Visual Roadmaps for Managed Enterprise Architecture Evolution

Sabine Buckl, Alexander M. Ernst, Florian Matthes, Christian M. Schweda
Chair for Informatics 19
Technische Universität München
E-mail: {buckls, ernst, matthes, schweda}@in.tum.de

***Abstract*—Managing the evolution of the Enterprise Architecture (EA) is a key challenge for modern enterprises. Current approaches to address this challenge focus on EA plans, indicating projected future states of the architecture. Nevertheless, these plans neglect the role of the information technology (IT) project, which actually performs the transformation of the current to a planned EA. In this paper, we account for the importance of IT projects as drivers of EA transformation by providing a viewpoint for roadmapping the development of the EA over time. Complementing, we further introduce a conceptual model, which explicates the information demands for such roadmap plans.**

I. Introduction & Motivation

Enterprise Architecture (EA) management is one of the major challenges of modern enterprises. It aims at aligning business and IT in order to optimize their interaction. Typically three central tasks are regarded to be part of an EA management process [19]:

- **Document** the current state architecture
- **Envision** a long-term perspective of the architecture
- **Plan** different intermediary architecture scenarios and decide on an architecture roadmap

The actual implementation of the selected architecture is often not regarded to be part of the EA management process itself [17], [11], but to be the duty of related enterprise-level management processes, such as project- or IT-portfolio management. Therefore, the respective implementation step is omitted in the above considerations.

In [1], an analysis comparing the tasks' *importance* and *degree of realization* from the perspective of EA practitioners was conducted. The core findings of the comparison are shorthanded here, as they indicate a deficit as experienced in practice, which we subsequently elaborate upon. Considering the perception of importance the three distinct tasks are ordered as *Envision*, *Plan*, and *Document*. Contrastingly, the perceived degree of realization leads to a different order of tasks, namely *Document*, *Envision*, and *Plan*. From this, one can derive that managing the *transformation* from a current EA to an envisioned EA via intermediary planned architectures is a challenge, many of todays enterprises have to face, but which is only "partially implemented". Hence, the EA is likely to evolve in an uncontrolled manner, i.e. the EA evolution is not managed. The term *transformation* respectively represents an intended architectural change, i.e. a *managed evolution*.

There are several possible causes for the aforementioned discrepancy, which are also discussed at the end of the respective analysis in [1]. These range from missing practical methodologies for architecture roadmapping, via inadequate representation of the concept of time in architectural models, to insufficient tool support for architecture planning. We regard these three causes to be equally important for the experienced discrepancy, although some further causes might play important roles. Creating a roadmap for the transformation of the EA has to incorporate a multitude of EA concepts, namely IT elements as well as the business support provided by them, but also *projects*, which actually change the architecture.

Currently, some approaches for managing the transformation of the EA exist, although many of these approaches concentrate on the architectural concepts and hence omit the projects. This leads to a description of an EA transformation[1] via a sequence of architectural *snapshots*, which indicate how the architecture is going to look like at a certain point in time. Architecture roadmaps, which display the change of the EA over a certain period of time, are not accounted for by most approaches to the topic as implemented in EA management tools or proposed in literature (cf. Section II). Hence, the approaches fail to provide a consistent view on the development of the architecture, which could only be gained by juxtaposing a set of respective landscape plans. This leads to the key challenge, which we want to address in this article – a challenge shorthanded in the question: **How do you visualize the development of the business support provided by (a part of) the application landscape over time and which information has to be gathered as well as maintained to do so?**

Therefore, we propose an information model for modeling the EA transformation process and provide exemplary visualizations for communicating transformation roadmaps to the respective stakeholders. The model and the technique proposed are extended versions of respective concepts as presented as part of a pattern approach to support EA management [3]. Before we introduce our model, we provide a short overview on the state-of-the-art in EA transformation modeling and visualization as found in literature in Section II. In Section III, we present the information model and elaborate on appropriate visualization techniques. A summary on the presented method is given in Section IV, eliciting questions for future research.

[1]For details on the approaches cf. Section II.

II. State-of-the-art

Defining a roadmap for the development of the enterprise in general and explicating the transformation of the application landscape in special is a major challenge of todays enterprises. Thereby, the term application landscape refers to the entirety of the business applications and their relationships to other elements, e.g. business processes in a company. Different approaches to model and visualize aspects related to transformational aspects exists, of which an extract is further discussed and related to our approach subsequently.

In [11], the transformation challenge is referred to as *planning the development of architectures*. Thereby, different time-related perspectives, e.g. the current version of a system, the planned system after a project has finished, and the future envisioned system that are relevant in this context are defined. In addition, detailed information on how to visualize transformation-related aspects using a so-called *landscape map* is given in [11], [20], [21]. Thereby, the landscape maps proposed pose a reference point in the future, which marks the point in time, the illustrated architecture should be realized. A visualization explicating the transformation tasks is omitted in [11]. Since the approach of [11] focuses on visualizations, only a simplified information model is provided, which gives an overview about the core concepts and main relationships.

Complementing the aforementioned approach, Ross et al. [18], [17] emphasize on the procedure of transformation. Therefore, they propose a high-level maturity model for EAs, which provides starting points for the design of the transformation process in order to enhance the used EA management process. However, neither an information model nor visualizations, which support the transformation process are discussed in the presentation of their approach.

Furthermore, [14] describes a procedure for EA transformation consisting of the process steps: *document, analyze, plan, act,* and *check*. These process steps can partially be mapped to the central tasks of EA management as presented in Section I. Whereas, we do not regard the implementation to be part of the EA management process, Niemann refers to this task in the act phase, further he emphasizes on the quality management by adding the phases analyze and check. In order to support the transformation process, [14] proposes the utilization of similar visualizations as proposed by [11], which he refers to as *product-process-application-matrix*. Although, these visualizations can be used to get a glimpse on the future architecture no information about the transformation steps is provided, in addition no information model to store the respective information is given.

Engels, et al. refer to the future planning process in [8] as the *managed evolution* of the application landscape. Thereby, not only a methodology how to develop the roadmap is given, which focuses on the projects and programs as central objects to transform the application landscape, but also different strategies to achieve the intermediate planning steps are provided. The methodology is complemented by a set of visualizations, which facilitate the transformation process, and a simplified information model linking the used concepts. Although these visualizations illustrate changes in the application landscape no transfer of business support can be explicated using the visualizations.

In addition to the various approaches existing in literature provided by practitioners as well as researchers, EA management tools form a second source providing methodologies, viewpoints, and conceptual models to address the challenge of EA transformation.

In [13] an extensive analysis of EA management tools is described, which was conducted in cooperation with 30 industry partners (among others Allianz Group IT, Siemens, O2 Germany, BMW Group, Deutsche Bank, Nokia Siemens Networks, Procter & Gamble, and Wacker Chemie). The survey analyzed the products of nine major players in the market of EA management tools: adaptive EAM (adaptive, Inc.), planningIT (alfabet AG), ADOit (BOC GmbH), EA/Studio (Embarcadero), ARIS IT Architect (IDS Scheer AG), MEGA Modeling Suite 2007 (MEGA International SA), Metastorm ProVision (Metastorm), System Architect (Telelogic AB[2]), and Troux (Troux Technologies, Inc.). Thereby, the survey pursued a threefold evaluation approach, relying on two distinct sets of scenarios complemented by an online questionnaire. We sketch some of the results of the survey, as they indicate that EA transformations are not fully sufficiently addressed by the tools.

The first set of scenarios focused on specific functionality, an EA management tool should provide, e.g. *creating visualizations, flexibility of the information model,* and *usability*, without connecting these functionalities to the execution of a typical EA management task. In addition, the EA management tools were further evaluated by the scenarios of the second set, which reflected tasks that have been identified as essential constituents of many EA management endeavors, e.g. *project portfolio management, IT architecture management*, and *business object management*. Two of the most prominent scenarios of the second part are the scenarios *landscape management* and *SOA transformation*, which are concerned with the managed evolution of the application landscape.

Within these scenarios, the integrated support of the EA management tools regarding the used methodology, visualization, and conceptual model is analyzed. Thereby, the provided support of the analyzed tools regarding the documentation and visualization of transfered business support vary widely [13]. The scenario *SOA transformation*, for instance, contains a visualizations similar to the one presented in Section III, which explicates the transfer of business support from one business application to another. Based on the evaluation results of the analyzed tools, a lack of support regarding the automated creation of such visualizations can be stated, as none of such tools was capable of generating such visualizations. In contrast some tools provided support regarding the generation of future architecture visualizations, which illustrate the architecture at a given time in the future.

[2]Telelogic has been acquired by IBM in 2007.

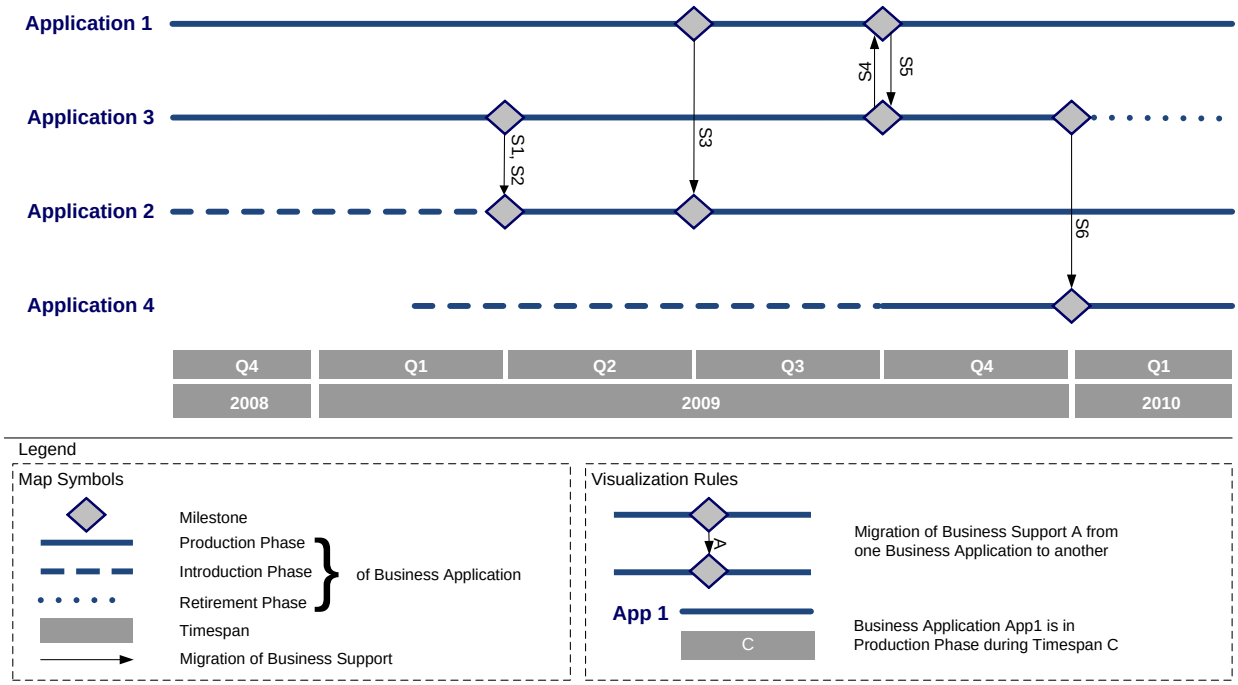

Fig. 1. Business Support Migration Roadmap Plan

Besides, the visualization of transformation related aspects, the conceptual information modeling is a central challenge. An in depth discussion about the handling of EA management tools regarding time-related aspects in the information models can be found in [2] and [6].

Subsuming the state of the art regarding EA transformation modeling and visualizations as presented in literature, a common understanding for the importance can be noticed. Contrastingly, the approaches presented vary widely regarding the coverage of explicating information models, visualizations, and methodologies supporting the transformation process.

III. EAM Patterns for addressing Transformation Management

In response to the challenge from Section I, we subsequently introduce an appropriate graphical *viewpoint* supporting EA transformation documentation. Further, we provide an object-oriented conceptual model, explicating the information demands, which have to be satisfied for creating a visual documentation (*view*) according to the aforementioned viewpoint. These two constituents of our response, viewpoint and information model, form reusable building blocks for an organization-specific EA management approach and are thus regarded EA management patterns in the terminology consistent with [5][3]. The terms *viewpoint* and *view* respectively are used as introduced in [10] – a standard for documenting the architecture of *software-intensive systems*.

[3]Due to length restrictions the two EA management patterns are presented in condensed form in this article and do not follow the structure proposed in [9].

Figure 1 presents an exemplary visualization of a roadmap plan. The underlying viewpoint, i.e. visualization technique, is inspired by the *Gantt-chart* as widely used in project management. This is especially true for the notion of the *milestone*, which indicates an important transformation event in the life-cycle of a business application. Other life-cycle information describing intervals, in which the application is *under development*, *operational*, or *in retirement*, are indicated using different line-styles, as *dashed*, *solid*, or *dotted*. Alternatively, color-coding could be used to differentiate the distinct life-cycle phases. Such notation would resemble the visualization of the respective information in the *time-interval maps* used in software cartography [12]. The migration of a relevant business support from one business application to another is indicated by a directed link between the respective milestones, which can be regarded as *start* and *end of business support provision* respectively. The link representing the migration of support is textually annotated with the name of the business support, which is transfered. To support the creation of concise models, multiple migrations in the same direction are represented in an aggregated manner as a single link annotated with a comma-separated list of business support names.

Milestone symbols and business application lifelines are strictly arranged alongside the temporal axis, which is depicted as x-axis at the bottom of each visualization. The links representing migrations of business support form an exception in this respect. Their time-dependencies are explicated by the attachment to respective milestone symbols. This exception is made to allow convenient depiction of multiple migrations originating from or ending at the same milestone, as a strict

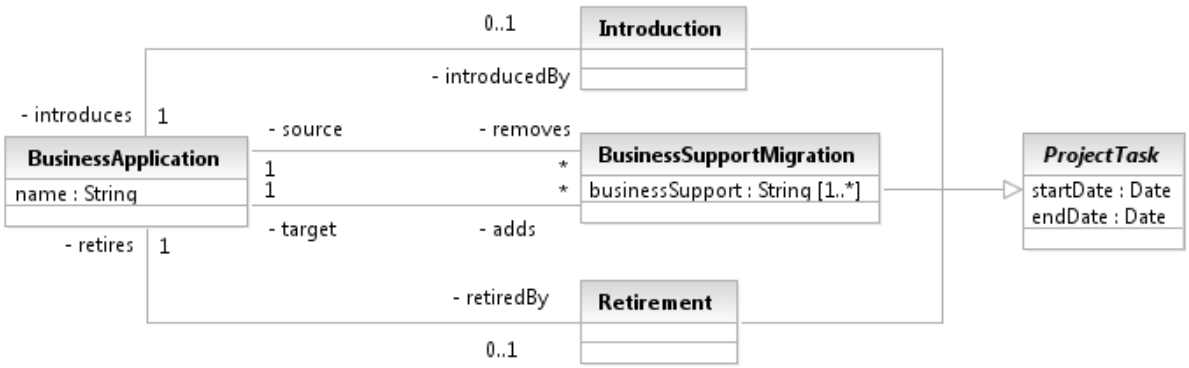

Fig. 2. Information Model

temporally alignment semantics would cause the respective graphical links to overlap. Further, this exception is made use of, when a business support exchange between two business applications at a certain point in time should be modeled. If a strict temporal notion of the links was employed in such a situation, the respective lines would overlap – creating the visual appearance of a double-headed link, for which no clear interpretation existed.

The viewpoint presented above is complemented by a respective information model fragment, providing the concepts necessary to describe a roadmap plan. These concepts are among others *business applications*, *business support*, and *business support migrations*. Additionally, the associations between the concepts have to be explicated in the information model to support consistent modeling. A UML class diagram is used to display the structure of the respective information model fragment (cf. Figure 2) – a subsequent glossary of terms contained therein is provided to explain the semantics of the respective classes, attributes[4], and associations:

- **Business Application** is a software system, which is part of a business information system of an organization. A business application thus provides support for at least one business process, i.e. infrastructure systems are not considered business applications in this context.
- **Business Support Migration** represents a project task migrating the provision of a specific business support from a **source** business application to a **target** one. The business support is considered fully migrated, once the date specified in **endsAt** has passed.
- **Introduction** is a specific type of project task introducing a distinct business application. After the date specified in **endsAt**, the associated business application is considered to be *in production*.
- **Project Task** is the abstract base concept for the different accomplishments of projects as considered in this pattern. Each project task spans a distinct period of time, enclosed by the two points in time **startsAt** and **endsAt**. The project tasks indicate the discrete events of change, connecting the different states of the EA to a chronological sequence.
- **Retirement** is a specific type of project task retiring a distinct business application. After the date specified in **startsAt**, the associated business application is considered to be *in retirement*.

The information model fragment is complemented with OCL constraints [15], which are used to enforce complex consistency requirements that have to hold in respective instance models. The first constraint imposes, that no project task may end, before it has started.

```
  context Project Task
inv: startsAt ≤ endsAt
```

A second constraint applies, demanding that a business support is always migrated between different business applications, i.e. that the source and target application of the respective migration must not be identical.

```
  context Business Support Migration
inv: source ≠ target
```

A third constraint concludes the consistency conditions, by imposing, that the respective source and target business applications of a business support migration must exist, i.e. they must be at least under development or in retirement.

```
  context Business Support Migration
inv: source.introducedBy = null ∨
(startsAt ≥ source.introducedBy.startsAt)
inv: source.retiredBy = null ∨
(startsAt ≤ source.retiredBy.endsAt)
inv: target.introducedBy = null ∨
(endsAt ≥ target.introducedBy.startsAt)
inv: target.retiredBy = null ∨
(endsAt ≤ target.retiredBy.endsAt)
```

The above constraints additionally consider the absence of an introducing or retiring project task, i.e. if none such task is associated, the respective business application is considered to have been *in production* ever since or remain *in production* ever after. Especially the first case may appear in practice, as

[4]The attributes' datatypes are defined as described in http://www.w3.org/TR/xmlschema-2 (as of 12-30-2008).

perhaps the introducing project task of a business application has not been modeled.

Based on the information stored according to the model introduced above, not only roadmaps of EA transformation can be created – it is further possible to create *diffs* of two EAs (or cutouts thereof), indicating the architectural delta between the state before and after the execution of a distinct sets of project tasks. This can be achieved as the sequence of project tasks forms a kind of *discrete change log* of an evolving EA.

The aforementioned information model puts a special emphasis on the creation of migration roadmaps for business support. It may nevertheless be easily integrated with an information model for describing snapshots of the application landscape and their dependencies to projects. An example for such an information model can be found in [2]. The point of integration is the relationship between the *project*, as contained in the model of [2], and the *project task*, as modeled above – a *whole-part* relationship. Thus, an integration can be performed easily, further fulfilling the promise of the *building-block* for an organization-specific EA management information model.

Another interesting augmentation to this article's information model can be derived from the discussion on time-dependence as found in [2]. Therein, planned start- or end-dates for projects (or constituting tasks respectively) are not considered fixed. In contrast, it is explicitly accounted for the possibility of a revision of a plan, i.e. for the plan dates to be changed. Thus, the periods of time spanned by the project tasks may undergo changes during EA planning – due to delays in project execution, an actual function transfer might be postponed. This aspect is not considered in the information model of this article, but could be included utilizing the aforementioned linkage to the *project* concept as introduced in the temporal EA management pattern for application landscape management, presented in [2].

IV. Resume and Outlook

The viewpoint presented in Section III was, as alluded to above, found to be used at a telecommunication company and in a slightly modified version at a re-insurance company. Nevertheless, no integrated information model was employed there to facilitate data collection and visualization generation for this aspects. In consequence, a practical validation of the information model introduced in this article has not been undertaken. Nevertheless, we expect no problems in generating the visualizations from data according to the information model – the respective generation algorithm has prototypically been implemented in a tool for visualizing EAs, which has been developed at our research group[5].

The information model and the viewpoint presented in this article are, as stated above, drawn from a pattern-based approach for EA management, see [3]. They thus are not limited to a single utilization context, but form re-usable EA management patterns, i. e. building blocks, which can be integrated into a multitude of organization-specific EA management approaches. In order to show the applicability of the approach described in this article, we plan to perform a case-study at a financial services provider. This provider currently has an organization specific EA management approach using a self-developed information model in place, which should be augmented to better support roadmap planning for EAs. This case study can additionally show, how the EA management pattern can be applied to evolve existing EA management approaches in companies – that would complement our findings on the usage of EA management patterns for creating a new information model as e. g. researched in [7], [16].

[5]For further information on the visualization tool see e. g. [4].

References

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

[2] S. Buckl, A. Ernst, F. Matthes, and C. M. Schweda. An Information Model for Landscape Management – Discussing Temporality Aspects. In P. Johnson, J. Schelp, and S. Aier, editors, *Proceedings of the 3rd International Workshop on Trends in Enterprise Architecture Research 2008 (TEAR 2008)*, Sydney, Australia, 2008.

[3] S. Buckl, A. M. Ernst, J. Lankes, and F. Matthes. *Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008)*. Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Munich, 2008.

[4] S. Buckl, A. M. Ernst, J. Lankes, F. Matthes, C. Schweda, and A. Wittenburg. Generating Visualizations of Enterprise Architectures using Model Transformation (Extended Version). *Enterprise Modelling and Information Systems Architectures – An International Journal*, 2(2), 2007.

[5] S. Buckl, A. M. Ernst, J. Lankes, K. Schneider, and C. M. Schweda. A Pattern based Approach for Constructing Enterprise Architecture Management Information Models. In *Wirtschaftsinformatik 2007*, pages 145 – 162, Karlsruhe, Germany, 2007. Universitätsverlag Karlsruhe.

[6] S. Buckl, A. M. Ernst, F. Matthes, and C. M. Schweda. An Information Model Capturing the managed Evolution of Application Landscapes. In *The* 21^{st} *International Conference on Advanced Information Systems (CAiSE09)*, Amsterdam, The Netherlands, 2009.

[7] T. Dierl. *Models, Methods, and Visualizations for Complicance Management*. Bachelor's thesis, Fakultät für Informatik, Technische Universität München, 2008.

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

[9] A. Ernst. Enterprise Architecture Management Patterns. In *PLoP 08: Proceedings of the Pattern Languages of Programs Conference 2008*, Nashville, 2008.

[10] IEEE. *IEEE Std 1471-2000 Recommended Practice for Architectural Description of Software-Intensive Systems*, 2000.

[11] M. Lankhorst. Introduction to enterprise architecture. In *Enterprise Architecture at Work*, Berlin, Heidelberg, New York, 2005. Springer.

[12] F. Matthes. Softwarekartographie. *Informatik Spektrum*, 31(6), pages 527-56, 2008.

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

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

[15] OMG. *Object Constraint Language (OCL) Available Specification, version 2.0 (formal/06-05-01)*. Object Management Group (OMG), 2006.

[16] K. Pflügler. *Evaluation and Extension of the EAM Pattern Catalog in a German Insurance Company*. Bachelor's thesis, Fakultät für Informatik, Technische Universität München, 2008.

[17] J. W. Ross. Creating a strategic it architecture competency: Learning in stages. *MIS Quarterly Executive*, 2(1), 2003.

[18] W. Ross, Jeanne, P. Weill, and C. Robertson, David. *Enterprise Architecture as Strategy*. Harvard Business School Press, Boston, Massachusetts, 2006.

[19] B. van der Raadt and H. van Vliet. Designing the Enterprise Architecture Function. In S. Becker, F. Plasil, and R. Reussner, editors, *Quality of Software Architectures (QoSA 2008)*, pages 103–118, Karlsruhe, 2008.

[20] L. W. N. van der Torre, M. M. Lankhorst, H. W. L. t. Doest, J. T. P. Campschroer, and F. Arbab. *Landscape Maps for Enterprise Architectures.* Technical report, Information Centre of Telematica Instituut, Enschede, Netherlands, 2004.

[21] L. W. N. van der Torre, M. M. Lankhorst, H. W. L. ter Doest, J. T. P. Campschroer, and F. Arbab. Landscape Maps for Enterprise Architectures. In E. Dubois and K. Pohl, editors, *The* 18^{th} *International Conference on Advanced Information Systems Engineering. (CAiSE 2006)*, volume 4001 of *Lecture Notes in Computer Science*, pages 351–366. Springer, 2006.