from current towards target landscape. These intermediary roadmap landscapes should according to the approach maintain relationships to the respective projects, nevertheless means for tracing back the evolution of the planned landscapes are not discussed in the approach.

The Open Group is a consortium of practitioners addressing the field of EA management, whose main purpose is to develop standards in this area. Therefore, they proposed *The Open Group Architecture Framework (TOGAF)* [26], which provides a cyclic process model for EA management, the *Architecture Development Method*. This cycle contains, among others the phase *architecture vision*, which is concerned with the development of an target architecture. In order to manage an evolution in the direction of the target architecture, several intermediate *transition architectures*, which we would refer to as planned landscapes, are developed. In addition to the cyclic process, TOGAF 9 [26] includes an information model, which describes the content of architecture development. Thereby, projects are introduced via the concept of *work packages*. Nevertheless, these work packages are neither linked in the information model to the architectural elements, which they affect, nor does the provided information model provides concepts to model time-dependencies between the different elements.

3 Elicit Requirements for Landscape Management

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 in 2008 [21]. The survey pursues 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, sd&m – software design & management, Siemens IT Solutions and Services, Munich Re, O2 Germany, BMW Group, 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*

[5] *Target* landscape in the terms used throughout this paper.

visualizations, or *3) impact analysis and reporting*. The EA management tools are further evaluated by the scenarios of the second set, which reflect tasks that have been identified as essential constituents of many EA management endeavors, e.g. *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 [2]. 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. [21]

Closely related to the landscape management scenario is the *project portfolio management*, which is concerned with providing decision support for the selection of an appropriate portfolio of projects to be realized within the next planning period as alluded to in Section 1. Subsequently, a catalog of typical questions in the context of landscape and project portfolio 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?
- Which EA artifacts, e.g. business applications, are modified/created/retired by the individual project proposal?
- Which project proposals (run the enterprise) have to be accomplished in any case?

Based on the questions from the industry partners, the different landscape types, and time-related dimensions relevant for landscape management (see Section 1), 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 providers 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,
R5 foster the creation of landscape variants based on distinct project portfolios in order to tightly integrate project portfolio management activities, and
R6 allow impact analyses regarding dependencies between different projects, which affect the same EA elements, e.g. organizational units, business application, business processes.

These requirements have been used in [6] to evaluate the support for landscape management as provided in approaches from literature and in approaches provided in three commercially available EA management tools. The result of this evaluation is that none of the analyzed approaches completely fulfills all requirements given above.

4 Developing a Temporal Information Model

In this section, we present an information model capable to fulfill the requirements as introduced above. Hence, the model addresses the research question as stated in Section 1. Such a model could be described using different modeling languages, of which an object-oriented one – namely the UML – has been chosen. This choice seems to us equally suitable to potential alternatives, as e.g. Entity/Relationship (E/R) modeling. This opinion is supported by the fact that the subsequently presented object-oriented information model can be easily converted to an E/R model. It has further to be noted that we do not regard the UML as the language of choice for presenting the information modeled according to the information model – other graphical notations, i.e. means to define viewpoints [16], exist, which a by far more appropriate for visualizing enterprise architecture information. The object-oriented information model hence only defines the schema for storing this information.

To prepare the discussions on the information model, we provide a short glossary (see Section 4.1) of core model concepts. These concepts are reflected in the information provided in Section 4.2, which is an augmentation of a model initially discussed in [6].

4.1 Glossary

In this section, the core concepts relevant in application landscape management are introduced and defined in an informal way. The definition are taken from the glossary as presented in [7], although minor adaptations have been applied to suite the specific setting 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 a 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.

Envisioned support provider. An envisioned support provider is a constituent of a target 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. Inspite 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 envisioned 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.

Project. Projects are executors 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 can be split into smaller constituents, so called *project tasks*.

With these core concepts and main attributes at hand, an information model satisfying the requirements corresponding to application landscape management can be developed.

4.2 An EA Information Model for Modeling Project Dependencies

Based on the discussions in [6] an information model has been proposed. This information model (cf. Figure 2) is capable to satisfy the requirements **(R1)** to **(R4)**. It also fulfills **(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 project portfolio management discussion process, 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 introduce 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)** (cf. [21]), did not state such medium-term multi-project portfolio variants as a topic of interest.

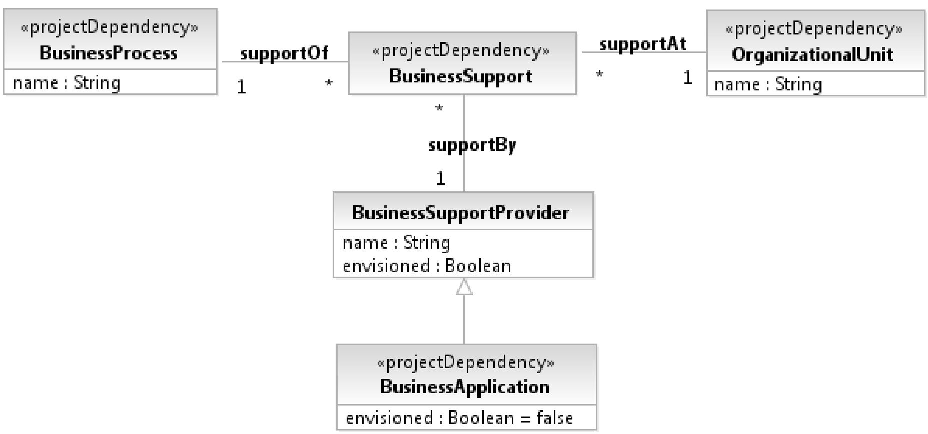

Fig. 2. Information model satisfying **(R1)**, **(R2)**, **(R3)**, **(R4)**, and **(R5)**

The aforementioned model uses two UML stereotypes [23] (`<<temporal>>` and `<<projectDependency>>`) to enhance model clarity and support concise modeling. As these stereotypes cannot be considered widely-known, they are subsequently explained in detail.

The first stereotype `<<temporal>>` has been proposed in [9] in the context of the modeling pattern *temporal property*[6]. This pattern allows to model that a property of an object can change over time and these changes have to be tracked. Nevertheless, using this pattern to address issues of time-dependency for properties does not come without costs – the attribute, which is converted to a temporal property, is changed to a multi-valued one, i.e. one of multiplicity *. A class owner may have exactly one value for a property assigned at a specific point in time. Nevertheless, there may be multiple instances of the respective value

[6] This pattern is also known as *historical mapping* or *time-value pairs*.

class assigned to the same owner, as they represent the history of property values over time. This issue is resolved by introducing the `<<temporal>>` stereotype, indicating that a property might have multiple values without overlap in their periods of validity.

The second stereotype `<<projectDependency>>` is introduced, to support concise modeling of the relationship between the projects, i.e. their constituing tasks, and the architectural constituents. A project task can affect an architectural constituent in four different ways:

– *Introduce* a new constituent to the architecture.
– *Migrate* from one constituent to another, e.g. a functionality.
– *Retire* a constituent from the architecture.
– *Change* an existing constituent *below* the EA level.

The first three types are quite obvious, although the fourth type is also important, as it can be used to subsume changes *below* the architectural level. These may e.g. be changes to the components of a business application leading to a new version of the application, although the application itself has not changed from an EA point of view. Making projects performing minor changes explicit is necessary to completely fulfill **(R6)**, in order to prevent multiple projects from performing concurrent and potentially conflicting changes.

EA constituents, which can be affected by projects, must hence be related to the corresponding project tasks, which can be achieved in many different ways in the information model. 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 3.

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*.

In order to ensure model consistency, a modeling constraint applies – defining that a project task might not migrate between EA constituents of different types:

```
inv: Migration
introduces.type == retires.type
```

Completing the information model the value of the derived attribute *isMaintenace* is complemented with a computation formalism to automatically derive

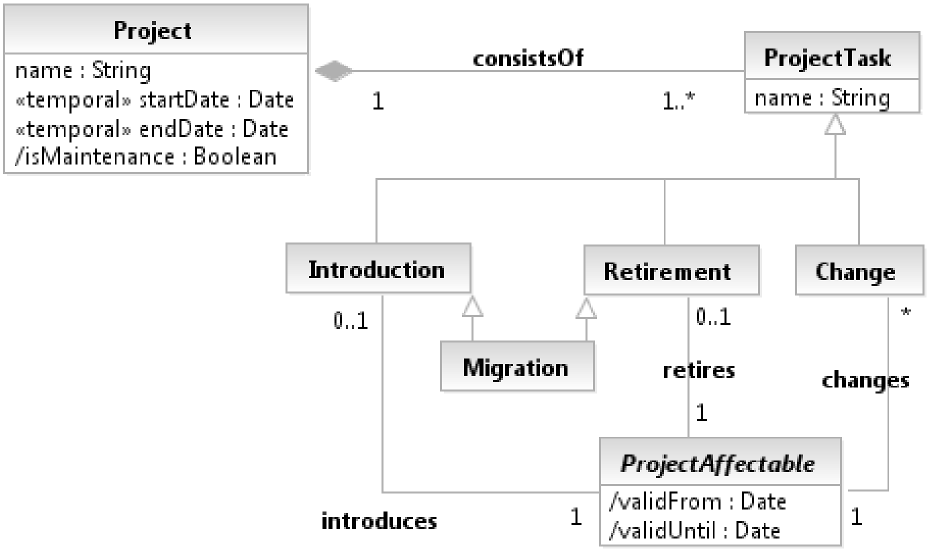

Fig. 3. Project affectable and project with exemplary child class

the distinction between *maintenance* and *transformation* projects as discussed above.

```
derive: Project
isMaintenance = consistsOf->forAll(t|t.oclIsTypeOf(Change))
```

The periods of validity for an architectural constituent are derived, as alluded above, from the associated project tasks:

```
derive:ProjectAffectable
validFrom = introduces==null?null:introduces.project.endDate
```

A similar derivation rules applies for the end date for the period of validity. For both dates, the special value *null* can be computed, which indicates that the corresponding architectural element has no distinct date of introduction or retirement. This means that the project, which introduced the element, took place before EA documentation was introduced or that no retirement project is yet planned respectively. For further discussions on how to incorporate these special dates into landscape transformation planning see e.g. [5].

5 Reflection and Outlook

In this article, we discussed techniques for modeling the project dependencies 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 project- and time-dependencies in their information models. Requirements for modeling the management evolution of an application landscape, which have been collected at EA management practitioners (cf. [21]), were outlined in Section 3. Subsequently, we created and presented an information model capable of fulfilling these requirements. Therein, we applied temporal patterns, i.e. patterns for things that change over time.

The information model presented in this paper has yet not been validated in practice. In doing so especially the complexity of the project dependency modeling might be a usage impediment, which should be addressed by an appropriate user interface. Such an interface can be helpful to conceal large parts of the complexity – thereby making a convenient modeling experience possible. Nevertheless, no such user interface has yet been created, which would be a prerequisite to testing the information model in a practical environment.

The model introduced in the paper is further limited to projects affecting business applications, business processes, organizational units, and their relationships to each other. This does not completely reflect on the role of the project in EA management in general, as a project can also affect and change other EA constituents, such as e.g. infrastructure components or hardware devices. The concept of the *project affectable* as presented in Section 4 could nevertheless be extended to other EA constituents and hence form a reusable building block for incorporating project dependencies in EA information models. This relates well to the approach of EA management patterns as presented in [7], although more in-depth research is yet to be undertaken.

The latter discussion points towards another interesting direction of research. Object-oriented modeling languages, albeit their wide proliferation as discussed in Section 4, do not provide dedicated means for constructing time- and project-dependent EA information models. Hence, techniques as temporal patterns have to be utilized. These techniques could nevertheless by incorporated in an augmented object-oriented modeling language with specific support for creating EA information models. Future research is to show, how such a language could look alike.

References

1. Aier, S., Riege, C., Winter, R.: Unternehmensarchitektur – Literaturüberblick Stand der Praxis. Wirtschaftsinformatik 50(4), 292–304 (2008)
2. Aier, S., Schönherr, M.: Enterprise Application Integration – Flexibilisierung komplexer Unternehmensarchitekturen, Gito, Berlin (2007) (in German)
3. Aier, S., Schönherr, M.: Flexibilisierung von Organisations- und IT-Architekturen durch EAI. In: Enterprise Application Integration – Flexibilisierung komplexer Unternehmensarchitekturen Band I, Berlin, Gito (2007) (in German)
4. Braun, C., Winter, R.: A comprehensive Enterprise Architecture Metamodel. In: Desel, J., Frank, U. (eds.) Enterprise Modelling and Information Systems Architectures 2005. LNI, vol. 75, pp. 64–79. GI (2005)
5. Buckl, S., Ernst, A., Kopper, H., Marliani, R., Matthes, F., Petschownik, P., Schweda, C.M.: EAM Pattern for Consolidations after Mergers. In: SE 2009 – Workshopband, Kaiserslautern (2009)

6. Buckl, S., Ernst, A., Matthes, F., Schweda, C.M.: An Information Model for Landscape Management – Discussing temporality Aspects. In: Johnson, P., Schelp, J., Aier, S. (eds.) Proceedings of the 3rd International Workshop on Trends in Enterprise Architecture Research, Sydney, Australia (2008)
7. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F.: Enterprise Architecture Management Pattern Catalog, Version 1.0. Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Munich (February 2008)
8. Buckl, S., Ernst, A.M., Lankes, J., Matthes, F., Schweda, C., Wittenburg, A.: Generating Visualizations of Enterprise Architectures using Model Transformation (Extended Version). Enterprise Modelling and Information Systems Architectures – An International Journal 2(2) (2007)
9. Carlson, A., Estepp, S., Fowler, M.: Temporal patterns. In: Pattern Languages of Program Design. Addison Wesley, Boston (1999)
10. Deming, E.W.: Out of the crisis, Massachusetts Institute of Technology, Cambridge (1982)
11. Department of Defense (DoD) USA. DoD Architecture Framework Version 1.5: Volume I: Definitions and Guidelines (cited 2008-03-19) (2008), `http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_I.pdf`
12. Engels, G., Hess, A., Humm, B., Juwig, O., Lohmann, M., Richter, J.-P.: Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten. dpunkt.verlag, Heidelberg (2008)
13. Frank, U.: Multi-perspective Enterprise Modeling (MEMO) – Conceptual Framework and Modeling Languages. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences, vol. 35, pp. 1258–1267 (2002)
14. Garg, A., Kazman, R., Chen, H.-M.: Interface Descriptions for Enterprise Architecture. Science of Computer Programming 61(1), 4–15 (2006)
15. Henderson, J.C., Venkatraman, N.: Strategic Alignment: Leveraging Information Technology for Transforming Organizations. IBM Systems Journal 38(2-3), 472–484 (1999)
16. IEEE. IEEE Std 1471-2000 for recommended Practice for Architectural Description of Software-intensive Systems (2000)
17. Jonkers, H., Goenewegen, L., Bonsangue, M., van Buuren, R.: A Language for Enterprise Modelling. In: Lankhorst, M. (ed.) Enterprise Architecture at Work. Springer, Heidelberg (2005)
18. Kirchner, L.: Eine Methode zur Unterstützung des IT-Managements im Rahmen der Unternehmensmodellierung. PhD thesis, Universität Duisburg-Essen, Logos, Berlin (2008)
19. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Heidelberg (2005)
20. Le, L.-S., Wegmann, A.: Definition of an Object-oriented Modeling Language for Enterprise Architecture. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005. HICSS 2005, p. 179c (2005)
21. Matthes, F., Buckl, S., Leitel, J., Schweda, C.M.: Enterprise Architecture Management Tool Survey 2008. Chair for Informatics 19 (sebis), Technische Universität München, Munich (2008)
22. Niemann, K.D.: From Enterprise Architecture to IT Governance – Elements of Effective IT Management. Vieweg+Teubner (2006)
23. OMG. Unified Modeling Language: Superstructure, Version 2.0, formal/05-07-04 (2005)

24. Pulkkinen, M.: Systemic Management of Architectural Decisions in Enterprise Architecture Planning. Four Dimensions and three Abstraction Levels. In: Proceedings of the 39th Annual Hawaii International Conference on System Sciences, 2006. HICSS 2006, vol. 8, p. 179c (2006)
25. Shewart, W.A.: Statistical Method from the Viewpoint of Quality Control. Dover Publication, New York (1986)
26. The Open Group. TOFAF Enterprise Edition Version 9 (2009)
27. van der Torre, L.W.N., Lankhorst, M.M., ter Doest, H.W.L., Campschroer, J.T.P., Arbab, F.: Landscape Maps for Enterprise Architectures. Technical report, Information Centre of Telematica Instituut, Enschede, Netherlands (2004)
28. van der Torre, L.W.N., Lankhorst, M.M., ter Doest, H.W.L., Campschroer, J.T.P., Arbab, F.: Landscape Maps for Enterprise Architectures. In: Dubois, E., Pohl, K. (eds.) CAiSE 2006. LNCS, vol. 4001, pp. 351–366. Springer, Heidelberg (2006)
29. Wittenburg, A.: Softwarekartographie: Modelle und Methoden zur systematischen Visualisierung von Anwendungslandschaften. PhD thesis, Fakultät für Informatik, Technische Universität München (2007)