Roadmaps for Enterprise Architecture Evolution

Alexander M. Ernst, Alexander W. Schneider
Chair for Informatics 19
Technische Universität München
eMail: {ernst, schneial}@in.tum.de

Abstract: Managing the enterprise architecture (EA) is a major challenge of every larger modern enterprise. Aligning business and IT in order to optimize their interaction is the main goal of EA management. The first step of an EA management approach typically is to analyze and document the as-is-situation. Subsequently, the second step has to be the management of the EA evolution in order to achieve the desired target EA. Although IT projects guide this evolution by implementing the transformations from a current to an envisioned architecture, they are considered sparsely by EA approaches. Therefore, this article presents a methodology pattern (M-Pattern) describing a process including all necessary tasks and associated roles, which help to cope with the challenge of EA evolution management.

1 Introduction

Modern enterprises face the challenge to survive in an ever changing environment. In order to support the transformation of the enterprise according to new business demands, legal regulations, or changing market situations a holistic perspective on the enterprise architecture (EA) has to be taken. Thereby, EA is understood as the *fundamental conception of the system [enterprise] in its environment embodied in its elements, their relationships to each other and to its environment, and the principles guiding its design and evolution* [Int07]. Managing the EA is one of the major challenges of modern enterprises. It aims at aligning business and IT in order to optimize their interaction.

Documenting and managing the EA is an advanced topic, as the application landscape, which is part of the EA, often includes a few hundreds up to a few thousand business applications and their interconnections. Thereby, managing the EA is a task, that has to be executed as the need for a flexible IT is an integral concern of most companies. Another reason for the importance of EA management are regulations like e.g. the *Sarbanes Oxley Act* (SOX) [tC02], which determine the information a company has to have available about its EA.

This article includes patterns which are part of the *EAM Pattern Catalog*, a *pattern language for enterprise architecture management* [BEL+07, BELM08, BEL+08, Ern08, BEMS09b, Ern09], which uses a pattern based approach to EA management. The complete *EAM Pattern Catalog* is available online at http://eampc-wiki.systemcartography.info/ [Cha09] and currently includes 164 EAM patterns[1]. The intention behind this article is to further extend the existing *EAM Pattern Catalog* by a methodology pattern (M-Pattern), documenting best practices for managing the evolution of an EA, in order to advance the EAM pattern language.

The *EAM Pattern Catalog* introduces four different types of patterns:

M-Patterns specify a methodology to address management problems in a stepwise manner. The procedures defined by the M-Pattern can be very different, ranging from visualizations and group discussions to more formal techniques as e.g. metrics calculations [LS08]. M-Patterns explicate the methodologies in order to complement activities carried out in an ad-hoc manner or relying on implicit knowledge with activities carried out more systematically.

V-Patterns provide visualizations like diagrams, reports, etc., which are practically proven to be adequate to address problems in EA management. The data required to produce the visualization is documented in one or more I-Patterns.

I-Patterns supply best-practice information model fragments, including definitions and descriptions of the used concepts, which can be used to collect information to address a certain problem in EA management.

Anti patterns document typical mistakes made in the context of EA management, and provide revised solutions in order to support the pattern user to prevent these pitfalls.

The rest of this section gives a short overview about the intended audience, the nature of patterns, and an overview about included EAM patterns.

1.1 Intended Audience

This article and the herein included patterns are intended for people concerned with planning the enterprise architecture evolution, people who have to manage a project portfolio, and people responsible for application life cycles.

1.2 Pattern-based approach to EA management

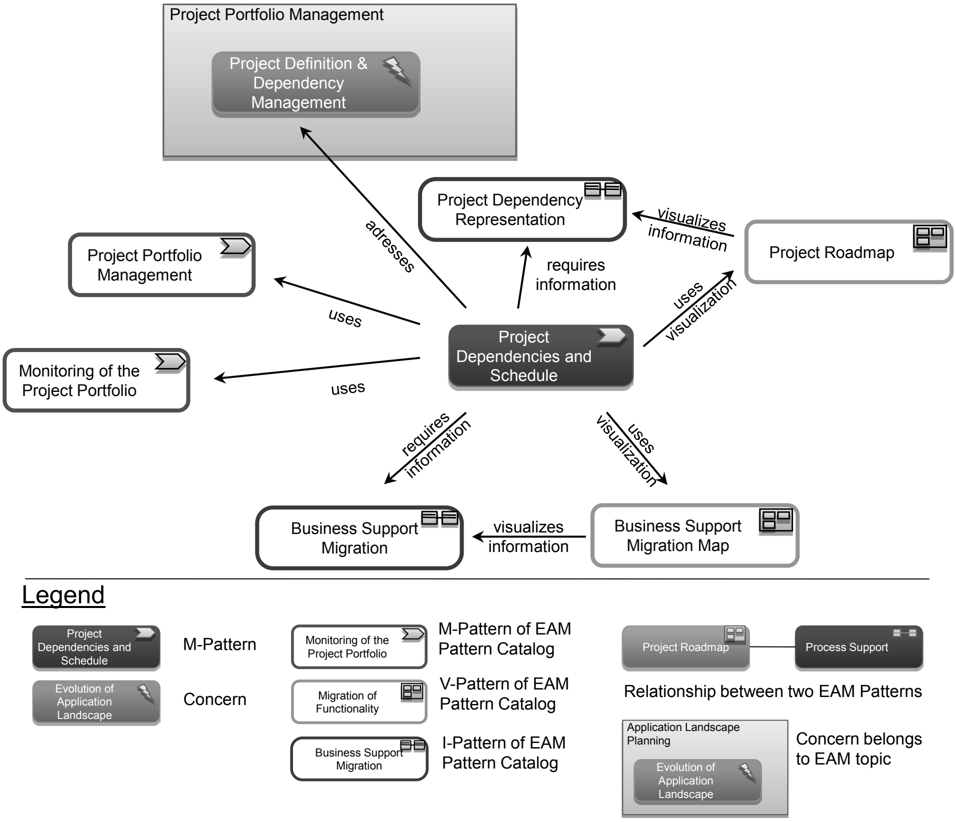
Patterns in general, and so do EAM Patterns, provide well-known and proven practice solutions for recurring problems in a given context. At first glance it might appear that there is no benefit in documenting such solutions. But the opposite is true because patterns are a valuable source of information for novice practitioners as well as a source for inspiration for experienced ones. This is one of the reasons for the adoption of the pattern concept in various disciplines. Ernst [Ern09] presents an in-depth introduction to the pattern-based approach to EA management.

[1] For a detailed explanation of the concept of EAM patterns refer to [Ern08, Ern09].

Typically, a pattern has to be found in practice at least three times to prove that it addresses a recurring problem and provides a proven practice solution. The EAM Pattern described in this article could, at the time of publication, be found in one company. Nevertheless, it can be expected that it holds for other contexts as well. Running and managing projects is a common task for companies. If there is more than one project accomplished at once, the described pattern can be used because it does not depend on the projects' topics.

1.3 Overview of included EAM Patterns

This article includes the M-Pattern EA DEPENDENCY GUIDED PROJECT MANAGEMENT. Figure 1 shows EA DEPENDENCY GUIDED PROJECT MANAGEMENT and its relationships to other EAM Patterns of the *EAM Pattern Catalog*. The figure also shows the concern addressed by EA DEPENDENCY GUIDED PROJECT MANAGEMENT and the topic the concern belongs to.



Project Portfolio Management
Project Definition & Dependency Management
adresses
Project Dependency Representation
visualizes information
Project Roadmap
requires information
Project Portfolio Management
uses
Project Dependencies and Schedule
uses visualization
Monitoring of the Project Portfolio
uses
requires information
uses visualization
Business Support Migration
visualizes information
Business Support Migration Map
Legend
Project Dependencies and Schedule
M-Pattern
Evolution of Application Landscape
Concern
Monitoring of the Project Portfolio
M-Pattern of EAM Pattern Catalog
Migration of Functionality
V-Pattern of EAM Pattern Catalog
Business Support Migration
I-Pattern of EAM Pattern Catalog
Project Roadmap
Process Support
Relationship between two EAM Patterns
Application Landscape Planning
Evolution of Application Landscape
Concern belongs to EAM topic

Figure 1: EAM Pattern Map

The following description gives an overview about the not included EAM Patterns:

Project Dependency Representation provides a structure for organizing information about projects and their interrelationships.

Project Roadmap provides a way to visualize information about projects, their progress, and their relations to each other. This includes project phases, milestones, and dependencies between projects.

Business Support Migration Map provides a way to visualize which application will take over functionality of other applications.

Business Support Migration provides a structure for organizing information about the EA evolution including the introduction and retirement of applications and the migration of business supports.

Monitoring of the Project Portfolio is concerned with the monitoring of the project portfolio.

Project Portfolio Management is used to analyze the project portfolio concerning the defined strategies.

Additional detail about those EAM patterns can be found in [Sch09].

2 EA dependency guided Project Management

The objective of EA DEPENDENCY GUIDED PROJECT MANAGEMENT is identify all important dependencies between current and planned projects from an EA management point of view, in order to schedule projects and react on delays timely.

2.1 Example

The MRAm project was considered to roll out three newly developed or adapted business applications at a specific organizational unit. The custom-built systems used at this location should be replaced by global homogeneous applications. The first application was called FAB and was responsible for contract initiations with customers. The second was called RM-NL and was responsible for contract administration. The third was called FS-RI and was responsible for the generation of the annual report of the company. In this order data included in contracts had to pass these systems. Because of the high complexity of the supported business processes, each application was developed in a separate project which consisted of several sub-projects. Of course, project managers were aware that their application has to import data from another application. But initially, they did not exactly define the data because they did not regard this as a major issue and postponed it. After applying EA DEPENDENCY GUIDED PROJECT MANAGEMENT, all project managers described their dependencies in detail and discovered that the contracts to be administrated by these applications are very individual and cannot be standardized. Therefore, the refinement of the application's interfaces caused a huge delay in all three projects.

2.2 Context

You work in a company running multiple software development projects in parallel and you have to manage their dependencies from an EA management point of view. A multi project management approach might already be established in your company focusing typical tasks like the effective planning, evaluation, and control of strategic projects [Kun05]. The individual project schedules should not be determined by EA management beforehand. All projects to be regarded are already prioritized and have a fixed budget. For each project a project plan, e.g. according to the project management body of knowledge [Dun96], is in use.

2.3 Problem

Which tasks have to be performed to ensure an effective inter-project dependency identification to foster EA evolution management?

The following *forces* influence the solution:

Strict dependency management versus laissez-fair Who is responsible for identified dependencies? Are projects allowed to handle their dependencies on their own or is there a need for a steering committee?

Completeness of dependencies versus effort How can you ensure the completeness of identified dependencies but keep the needed effort slight?

One-time versus continuous approach Should you manage changing project requirements and schedules continuously or by a one-time approach?

All-embracing versus selective approach How many projects can be regarded and how much information is appropriate to ensure manageability? Is it necessary to manage all projects or is it sufficient to consider critical projects?

High versus low transparency Does the constraint of public status reporting result in information withholding? Are project managers willing to provide the needed information or is there certain governance needed?

2.4 Solution

EA DEPENDENCY GUIDED PROJECT MANAGEMENT follows a four-step approach, which is shown in Figure 2. The steps **Dependency identification** and **Business support migration** form the analysis phase, which regards the projects in the portfolio context and their impacts to the EA. The second phase, covering the steps **Joint scheduling** and **CPM recomputing**, is an implementation phase which applies insights of previously executed phases.

There are at least three conceivable events that are considered to occur likely and result in a new iteration of EA DEPENDENCY GUIDED PROJECT MANAGEMENT:

- *New requirements from business emerge* due to changing processes or new strategies demanding for new projects to be set up.
- *Technical reasons* can require additional projects for new applications or infrastructure like operating systems, e.g. if the vendor's time of support is over.
- *EA management decisions* can result in new projects, for example to consolidate the application landscape.
- *Mergers and acquisitions* result in a program of integration projects which also have to be regarded from an EA point of view.

If one of the previously mentioned events occurs and a new iteration should begin all four steps have to be performed. In the following the respective steps are presented in detail:

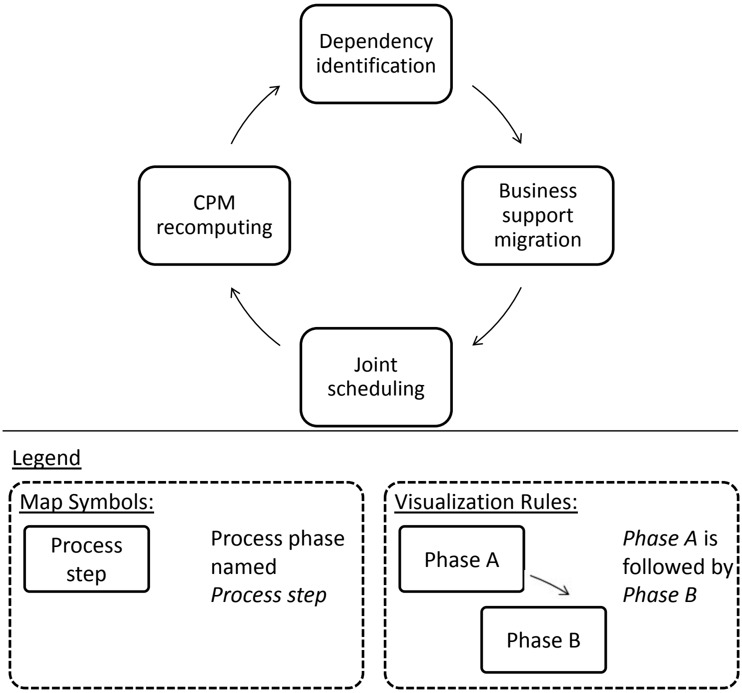

Dependency identification
Business support migration
Joint scheduling
CPM recomputing
Legend
Map Symbols:
Process step
Process phase named *Process step*
Visualization Rules:
Phase A
Phase B
Phase A is followed by *Phase B*

Figure 2: EA DEPENDENCY GUIDED PROJECT MANAGEMENT process

Dependency identification This step is as essential as challenging. The responsible project manager will find some dependencies during his planning activity, for example, that test data is needed from another project. Some dependencies like resources required at the same time might be identified by the multi project management. But there are also dependencies which are not that easy to identify. Therefore, this activity has to be performed separately by *project managers* and *IT architects* together. At least, they should regard three types of dependencies:

- **Organizational dependencies:** They result from the organizational structure and context of the company. For example, two projects are going to roll out their application at the same organizational unit at the same time and both estimate that the rollout will take the whole weekend. The two IT employees at that organizational unit might not be able to handle this. As a result, the two projects depend on one another because only one can rollout at this time. Another reason might be that the application which should deliver data to the project is not able to do this because it computes the annual accounts.
- **Interfaces between applications:** If a project uses data or functionality of another application currently developed, it depends on that projects ability to deliver.
- **Technical dependencies:** Usually, an application uses other software consid-

ered as infrastructure like, for example, databases or enterprise resource planning systems. A project depends on all other projects introducing or changing one of the used infrastructure systems.

- **Rollout dependencies:** In complex business processes applications often have to interact with each other. If more than one of them has to be changed, dependencies like "*has to be rolled out before X*" or "*has to be rolled out together with X*" can occur.
- **Retirement dependencies:** More often than not, new applications replace one or more old applications. If more than one new application is considered to retire such a legacy system, these new applications depend on each other, because they all need to be implemented before the retirement can begin.

In general, it is important, that all projects affected by a dependency formally commit to it. In addition, it is not enough to identify a dependency. Every dependency has to be described in detail so everybody is aware of all resulting impacts.

Business support migration This step might reveal additional dependencies and ensure the right handling of legacy systems, because new applications are not built on a greenfield, it is necessary to determine all applications or parts of them which are retired by the currently developed application. If a legacy system can be retired immediately, the retirement process might also require another project. If only a part of the functionality is adopted, it has to be decided which application adopts the other parts if they are still needed. If some functionality is not needed any more, this fact also has to be documented.

Joint schedule Based on the initial project prioritization the new projects can now be included in the overall schedule (roadmap). If the new projects have any impact to other projects, these impacts have to be communicated.

Critical Path Method (CPM) recomputing After a roadmap update, the critical path through all managed projects needs to be recomputed. That is the sequence of project tasks, which do not have any slacktime so a delay will cause the entire project or depending projects to be late [KW59]. The results might then have additional impacts to projects, which have to be applied and communicated. A method for critical path computing in a project portfolio environment is described in [MC10].

2.5 Implementation

In order to implement EA DEPENDENCY GUIDED PROJECT MANAGEMENTin your company, the process steps described in the solution section need to be assigned to different roles. Table 1 shows a RACI chart [Val10] which clarifies which role is involved in each activity. Overall there are five roles needed:

CIO The Chief Information Officer is a member of the executive board or reports directly to it and is in charge for the information technology (IT) of the company.

Project portfolio management board (PPM board) The project portfolio management board is responsible for tailoring and approving projects. They also prioritize projects according to the business or IT strategy and communicate rescheduling impacts.

Project manager A project manager is in charge for one or more projects of the project portfolio. He is responsible for the initial project data like phases and milestone dates.

IT architect The IT architect has the overview about the whole application landscape and is responsible for the consistent retirement of legacy systems and the adherence to the application landscape vision.

Business unit The business unit includes all people of a department. The business unit is involved in the business process supported by the application developed during the specific project. It can also be the project proposer.

	PPM board	**IT architect**	**Project manager**	**Business units**	**CIO**
Dependency identification	I	R/A	R		
Business support migration		R/A	C	C/I	
Joint scheduling	R/A			C/I	I
CPM recomputing	R/A				I

Table 1: RACI chart for EA DEPENDENCY GUIDED PROJECT MANAGEMENT

The different roles assigned in the RACI chart above are defined as follows [Val10]:

(R) = **Responsible** - owns the problem / project

(A) = to whom ”R” is **Accountable** - who must sign off (Approve) on work before it is effective

(C) = to be **Consulted** - has information and/or capability necessary to complete the work

(I) = to be **Informed** - must be notified of results, but need not be consulted

2.5.1 Known Uses

EA DEPENDENCY GUIDED PROJECT MANAGEMENT is in use at Munich Re.

2.6 Consequences

Strict dependency management versus laissez-fair A strict dependency management will clarify accountabilities for dependencies but also raises costs. In order to achieve a holistic overview of dependencies, a relatively strict dependency management including a formal process of commitment should be used. The steering committee is accountable to ensure the commitment of project managers to identified dependencies. Of course, this approach causes some additional effort, but it is necessary to ensure that project managers address dependencies within their planning.

Completeness of dependencies versus effort Achieving a complete list of dependencies will minimize the risk of undiscovered dependencies but raises costs at the beginning of the project. Undiscovered dependencies can cause greater cost and delays at the end. Achieving a complete list of dependencies among projects requires an intensive communication between IT architects, project managers, and business units.

One-time versus continuous approach From an EA management point of view, which regards long-term perspectives, an instant notification of schedule changes is not needed. An interval of about one month seems appropriate for the consolidation of project changes to the joint schedule.

All-embracing versus selective approach Because even non-critical or very small projects can be affected by the critical projects, they also have to be included in a joint schedule. But if a project does not have any dependencies to other projects, including the rollout, it can be removed from the overall schedule in order to reduce its complexity.

High versus low transparency The transparency achieved by this solution is high, but also raises costs. Thus, the company gets a better overview about its projects. Because the process depends on data about individual project schedules, it is important to convince project managers about their benefits of delivering this data and so prevent information withholding.

2.7 See Also

When using EA DEPENDENCY GUIDED PROJECT MANAGEMENT also the results of PROJECT PORTFOLIO MANAGEMENT and MONITORING OF THE PROJECT PORTFOLIO can be of interest and should be considered, because they are also concerned with projects and their relations. The implementation of this pattern can be supported using PROJECT PORTFOLIO ROADMAP, BUSINESS SUPPORT MIGRATION MAP, PROJECT DEPENDENCY REPRESENTATION, and BUSINESS SUPPORT MIGRATION [Sch09].

3 Acknowledgment and Outlook

This section includes acknowledgments to the people who supported the creation of this article and gives an outlook to the next steps in the development of the EAM pattern approach.

3.1 Acknowledgments

We want to thank all participants of the writer's workshop of PEAM 2010 and especially our shepherd Wolfgang Keller for the time they spent reading, commenting, and discussing this article. In addition we want to thank Ms. Dr. Walter and Mr. Perkens-Golomb for their support during the EAM Pattern documentation.

3.2 Next Steps in EAM Pattern Approach Development

The *EAM Pattern Catalog* is available at http://eampc-wiki.systemcartography.info/, based on the results of an extensive online survey, some articles on EAM patterns, see e.g. [Ern08, BEK$^+$09, LFB$^+$09, MJBS09, BEMS09a, BEMS09b], and a Ph.D. thesis [Ern09]. In order to improve the current version and to further exploit the advantages of patterns in EA management, an excerpt of the *EAM Pattern Catalog* had been included in this document to be discussed in the pattern community.

References

[BEK$^+$09] S. Buckl, A. Ernst, H. Kopper, R. Marliani, F. Matthes, P. Petschownik, and C. M. Schweda. EAM Pattern for Consolidations after Mergers. In *SE 2009 - Workshopband*, pages 67–78, Kaiserslautern, 2009.

[BEL$^+$07] Sabine Buckl, Alexander M. Ernst, Josef Lankes, Kathrin Schneider, and Christian M. Schweda. A pattern based Approach for constructing Enterprise Architecture Management Information Models. In *Wirtschaftsinformatik 2007*, pages 145 – 162, Karlsruhe, Germany, 2007. Universitätsverlag Karlsruhe.

[BEL$^+$08] Sabine Buckl, Alexander Ernst, Josef Lankes, Florian Matthes, and Christian M. Schweda. Enterprise Architecture Management Patterns – Exemplifying the Approach. In *The 12th IEEE International EDOC Conference (EDOC 2008)*, Munich, 2008. IEEE Computer Society.

[BELM08] Sabine Buckl, Alexander M. Ernst, Josef Lankes, and Florian Matthes. Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008). Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany, 2008.

[BEMS09a] Sabine Buckl, Alexander Ernst, Florian Matthes, and Christian Schweda. Enterprise Architecture Management Pattern for EA Visioning. In *Proceedings of EuroPLoP 2009*, 2009.

[BEMS09b] Sabine Buckl, Alexander M. Ernst, Florian Matthes, and Christian M. Schweda. How to make your enterprise architecture management endeavor fail! In *Pattern Languages of Programs 2009 (PLoP 2009), Chicago*, 2009.

[Cha09] Chair for Informatics 19 (sebis), Technische Universität München. EAM Pattern Catalog Wiki. http://eampc-wiki.systemcartography.info (cited 2009-11-06), 2009.

[Dun96] W.R. Duncan. A Guide to the Project Management Body of Knowledge: PMBOK Guide, 1996.

[Ern08] Alexander Ernst. Enterprise Architecture Management Patterns. In *PLoP 08: Proceedings of the Pattern Languages of Programs Conference 2008*, Nashville, USA, 2008.

[Ern09] Alexander M. Ernst. *A Pattern-Based Approach to Enterprise Architecture Management*. PhD thesis, Technische Universität München, München, Germany, 2009. (in submission).

[Int07] International Organization for Standardization. ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems, 2007.

[Kun05] Christian Kunz. *Strategisches Multiprojektmanagement Konzeption, Methoden und Strukturen*. PhD thesis, Universität Bamberg, Wiesbaden, Germany, 2005.

[KW59] James E. Kelley, Jr and Morgan R. Walker. Critical-path planning and scheduling. In *IRE-AIEE-ACM '59 (Eastern): Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference*, pages 160–173, New York, NY, USA, 1959. ACM.

[LFB+09] Armin Lau, Thomas Fischer, Sabine Buckl, Alexander M. Ernst, Florian Matthes, and Christian M. Schweda. EA Management Patterns for Smart Networks. In *SE 2009 - Workshopband*, pages 79–90, 2009.

[LS08] Josef Lankes and Chrisitian M. Schweda. Using Metrics to Evaluate Failure Propagation and Failure Impacts in Application Landscapes. In *Multikonferenz Wirtschaftsinformatik*, Berlin, Germany, 2008. GITO-Verlag.

[MC10] Wolfram Müller and Jens Coldewey. Critical Chain Project Management. *Object Spectrum*, 2010.

[MJBS09] Christoph Moser, Stefan Junginger, Matthias Brückmann, and Klaus-Manfred Schöne. Some Process Patterns for Enterprise Architecture Management. In *Software Engineering 2009 – Workshopband*, pages 19–30. Lecture Notes in Informatics (LNI), 2009.

[Sch09] Alexander Schneider. Roadmaps for Enterprise Architecture Evolution. Bachelor's Thesis, Technische Universität München, München, Germany, 2009.

[tC02] 107th Congress. Sarbanes-Oxley Act of 2002 (Public Law 107204). http://www.sec.gov/about/laws/soa2002.pdf, 2002.

[Val10] Value Based Management.net. RACI Model. http://www.valuebasedmanagement.net/methods_raci.html, 2010.