Teaching Enterprise Architecture Management
A Practical Experience

Sabine Buckl, Thomas Dierl, Florian Matthes, Christopher Schulz, Christian M. Schweda

Software Engineering for Business Information Systems (sebis)
Ernst Denert-Stiftungslehrstuhl
Chair for Informatics 19
Technische Universität München

Boltzmannstraße 3, 85748 Garching b. München, Germany

wwwmatthes.in.tum.de

About sebis

sebis is the chair for *Software Engineering for Business Information Systems* at the Institute for Informatics of the Technische Universität München. sebis has been established in 2002 with funding of the Ernst Denert-Stiftung and is headed by Professor Dr. Florian Matthes. The main research areas of *sebis* are:

- Software Cartography: Development of multi-faceted and formal models that help to manage (plan, build, operate, optimize) complex software application landscapes consisting of hundreds or thousands of information systems.
- Innovative technologies and software architectures for enterprise information and knowledge management (enterprise solutions, groupware and social software).
- Domain-specific and reflective languages and models for families of business applications.

sebis is using software engineering methods (model construction & abstraction, analysis & design, construction & evaluation) and is working in close relationship with industrial partners and with organizations from the public sector.

Professor Matthes puts particular emphasis on the knowledge transfer from academia to industry. For example, he is co-founder of CoreMedia AG, of infoAsset AG, and of 20six Web log services AG, which at present employ a total of approx. 150 employees.

Copyright

Trademarks

Abstract

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. Although a high demand for experts in this context exists, experiences and proven practices in teaching EA management, e.g. in university contexts are scarce. This misalignment can be explained on the one hand by the variety of problems addressed in the context of EA management and on the other hand by the challenge of reproducing organizational contexts, e.g. the different problems an enterprise architect has to deal with in the specific setting of an enterprise, like the existing culture, in a laboratory environment.

This report describes the structure, results, and lessons learned of a method to teach EA management based on experiences gained in a course at the Technische Universität München in the summer term 2009. The course follows a twofold approach consisting of a lecture, which provides theoretical knowledge in the area of EA management, and so-called "mini projects", which were conducted by student teams in cooperation with industry partners. The objective of the course is to equip students both with theoretical knowledge and a practical understanding of challenges arising in the context of managing complex EAs. Thus, the course builds on existing best practice solutions, EA management patterns, which are documented, reused, or developed to solve the problem of the industry partner.

Contents

CHAPTER 1

Introduction

While an increasing demand for experts in enterprise architecture (EA) management in practice exists, approaches and practical experience in teaching EA management are scarce. The experienced gap may be explained one the one hand by the existing plurality of EA management approaches and on the other hand by the diversity of practical problem statements, which are specific for the involved enterprise and its context and culture. Whereas several consultancies offer workshops for interested practitioners, which last for several days, universities have not yet adapted their curriculum to this demand of practice.

The challenge of teaching EA management is to find the right balance between theoretical foundations and practical experience. The difficulties arising in this context do not only result from the complexity of the management subject – the EA – but from the involvement of various stakeholders[1]. These stakeholders with business or IT background have different and often conflicting interests in the system, which need to be appropriately accounted for during EA management. Similarly, 'political' circumstances in an enterprise may negatively affect the prospects of the EA management endeavor.

A first attempt to address the aforementioned challenges and equip graduates with theoretical and practical experience in EA management was performed by the computer science department of the Utrecht University[2]. Koning et al. use a defined method with a given set of problems to be addressed, complemented by predefined viewpoints used to describe the EA of an industry partner. While the approach can be applied in many different contexts, due to its general procedure, it does nevertheless not well account for the aforementioned typical EA management challenges and the specificity of EA management concerns in the participating enterprises. Similarly, the need to adapt the method to the enterprises context and culture is neglected. Hence, we propose a different teaching approach, which is based on EAM patterns[3] suitable to address the specific concerns of each participating industry partner. Furthermore, the pattern-based methods are adaptable to the context and culture of the industry partner under consideration.

[1] A *stakeholder* is an individual, team, or organization with interests in, or concerns relative to a system (cf. ISO Std. 42010).

[2] Koning, H. Bos, R., and Brinkkemper, S.:*A Lightweight Method for the Modelling of Enterprise Architectures – Introduction and Usage Feedback* In: Pre-Proceedings of the 3rd Workshop on Trends in Enterprise Architecture Research 2008, pages 75-90, Sydney, Australia, 2008.

[3] see http://eampc-wiki.systemcartography.info

1.1 Structure and Outline of the lecture

The following Chapters 2-11 represent student papers from our lecture *Enterprise Architecture Management*, a master course of the Chair for Software Engineering for Business Information Systems[4] at Technische Universität München, Germany. The course was first offered in 2009 and consists of a lecture providing the theoretical background and so-called "mini projects", which were conducted by student teams in cooperation with industry partners. The objective of the course is to equip students both with theoretical knowledge on EA management and a practical understanding of challenges arising in the context of complex EAs. To achieve the latter, the students provided with means to solve small real world EA management tasks in practice.

In the lecture, the students are given an overview about the state-of-the-art in EA management, including prominent frameworks like TOGAF and Zachmann. In addition, theoretical foundations like models, metamodels, architectures in general and IT governance are presented. Based on these foundations, the EA management function and their interplay with other enterprise-level management functions[5], e.g. enterprise-wide demand management, project portfolio management, and strategies & goals management, is introduced (see Figure 1.1), followed by an in-depth discussion of the main activities of the EA management function: envisioning, documenting, planning, analyzing, and enforcing. Completing the theoretical foundation the EA management pattern catalog (EAMPC)[6] a collection of proven best-practice solutions is presented.

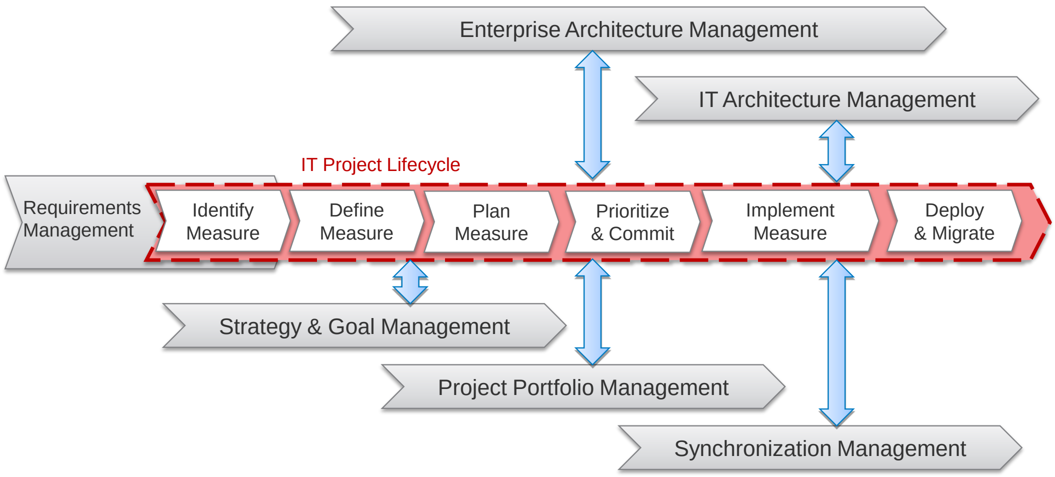

Figure 1.1: Interplay of EA management function with other management functions

In order to apply the theoretical knowledge gained in the lecture, the students were given the chance to solve a real-world EA management problem at an industry partner during a so-called mini project. The industry partners are global acting enterprises with a subsidiary in Munich, which are interested in establishing or enhancing the EA management function. In preparation of the lecture, well-defined EA management problems with these industry partners are identified in close cooperation. These EA management problems reflect the maturity of the different enterprises in terms of their EA management and emphasize on the enterprise specificity of the topic. The mini projects took three month and started in May 2009 with an initial presentation, in which each student team introduced its specific

[4]see wwwmatthes.in.tum.de

[5]see e.g. Wittenburg, A; Matthes, F.; Fischer, F. and Hallermeier, T.: *Building an integrated IT governance platform at the BMW Group.* In: International Journal on Business Process Integration and Management, 2(4), 2007.

[6]see http://eampc-wiki.systemcartography.info

EA management problem to the fellow students and industry partners. In order to solve the specific problems, EA management patterns as contained in the EAMPC should be utilized or adapted. Nevertheless, as some EA management problems of the participating enterprises had yet not been addressed by EA management patterns, new patterns were developed by some of the student teams.

To successfully complete the mini projects, the student teams had to provide three deliverables:

Initial presentation: Two weeks after the project had started, the student teams had to give an initial presentation, which introduced the respective problem to the other student teams and the participating industry partners. The objective of the presentation was to illustrate the idea of the chosen solution and discuss it with other student teams with possibly related problem statements. Furthermore, the industry partners were asked to participate in the discussion to provide feedback to the proposed solution from a practitioners point of view and to meet other practitioners in the field of EA management.

Final presentation: Two months after the initial presentation, the final presentations took place. The objective of each team's presentation was not only to present the developed solutions but also to discuss experienced pitfalls, surprises, and lessons learned from the student team perspective as well as the industry partner perspective.

Summary: Four weeks after the final presentation, the students had to submit a written summary of their mini project. These summaries could be formulated as EAM pattern, providing a reusable solution, or as a case study describing the used approach.

1.2 Project partners and cases

The mini projects accompanying the lecture were conducted in close cooperation with nine industry partners. The setting at each industry partner differs regarding the scope and reach of the established EA management. Whereas the mini projects can be seen as the initial EA management endeavor for some industry partners, others already had an established approach, which was documented, assessed, or improved in the mini project. The different scope of the projects becomes obvious, when the areas of interest of the specific mini projects are examined. Figure 1.2 gives an overview on the layers and cross-functions as introduced in the lecture and illustrates, which different aspects were relevant for the ten mini projects[7] illustrates the layers and cross functions of a holistic EA management perspective and positions the mini projects.

Figure 1.2: The layers and cross functions of a holistic EA perspective

[7]One industry partner hosted two mini projects.

Table 1.1 presents an overview on the mini projects conducted in 2009 and details on the type of summary as mentioned before:

Chapter	Title	Type of Summary
2	Cost reduction analysis for an application landscape in a corporate environment	EAM pattern
3	EAM patterns for the project decision process at a financial service provider	EAM pattern
4	eLearning in medicine	Case study & EAM pattern
5	Modeling and Documentation of IntegraTUM's System Landscape	Case study
6	Development of candidate patterns for the activities "Envision EA" and "Plan EA" from the Enterprise Architecture Management Process	EAM pattern
7	Comparing the Usage of a Content Management System in the Radio and Television Archives of a Public Broadcasting Company	EAM pattern
8	EA Management Patterns for Analyzing of Business Applications	EAM pattern
9	Analysis of a risk management system with its components and interfaces at a financial service provider	Case study
10	Prototypic Requirements Elicitation and Evaluation of Application Information Needs	EAM pattern
11	Analysis and development of an Architecture Check Process embedded into Enterprise Architecture Management	Case study

Table 1.1: Mini projects conducted in 2009

The mini projects accompanying the lecture were conducted in cooperation with nine industry partners. We want to thank the following companies for participating and for providing the possibility for students to experience real world EA management cases:

- Allianz Deutschland
- Bayerischer Rundfunk
- FIDUCIA
- Leibniz Rechenzentrum
- Siemens Financial Services
- Technische Universität München
- UniCredit Global Information Services
- Wacker Chemie

CHAPTER 2

Cost reduction analysis for an application landscape in a corporate environment

Maximilian Barnert (maximilian.barnert@mytum.de),
Stefan Klink (klink@in.tum.de),
Robert Meyer (robert.m@mytum.de),
and Yankun Ruan (yankun.ruan@mytum.de)

Abstract:
The objective of this report is to present a cost reduction analysis method for an application landscape in a corporate environment. The method is developed based on a case study conducted in a large company's laboratory department. The identification is performed by analyzing the existing application landscape using EAM (Enterprise Architecture Management) Patterns and thereafter outlining respective conclusions on potential cost savings, based on redundant software installations in the laboratory's software topography.

2.1 Motivation

Corporate IT, due to its mostly historical grown complexity and richness, always had been a target for potential cost savings. CFOs and CIOs alike strive for making their application landscape more effective, and at the same time more cost efficient. While this is a laudable goal, there is always the impending danger of missing the big picture, i.e., cutting costs without avoiding the implicated drawbacks [Lu03]. The story of the infamous 'magic orange', mentioning a CFO always urging the CIO to squeeze more money from IT through cost savings (thus creating the allegory), illustrates the commonality of this issue quite well. However, this wide-spread problem mostly stems from the absence of a holistic view on the application landscape and its interdependencies. The goal of this article is to provide a systematic approach for analyzing a given application landscape by means of the EAM (Enterprise Architecture Management) Pattern Catalog and drawing respective conclusions about potential cost savings. The outcome has been developed and discussed in collaboration with the industry partner which is referred to as 'John Q. Public Ltd'. The company kindly provided the team with data about an excerpt of its application landscape, namely the laboratory department, which represented the foundation for the validation of the elaborated method. In the following, *Analysis of an application landscape according to potential cost savings* (see section 2.2) documents the steps, which have to be performed in order to analyze a given application landscape cut-out. Afterwards, an outline on the used information models is drawn in section 2.3 to further clarify the semantics of the compiled visualizations. The recommended entry point for the usage of the EAM Pattern Catalog is a proper distinction of relevant concerns from non-relevant ones (with respect to the stakeholder). As an example specific concerns, which were found applicable in John Q. Public Ltd's context, have been mapped to matching general concerns of the EAM Pattern Catalog in cooperation with the project partners. During these efforts it became clear, that often more than one general concern may fit to a specific concern. Therefore, the team decided to apply a two step process in order to narrow down the set of general concerns: At first, a so called longlist of potential relevant concerns was compiled and presented to the project partners, who pointed out the ones that were most apposite to them. The result was a corresponding shortlist. This shortlist encompassed mainly three concerns taken from the EAM Pattern Catalog and is depicted in table 2.1.

General concern	Short description
C-2	"Where are architectural blueprints or architectural standards used, and are there areas where those standards are breached?"
C-19	"Do the business applications currently used correspond to the architectural blueprints and architectural solutions (architectural standards)? If not, are there documented reasons for this, as e.g., strategic decisions?"
C-44	"How can the operating expenses and maintenance costs be reduced, e.g., by identification of business applications providing the same functionality redundancy)?"

Table 2.1: Shortlist of general concerns based on [Bu08] being used in the further analysis process

2.2 Analysis of application landscapes according to potential cost savings

2.2.1 Context

A historically grown application landscape exists in the company, which was cut down to reasonable measures by the introduction of enterprise-wide technology standards and restructuring initiatives, partly enacted by the management board. Recently, efforts have been made to establish an application repository, holding the relevant data of all applications in the company.

2.2.2 Problem

You already undertook measures for creating a holistic overview on the different application systems. However, you are still facing the deficiency of not being intuitive enough in your expositions and findings for a broad use in management, especially if stakeholders with a business background are concerned. More precisely: There is still some degree of uncertainty on which users use what systems in what processes ('which' and 'what' are two of the key questions EAM has entitled itself to answer). In this particular case you are faced with the problem of applications being redundant and offering the same functionality. Furthermore, your *standardization* endeavors engaged by an enterprise architecture board have been mostly adopted by the company as a whole. It is, however, not always clear where standards are breached and moreover, if those breaches are really necessary or wanted. In addition, your company differentiates two kinds of standards:

- Standards regarding the *platform* underlying an application
- Standards related to the *technology* realizing an application

That is why a solution is needed to systematically detect redundancy and heterogeneity within an application landscape that allows a more distinct location of cost saving potentials within the IT.

2.2.3 Solution

2.2.3.1 Identification of relevant concepts

A variety of information is required in order to analyze an application landscape with regard to redundancy and heterogeneity. Therefore, having an already pre-compiled application repository at hand is highly recommended in the course of the further proceedings. This repository should contain detailed information about the existing applications, e.g.: Is an application standard-conform in terms of the technology by which an application is realized and in terms of the underlying platform. Besides, you should know which application supports which process step. Moreover, you should have knowledge, which department owns what application (department of owners) and what department uses it (department of users). If any of this information is currently not at hand, e.g., through an application repository, additional pieces of information have to be collected, for example by interviewing responsible stakeholders.

Application	Department of owner	Departments of users
IPS	1	D-1, D-2, D-3, D-4, D-5
NONCID	2	D-4
NEL	2	D-4, D-5, D-6
MILS	2	D-5, D-6
DORLS	2	D-5
MISA	2	no information available
VBSD	2	no information available
QAMPS	3	D-1, D-2, D-3, D-4, D-5, D-6
PAP SP	3	D-1, D-3, D-4, D-5
PAP SS	4	D-1, D-2, D-3, D-4, D-5, D-6
KNIZON	5	no information available

Table 2.2: Current list of applications used at John Q. Public Ltd laboratory environment

The data which was gathered by analyzing John Q. Public Ltd application repository is listed in table 2.2. The information in figure 2.1 was collected with the help of an additional presentation given by one of the company's stakeholders, who possessed the necessary information and showed parts of the test processes at John Q. Public Ltd laboratory.

2.2.3.2 Identification and adaptation of visualizations

The methodologies already recommended in the EAM Pattern Catalog serve as a source for V-Patterns that can be reused in an similar context as described above. In order to fit to the given situation, they normally have to be modified according to the respective conditions.

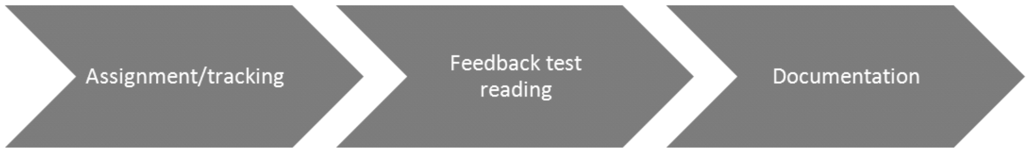

Figure 2.1: Test process steps at John Q. Public Ltd's laboratory (excerpt)

Heterogeneity The cluster map of V-Pattern V-67 is suitable for the analysis of heterogeneity, but has to be slightly adapted to fit to the intended concern. As mentioned before, two types of standards are distinguished: standards regarding the platform and standards regarding the underlying technology.

The use of two different types of standards accordingly leads to two different cluster maps. Both maps are based on pattern V-67 [Bu08] and use the information described in the I-Patterns I-24 and I-67. The pattern I-24 has to be customized by replacing the organizational unit with the department of owners (see section 2.3 for details). The first map considers technology standard conformity exceptions, while the second one deals with platform standard conformity exceptions. In order to obtain a better overview, a third map which merges the two cluster maps is recommended. The resulting *combined map* is shown in figure 2.2, using the provided example data of John Q. Public Ltd. In this visualization each department of owner is represented by a rectangle. Moreover, every application, which is owned by a department, is assigned to the appropriate rectangle. In addition, the standard conformity of an application is marked via color coding: An application, which is marked as standard conform, uses standards concerning the platform and the technology. If the necessary information is missing, the rectangle is simply left blank. The information model, which is used for this visualization, is described more closely in section 2.3.

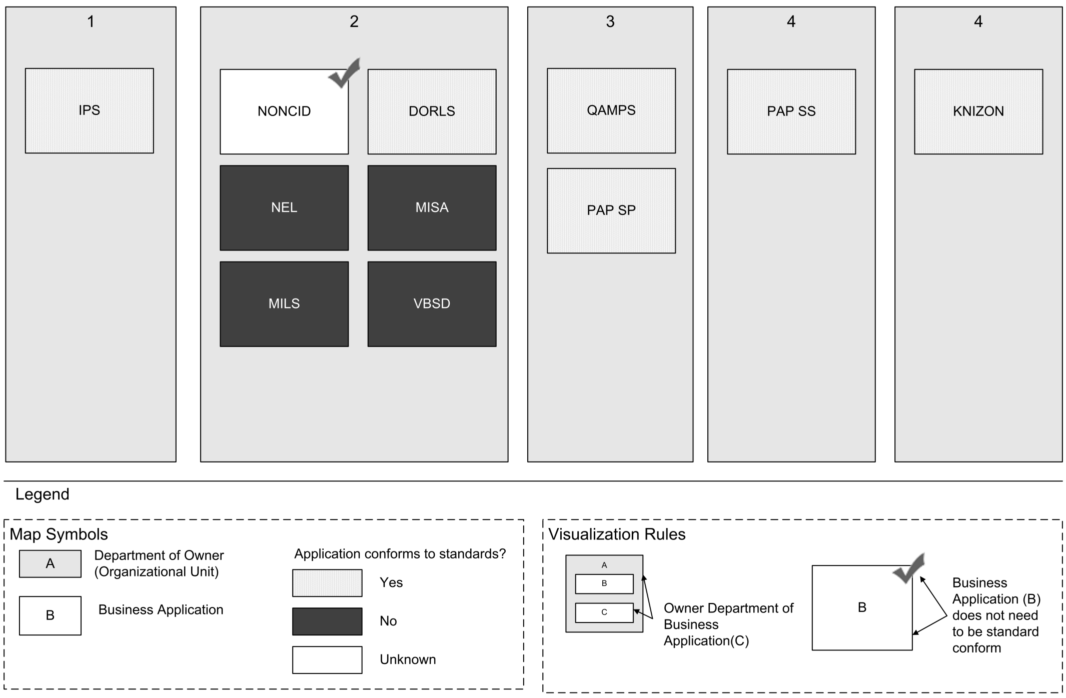

Figure 2.2: V-67 with technology and platform standard conformity exceptions

Redundancy Redundancies within application landscapes can be identified by using the V-Pattern V-29 of the EAM Pattern Catalog [Bu08]. Since this pattern is based on the V-Pattern V-17, it is recommended to model the process support map of this pattern at first. Subsequently the vertical integration of V-29 can be included into this map. The information, which is necessary for the patterns V-17 and V-29, can be inferred from I-Pattern I-30 of the Catalog. Before you can use this I-Pattern, the organizational unit must be replaced with the department of users.

Heterogeneity and redundancy Since the focus of the described methodology lies on both, reducing heterogeneity and redundancy, an extension of V-Pattern V-29 is required. The resulting visualization uses the process support map suggested by pattern V-29 and includes the standard conformity exceptions at the same time. This additional information can be taken from the combined cluster map of V-Pattern V-67, set up in advance. Thus, you receive a process support map, which takes into account both, vertical integration and the standard conformity of the applications landscape. An extended process support map is illustrated in figure 2.3, which shows an excerpt of the John Q. Public Ltd's application landscape. In this visualization the departments of users are depicted on the y-axis and the regarded process steps are illustrated on the x-axis. Moreover, the considered applications are assigned to these two axes. In addition, a color coding shows if an application is standard conform or respectively not conform. If an application is green dyed, it is standard conform. A red labeled application does not conform to standards. A white rectangle labels an application that misses information about standard conformity. A green hook labels an application as not being obligated to conform to a given standard set up by the stakeholders. The information model, which is used for this visualization, is likewise described in section 2.3.

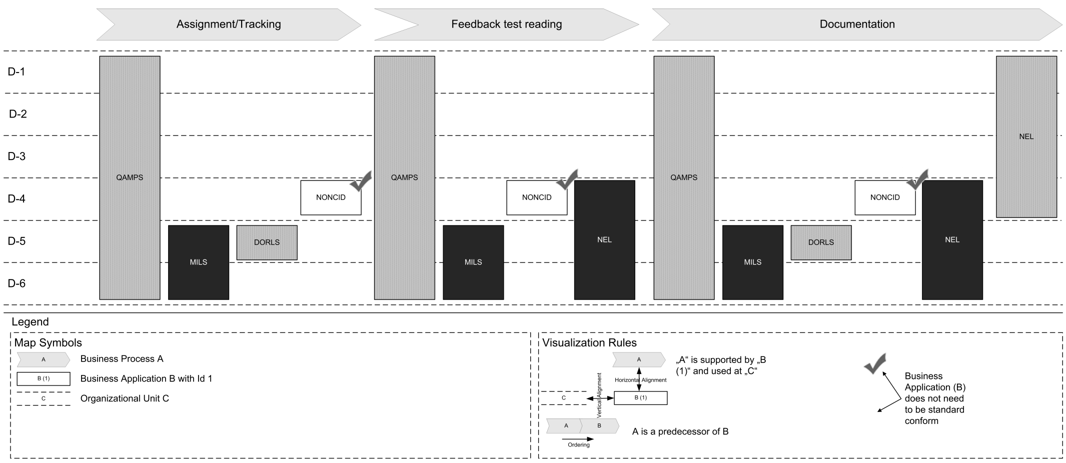

Figure 2.3: Extended V-29 including standard conformity exceptions

2.2.3.3 Identification of potential cost savings

The visualizations, which are recommended in section 2.2.3.2, can be used to find areas of improvement within the application landscape.

Heterogeneity within application landscape One cost saving potential consists of reducing heterogeneity. An application is standard conform, if it uses standards in platform and technology. In order to find those applications, the combined cluster map, which is based on the pattern V-67, can be used (see section 2.2.3.2 and figure 2.2). Besides, the extended process support map (see figure 2.3) proved itself as handy. Both visualizations help to get a general idea about heterogeneity within an application landscape. However, it is not recommended to remove applications only because they are not standard conform. Although an application does not comply, it may still be needed because of a unique functionality. Additionally, it is not guaranteed that every standard conform replacement will reduce costs at all. This is why heterogeneity should conjointly be taken into account with redundancy.

Redundancy of application functionalities The *extended process support map* (see figure 2.3), which combines vertical integration and standard exceptions, can be useful to find applications with equal functionalities. Additional information, which describes the functionalities of an application in detail, is necessary to find those redundancies. An outline of the John Q. Public Ltd's application landscape is shown in figure 2.3. It contains two redundant applications namely QAMPS and MILS. Since QAMPS is used by all departments and MILS only by two departments, MILS might be a candidate for removal.

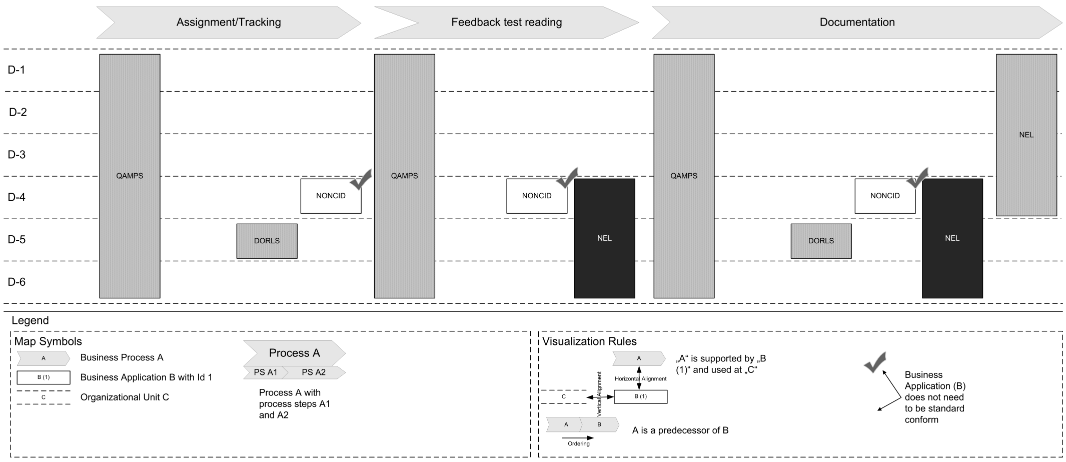

Figure 2.4: Optimized application landscape

Heterogeneity and redundancy So far heterogeneity and redundancy are considered separately. But it is also sensible, to take both simultaneously into account. Applications, which are not standard conform and do not provide a unique functionality, are candidates for removal. Again the *extended process support map*, which is based on the pattern V-29 (see figure 2.3), has proven itself as being very helpful. An example of an optimized application landscape is shown in figure 2.2.3.3. Compared to figure 2.3 the application MILS has been left out, because it is not standard conform and provides the same functionality as QAMPS. All departments except from two use QAMPS to fulfill the same functionality as MILS does. This is why MILS might be phased out. In contrast, NEL cannot be removed: Although IPS and DORLS can be combined to provide the same functionality as NEL, department D-6 does not use any of these two systems. Therefore, NEL is still needed.

2.2.4 Consequences

The identification and removal of non standard conform or redundant applications leads to an optimized application landscape. Quantitative statements about costs savings through the optimization are in most cases very difficult to make if you don't have data in the required quality and granularity. Principally, costs can be reduced in the following areas:

Training costs: If applications are not used any more, there is no need to continue with training users for these applications. This is a potential often not considered.

Operating expense: A lot of complementary assets like workstations for the users or servers are needed in order to operate an application. They become obsolete, if an application is not used any more.

Maintenance costs: During the life span of an application, different maintenance costs occur. These costs result from operating servers or user terminals, for example. They can be saved by reducing heterogeneity and especially by removing redundancy within applications.

2.3 Information models

As mentioned before, the visualizations generated in section 2.2.3.2 represent adaptations of V-Patterns originally found in the EAM Pattern Catalog. During the course of the alignment process, some semantic changes have been applied to the original V-Patterns. They were necessary for the correct interpretation of the artifacts generated throughout the methodology.

The data, which underlies the created visualizations, can be described by means of classes and relationships, which constitute themselves as follows:

The I-Pattern (figure 2.3) concerning the customized pattern V-67 encompasses the artifact `BusinessApplication`, which denotes an actual software installation hosted by a `DepartmentOfOwners`. This depicts a structural entity within an enterprise or company, which has the power over a hosted application.

Furthermore, business applications conform to an `ArchitecturalSolution`, which specifies the concrete underlying technology components from which the application is made off. Architectural solutions, however, do not necessarily meet current business desires or new policies. This is expressed through the entity `DemandedSolution`. An architectural solution conforms to a demanded solution if the attributes `conformsToPlattform` and `conformsToTechnology` both equal true. If this is not the case, there may be reasons for allowing a standard violation (`standardViolationAllowed`), further clarified in `reasonsForAllowingViolations`. To ensure model consistency, all attributes should never be null.

A special `ArchitecturalSolution` is the `NoArchSol`. This class is singleton and means, that the associated `BusinessApplication` does not have a `ArchitecturalSolution`. The stereotype singleton is important, because there should be only one `NoArchSol` instance, to ensure consistency. This way an unknown `ArchitecturalSolution` and a `BusinessApplication` with no specific ArchitecturalSolution can be distinguished.

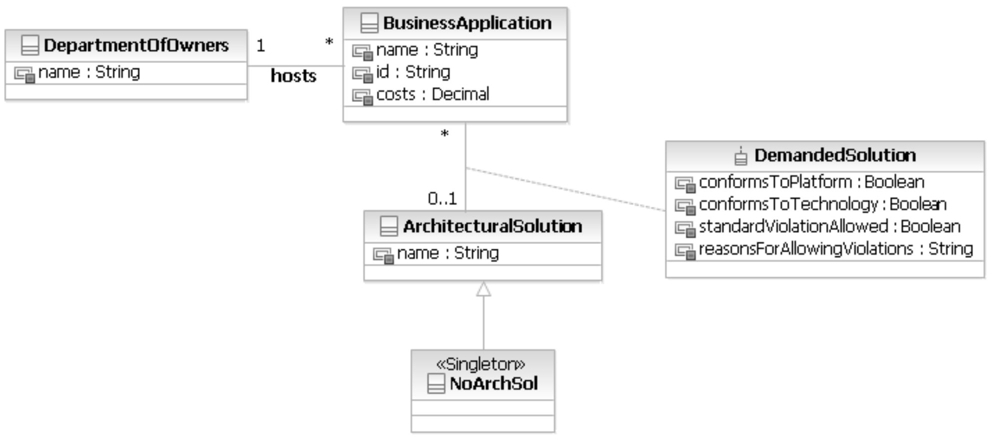

Figure 2.5: I-Pattern for customized V-67

Additionally, V-29 has been semantically modified to fit into the given context. Basically, the merging of the two V-Patterns entails a merging of the two corresponding I-Patterns, underlying the visualizations. The resulting I-Pattern for the customized version of V-29 is shown in figure 2.6. It includes several entities mentioned beforehand (e.g. `BusinessApplication`, `ArchitecturalSolution`, `DemandedSolution`, and `NoArchSol`), as well as some new ones.

A business application supports a `BusinessProcess`, which represents a defined set of steps in a company's workflow. Business processes and their business applications are usually supported at a `DepartmentOfUsers`, which is the counterpart to the afore mentioned department of owners, i.e. the class represents an application's user base.

`SupportRelationship` acts as a supplementary class embodying the `support` of a business process with a specific business application (`supportsWith`) at a certain organizational unit (`supportsAt`) [Bu08].

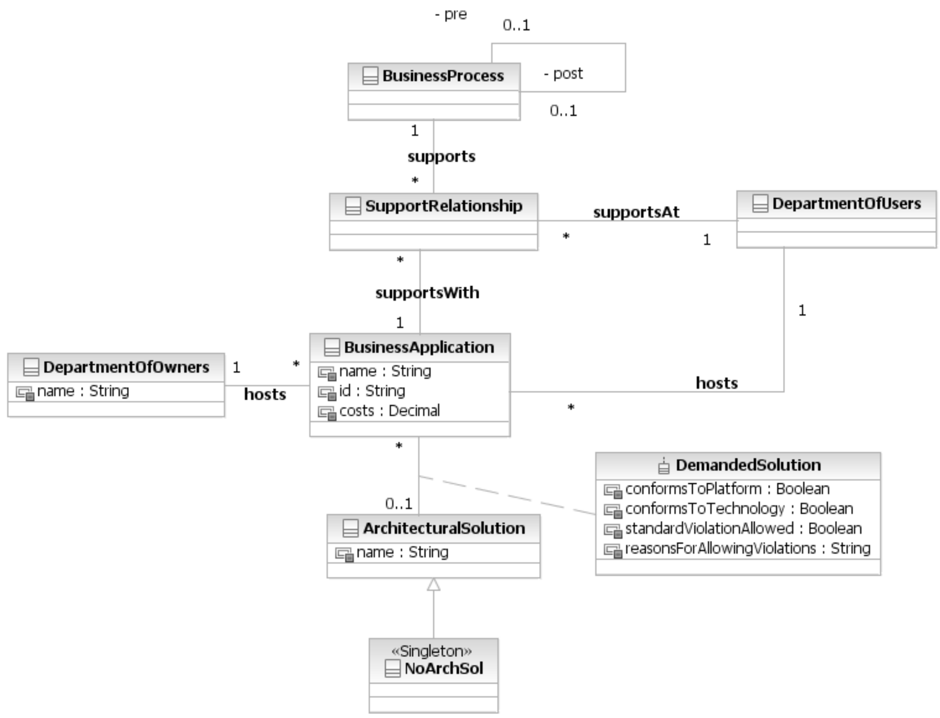

Figure 2.6: I-Pattern for customized V-29 including standard conformity exceptions

2.4 Conclusion

After the expositions about the condition of the laboratory's IT at John Q. Public Ltd, the team would like to share key learnings with the reader gathered during the project's execution. This allows for the incorporation of important experiences made into further projects that will be subject to the companies EAM approach.

At First, it should be mentioned, that EAM, in every respective dimension, proved itself as a *highly political subject* and therefore should be treated with caution. In practice, this means that the reasons for complete project failures very often do not lie within technical issues or the nature of the perceived task itself, but rather in inefficient cooperation of stakeholders and deliberate or undeliberate obstruction of project efforts by stakeholders and decision makers. Those may feel ignored in the sense of e.g. that they are losing their established position of power in an enterprise because the applications they manage are becoming subject to rationalization measures. Therefore, if a report about a very sensitive issue like cost saving or putting down redundant applications is to be compiled, no biased assumptions should be made in the run-up to gathering the necessary intelligence and data.

Furthermore, the team would like to point out, that the whole project proved to be a more *time consuming* effort than previously thought. Since no past field-experiences with the subject were made no one expected, that time would prove itself to be one of the most constraining resources in the project course. With respect to the afore mentioned human factor in EAM, it can be stated here, that one of the most tedious tasks was the visualization of the given data as well as the resulting high need of coordination with the project partners at John Q. Public Ltd. This is an even more remarkable circumstance, taking into account the fact, that the project team had dedicated and competent partners at the company's side, which were highly endeavored in providing the required details needed for modeling the various visualizations. A team without such helping hands could very well face enormous difficulties in generating output in an acceptable quality at the slightest sign of hindrance within the reviewed companies divisions.

At last, with regard to the more academic aspects of this article, we would like to bring up our *experiences with the EAM Pattern Catalog* to give future project teams the chance to develop a broader perspective on the matter. The Pattern Catalog's intent was to ease EAM efforts, by providing general reusable solutions to commonly occurring problems in a given context, i.e., patterns. Practically, this means that for every special company's problem concerning EAM, a general solution exists in form of a pattern. How you apply that pattern, however, is totally up to the reader's needs. It is not necessary to unyieldingly persist on the application of a pattern as it is pointed out in the Pattern Catalog. In this vein, it is a kind of reference model catalog, which delivers very good ideas on how to deal with recurring problems in the context of EAM but should not be taken too strictly in practice. The modified pattern V-29, which is presented in this paper, is owed to this perception: The project partners wanted a visualization, that comprised both, standard conformity exceptions and process support details, but the catalog did not provide the corresponding pattern. Therefore, a new 'hybrid' visualization was created using the given approaches, which satisfied the industry partner's need.

Bibliography

[Bu08] Buckl, S. et al.:. *Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008)*. Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2008.

[Lu03] Lutchen, M.:. *ManagingIT as a business: a survival guide for CEOs*. John Wiley and Sons. 2003.

CHAPTER 3

EAM patterns for the project decision process at a financial service provider

Victoria Bodishevskaya (badzishe@in.tum.de),
Katharina Pflügler (pfluegIk@in.tum.de),
Moissej Sverdlin (sverdlin@in.tum.de),
and Jiayan Xu (xuj@in.tum.de)

Abstract:
Due to the increasing complexity of business transactions and an accelerated rate of change in business models companies are forced to continuously adapt their corporate strategies to maintain their competitive advantages. At the same time, regulatory frameworks as well as the growing dependency on information technology (IT) demand a continuous alignment of business and IT. Therefore, many companies introduce enterprise architecture management (EAM). The EAM Pattern Catalog provides a holistic and generic view on the problem of EAM and is based on best practices of experienced practitioners, which are captured in patterns.

3.1 Introduction

The constantly increasing complexity of business transactions, the accelerated rate of change in business models, and numerous regulatory frameworks as well as the resulting growing dependency on IT contribute to the fact that companies are more and more challenged with technology related risks [FAW07]. For these reasons, enterprises are forced to pay special attention to the alignment of corporate structures (such as organizational structures and processes or corporate information systems and technologies) with strategic goals. Consequently, the acceptance of enterprise architecture management as an approach for managing change and fostering the alignment of business and IT in an organization is rising [FAW07].

Enterprise Architecture Management (EAM) is "a continuous and iterative process controlling and improving the existing and planned information technology (IT) support for an organization. The process not only considers the IT of the enterprise, but also business processes, business goals, strategies, etc. are considered in order to build a holistic and integrated view on the enterprise. The goal is a common vision regarding the status quo of business and IT as well as of opportunities and problems arising from these fields, used as a basis for a continually aligned steering of IT and business" [Er06].

Within the scope of our project, the *EAM Pattern Catalog* [Bu08] is applied to document the project proposal decision process at a functional service provider of an international financial institution.

3.1.1 Project placement in the context of EAM

The EAM mini project is concerned with the two enterprise-level management processes Enterprise Architecture Management and Portfolio Management, which are highlighted in figure 3.1 by red rectangles.

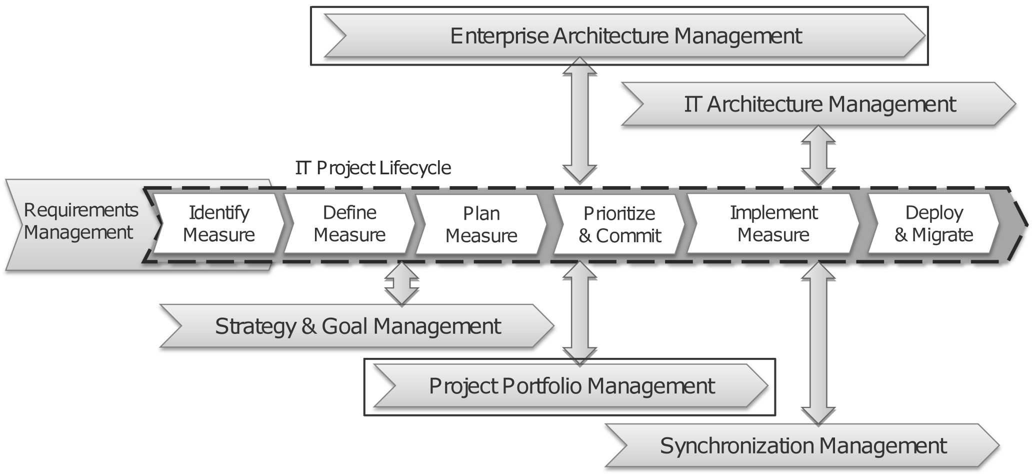

Figure 3.1: Enterprise-level management processes. Source: [Wi07]

The EAM mini project can be placed in the context of EAM activities, which are depicted in [Ch09]. Here the EAM mini project pertains to the activity "Plan EA" shown in figure 3.2.

3.1.2 Project context

The project was carried out in cooperation with a functional service provider of an international financial institution. This company claims to be the largest internal service provider, supplying its services to subsidiaries of the parent company in more than 20 countries. The scope of company's services includes ensuring quality of service levels as well as their further development, securing integrity of data and business processes, and business process monitoring.

EAM is shared between two architecture subdivisions. On the one hand the infrastructure and customer services department, and on the other hand a unit managing the application landscape, business relationship management, and single domains. The coordination between these two divisions is carried out via a cross-functional committee.

The EAM mini project is concerned with the project proposal decision process already established at the organization. Obviously, the company cannot approve all of the submitted proposals because of the limitations in budget and other resources. To enable a reasonable evaluation and selection of the proposals, a project proposal documentation process takes place.

Within this project proposal decision process a subprocess is concerned with the project proposal documentation. This subprocess in turn is composed of further subprocesses: identify and visualize

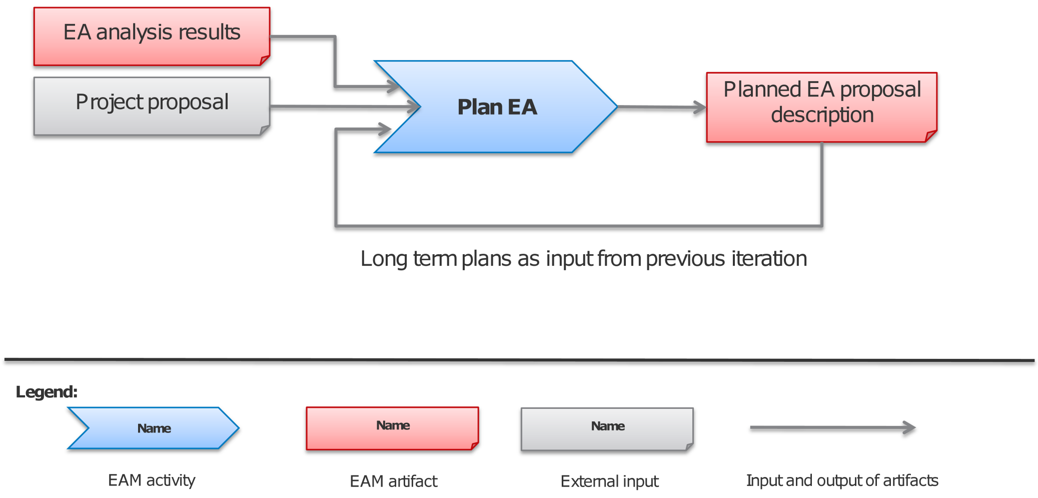

Figure 3.2: EAM activity - plan EA. Source: [Ch09]

affected architecture elements, define and estimate indicators, estimate costs, create solution concept. The scope of the EAM mini project is limited to the subprocess of identifying and visualizing affected architecture elements, which is an essential aspect, since the company's application landscape contains a large amount of business applications with heterogeneous technologies. Responsibilities in this subprocess are divided among the roles of solution designer, project manager, architecture team and team responsible for the business application to be developed. There are two types of artifacts resulting from the subprocess. Firstly, a visualization containing the business applications affected by a project proposal together with their domains. The second artifact is basically identical except for additional technical information, which is added to the latter.

3.1.3 Course of action

In the course of this project a subprocess of the project proposal decision process was described and thus documented with the aid of EAM patterns. This subprocess is concerned with the identification and visualization of architecture elements affected by a certain project proposal.

At the beginning all M-Patterns, V-Patterns and I-Patterns, which can be applied to support a project proposal decision process were identified. This set of EAM Patterns, in this case called long list (see 3.3), constitutes the starting point for a meeting with the industry partner, in which a short list was compiled. Criteria for sorting out a certain EAM pattern were, e.g., EAM pattern is applied at the organization, but beyond the scope of this project or EAM Pattern is not relevant. In order to familiarize with the topic of project proposal decision processes scientific literature was consulted as well as documentation of the organization-specific project proposal decision process supplied by the company. Based on this information, new EAM patterns, described in Section 3.2, were created to supplement the existing EAM pattern base in the field of identifying architecture elements affected by project proposals. Finally, the set of relevant I-Patterns and the new I-Patterns was integrated to an organization-specific information model.

3.2 Creation of new EAM patterns for the project proposal decision process

3.2.1 Identifying architecture elements affected by project proposals (M-a)

This M-Pattern supports the process of documenting a project proposal by providing guidance for identifying architecture elements affected by a certain project proposal.

Example

The SoCaStore has to evaluate a number of project proposals in order to determine projects to be carried out in the next period. Given limited resources, the SoCaStore tries to identify those project proposals, which could provide the greatest possible advantage and minimize potential risks. Thereby, an important factor to be taken into account is the number of architecture elements in the SoCaStore's application landscape affected by a certain project proposal.

Context

An enterprise with a large number of project proposals and an established project proposal decision process, in which project proposals have to be documented.

Problem

You have a large number of project proposals, which have to be documented and evaluated in the course of a project proposal decision process. You do not know yet which elements of the application landscape will be affected by a certain project proposal. You have to estimate the risk of a project proposal and therefore have to assess the number of affected architecture elements.

This methodology describes the basic steps for creating an overview on the impacts of a project proposal approval and thus assists the project proposal decision process by contributing to the documentation of project proposals.

Modeling business applications and domains with interfaces: To start with, information about business applications and interfaces is collected and clustered according to their respective domains (see Business Application and Domain Cluster Map with Interfaces (V-a) (section 3.2.2.1).

Modeling business applications and domains with technical details: In some cases more detailed information, for instance transfer protocols, integration technology and architectural solutions, may be required to support the decision making. This additional data can be visualized in Business Application and Domain Cluster Map with Technical Details (V-b) (section 3.2.2.2.

Modeling domain states: Further, the visualizations created can be extended to display certain characteristics of the affected architecture elements. In this step, the impacts of a project proposal on the existing domains can be revealed using the Domain Status (V-c) (section 3.2.2.3 layer.

Modeling life cycle phases of business applications: A further characteristic, which has to be taken into consideration, is information about the life cycle phases of business applications affected by a project proposal (see Business Application life cycle (V-d) (section 3.2.2.4).

Modeling project proposal effects: In this step the effect type of a certain project proposal on business applications is determined and visualized according to the V-Patterns Effects of a Project Proposal on the Application Landscape (V-37) and Business Application Planning (V-39) in [Bu08].

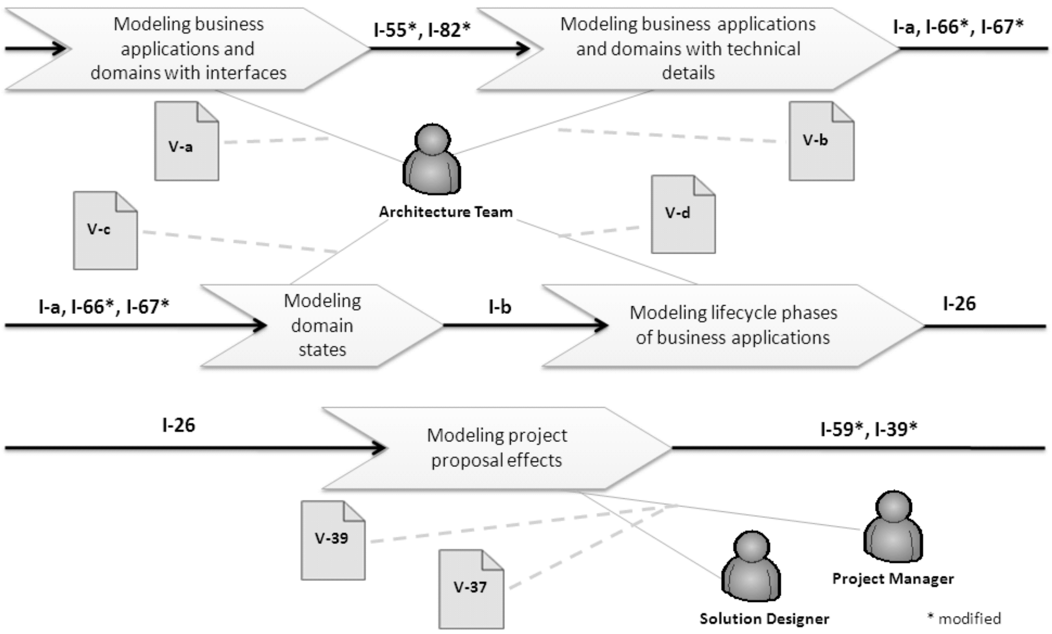

Figure 3.3: Methodology Process

Implementation

In order to implement this M-Pattern within an organization it is first of all necessary to identify the stakeholders and their accountabilities. There are many roles involved in the project proposal decision process. Thus, responsibilities can be divided between the project manager, solution designer, and an architecture team. Each of the identified stakeholders is then responsible for delivering corresponding artifacts.

Consequences

A benefit of this M-Pattern is that it provides a concise overview of the impacts on the application landscape caused by a project proposal. Therefore, one of the main advantages is transparency about the scope of effects triggered by the project proposal. As a result, risks may be estimated more accurately.

Since a lot of data concerning affected architecture elements needs to be collected, the implementation of this M-Pattern has to be thoroughly planned.

See Also

In order to support the implementation of M-Pattern identifying affected architecture elements of project proposals, the following V-Patterns should be considered:

- Business Application and Domain Cluster Map with Interfaces (V-a) (see section 3.2.2.1)
- Business Application and Domain Cluster Map with Technical Details (V-b) (see section 3.2.2.2)
- Domain Status (V-c) (see section 3.2.2.3)

- Business Application life cycle Layer (V-d) (see section 3.2.2.4)
- Effects of a Project Proposal on the Application Landscape(V-37) in [Bu08]
- Business Application Planning (V-39) in [Bu08]

3.2.2 V-Patterns

Within the scope of the project, four new V-Patterns were created. These are base maps (*Business Application and Domain Cluster Map with Interfaces, Business Application and Domain Cluster Map with Technical Details*), as well as layers (*Business Application life cycle and Domain Status*).

3.2.2.1 Business Application and Domain Cluster Map with Interfaces (V-a)

This V-Pattern visualizes relationships between business applications and domains by using the concept of clustering and by specifying the types of interfaces being used.

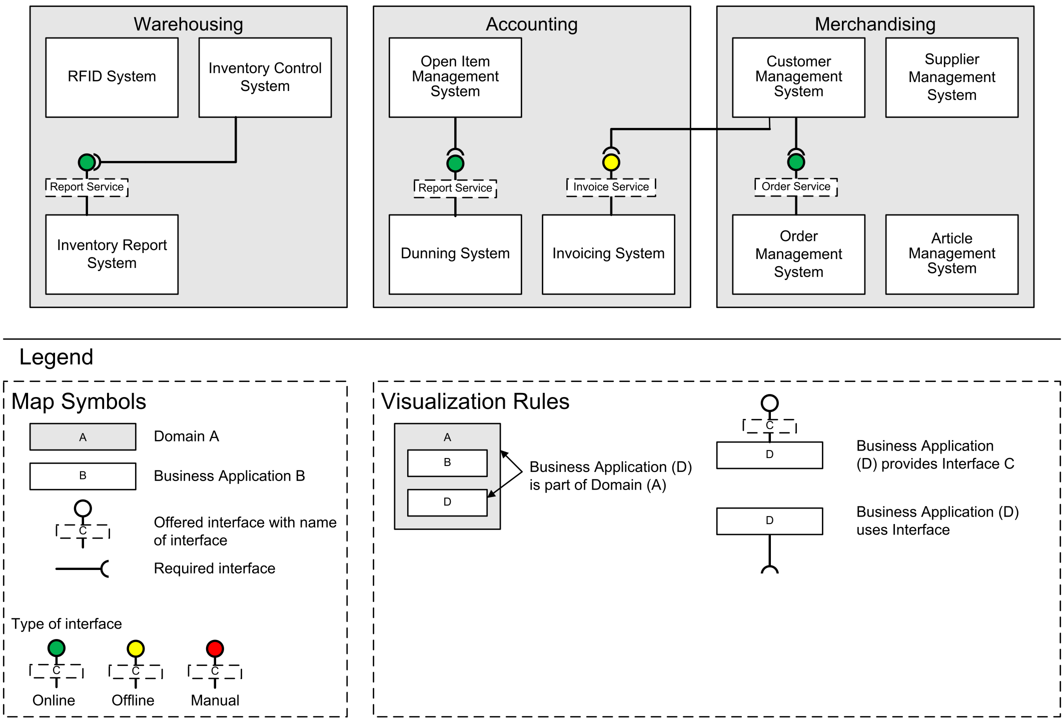

Figure 3.4: Exemplary view for Business Application and Domain Cluster Map with Interfaces (V-a)

This V-Pattern belongs to the software map type Cluster Map, which uses the concept of grouping (clustering) of elements in a visualization to express a relationship between them. The positioning of the different clusters is of minor importance as it does not convey any semantic information, but may be used to improve recognition. In this V-Pattern a cluster map like viewpoint is used to group business applications in domains. A business application may appear only once within a view corresponding to this V-Pattern, as domains are disjoint.

In a variant of this V-Pattern interfaces might be visualized using a different notation like, e.g., arrows.

This base map can also be utilized in combination with:

- Application Lifeycle (V-d) (see section 3.2.2.4)
- Domain Status (V-c) (see section 3.2.2.3)
- Effects of a Project Proposal on the Application Landscape (V-37) in [Bu08]
- Effects of a Project Proposal on the Application Landscape (detail) (V-39) in [Bu08]

A more detailed viewpoint is provided by the V-Pattern *Business Application and Domain Cluster Map with Technical Details* (V-b) (see section 3.2.2.2).

This V-Pattern is based on information according to I-Patterns:

- Domains and Components (I-55) in [Bu08]
- Interfaces and Business Objects (I-82) in [Bu08]

3.2.2.2 Business Application and Domain Cluster Map with Technical Details (V-b)

This V-Pattern visualizes the relationships between business applications and domains by using the concept of clustering. Additionally, it specifies the architectural solutions of business applications, types of interfaces and transfer protocols used, as well as the integration technology applied in domain connections.

This V-Pattern belongs to the software map type Cluster Map, which uses the concept of grouping (clustering) of elements in a visualization to express a relationship between them. The positioning of the different clusters is of minor importance as it does not convey any semantic information, but may be used to improve recognition. In this V-Pattern a cluster map like viewpoint is used to group business applications in domains.

In addition to the data contained in the Business Application and Domain Cluster Map with Interfaces, this V-Pattern specifies architectural solutions of business applications, integration technologies connecting domains, as well as transfer protocols.

Business Application and Domain Cluster Map with Interfaces is used by the M-Pattern *Identifying architecture elements affected by project proposals* (M-a) (section 3.2.1).

This base map can also be utilized in combination with:

- Application life cycle (V-d) (section 3.2.2.4)
- Domain Status (V-c) (section 3.2.2.3)

A less detailed viewpoint is provided by the V-Pattern *Business Application and Domain Cluster Map with Interfaces* (V-a) (section 3.2.2.1).

This V-Pattern is based on information according to subsequent I-Patterns:

- Domains and Components (I-55) in [Bu08]
- Interfaces and Business Objects (I-82) in [Bu08]
- Technology and Connector Usage (I-66) in [Bu08]
- Demanded Architectural Solutions (I-67) in [Bu08]
- Domain Integration (I-b) (section 3.2.3.2)

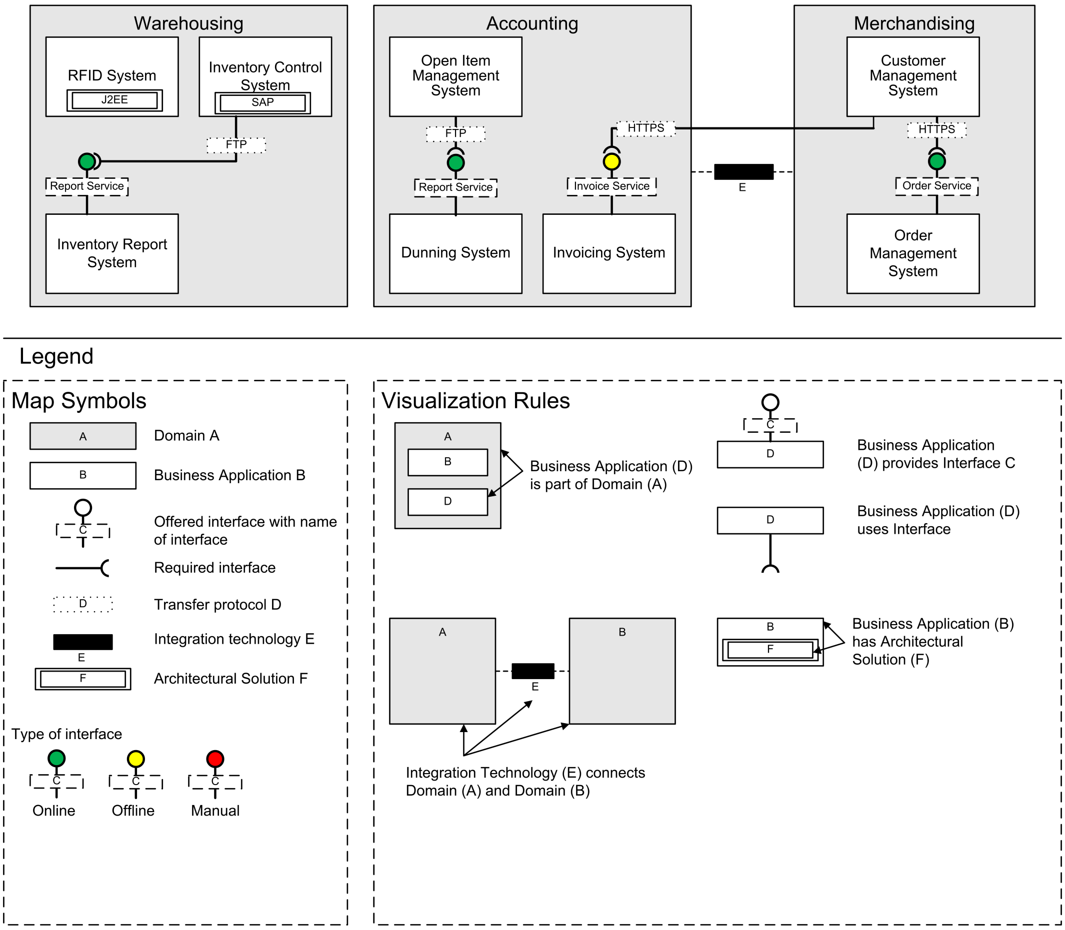

Figure 3.5: Exemplary view for Business Application and Domain Cluster Map with Technical Details (V-b)

3.2.2.3 Domain Status (V-c)

This V-Pattern visualizes current statuses of domains.

This V-Pattern describes a layer, showing the statuses of *Domains*, which contain business applications affected by a specific *project proposal*. The current status of each domain is indicated by a particular color, as described in the legend. Figure 3.6 above demonstrates how the *Domain Status Layer* can be utilized on an exemplary cluster map. Thereby, the diagram complements a textual project proposal description.

Further colors can be used to depict additional domain statuses.

This Domain Status layer is used by the M-Pattern Identifying architecture elements affected by project proposals (M-a) (section 3.2.1) and is based on the I-Pattern Domain Status (I-a) (section 3.2.3.1).

Since Domain Status constitutes a layer used in combination with a base map, it cannot be used solely. Following V-Patterns can serve as a base map for this layer:

- Business Application and Domain Cluster Map with Interfaces (V-a) (section 3.2.2.1)
- Business Application and Domain Cluster Map with Technical Details (V-b) (section 3.2.2.2)

This V-Pattern is based on information according to I-Pattern Domain Status (I-a) (section 3.2.3.1).

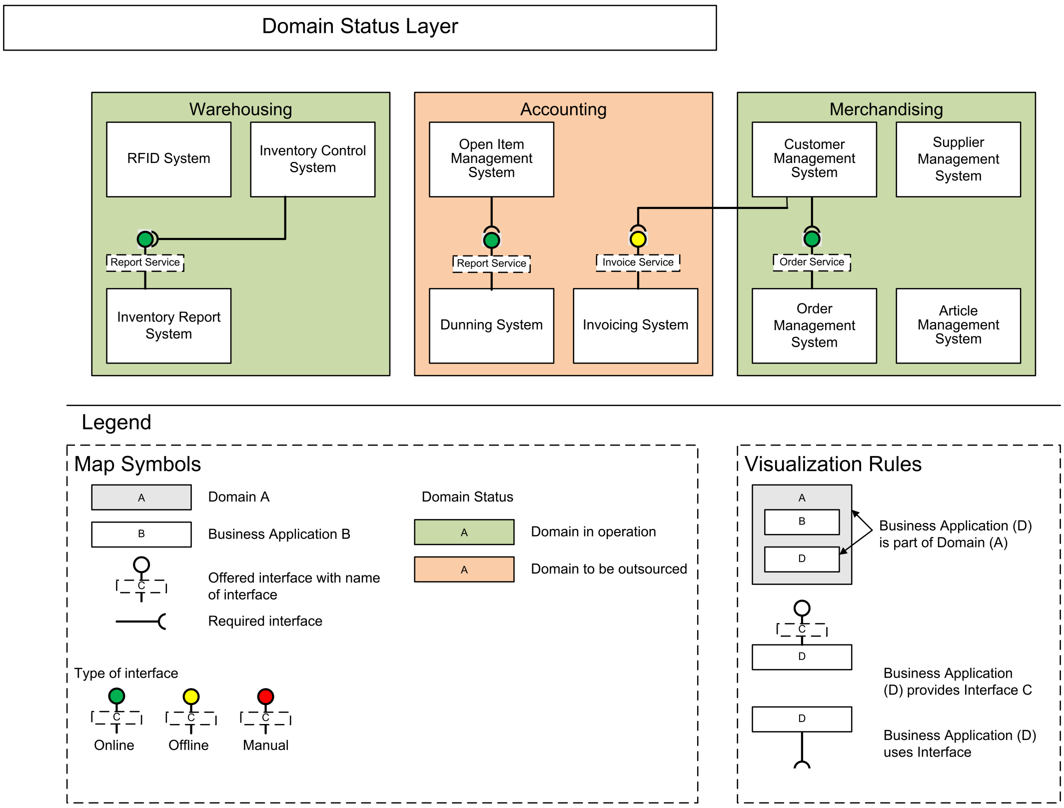

Figure 3.6: Exemplary view for Domain Status (V-c)

3.2.2.4 Business Application Life Cycle (V-d)

This V-Pattern visualizes current life cycle phases of business applications.

life cycle phases of business applications can be visualized on a layer. The current life cycle phase is depicted by overlaying each symbol representing a business application with a certain color. In the context of this V-Pattern red color shows that a business application is about to be replaced, while business applications in operation are respectively highlighted by yellow for local and blue for global ones. Further colors can be used to depict additional life cycle phases of business applications.

The V-Pattern *Domain Status* layer is used by the M-Pattern *Identifying architecture elements affected by project proposals* (M-a) (section 3.2.1).

This pattern constitutes a layer for a software map and therefore cannot be used solely. Possible V-Patterns, which can be used as a base map for this layer are the following:

- Business Application and Domain Cluster Map with Technical Details (V-b) (section 3.2.2.2)
- Business Application and Domain Cluster Map with Interfaces (V-a) (section 3.2.2.1)
- Process Support Map (V-17)
- Business Application and Organizational Unit Cluster Map (V-24) in [Bu08]
- Cluster Map for using Relationship (V-25) in [Bu08]
- Process Support Map visualizing horizontal Integration (V-28) in [Bu08]

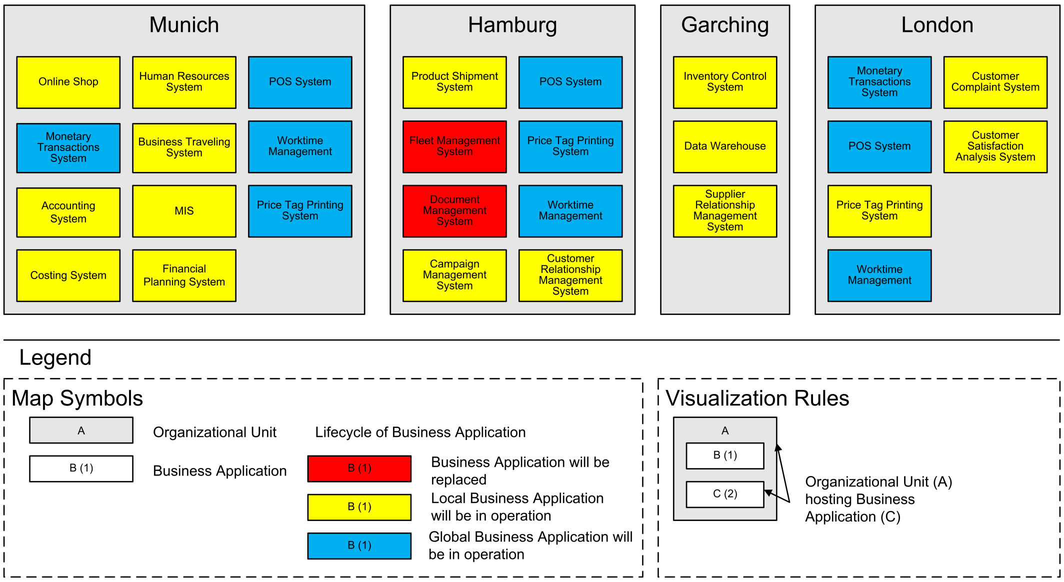

Figure 3.7: Exemplary view for Business Application life cycle (V-d)

- Process Support Map visualizing vertical Integration (V-29) in [Bu08]
- Process Support Map visualizing vertical and horizontal Integration (V-30) in [Bu08]
- Cluster Map visualizing Business Object Flows between Business Applications (V-48) in [Bu08]

This V-Pattern is based on information according to I-Pattern Business Application Life Cycles (I-26).

3.2.3 I-Patterns

The subsequent sections primarily depict the newly created I-Patterns and further provide a detailed description of integrating I-Patterns to the organization-specific information model.

3.2.3.1 Domain status (I-a)

This I-Pattern shows how information about the status of a domain can be stored.

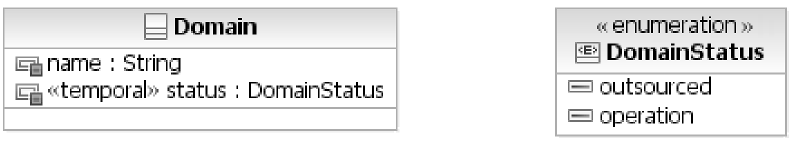

Figure 3.8: Information model fragment for Domain Status (I-a)

The solution for the problem described above is based on two entities and one relationship, which are defined as follows:

Domain: Describes a logical grouping into areas relevant to business, e.g., customer, products, etc.. The status attribute is temporal, because only one status is allowed at the same time, but history should be stored.

DomainStatus: Enumeration that describes the different statuses a domain can be in, e.g., outsourced, operation, etc..

3.2.3.2 Domain integration (I-b)

This I-Pattern shows how information about connection between different domains can be stored.

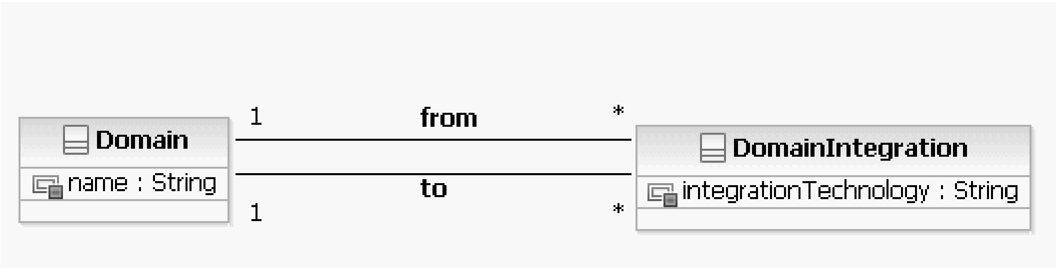

Figure 3.9: Information model fragment for Domain Integration (I-b)

The solution for the problem described above is based on two entities and two relationships, which are defined as follows:

Domain: Describes a logical grouping into areas relevant to business, e.g., customer, products, etc..

DomainIntegration: Describes the kind of integration (including technologies) which is used between domains.

DomainFromDomainIntegration: The association from indicates which DomainIntegration the first domain uses to get a connection to the second domain.

DomainIntegrationToDomain: The association to indicates which DomainIntegration is used by the first domain to get a connection the the second domain.

3.2.4 Integrating I-Patterns

I-Patterns relevant to the EAM mini project are depicted in the short list in 3.3. They were integrated to the organization-specific information model shown in figure 3.10 by proceeding as in [Pf08]. I-35, I-39, and I-57 are very akin and that is why one of them had to be chosen. The decision was made towards I-39, as it was perceived to suit at the best to the information requirements. As next, I-59 was integrated in order to enrich the class `ProjectProposal` with several attributes. The remaining I-Patterns shown in figure 3.7 were selected while creating the new V-Patterns. I-26 was integrated as information since this I-Pattern is required to visualize the `Lifecyclephases` of `BusinessApplications` in V-Pattern *Business Application Life Cycles*. Information according to I-55 is needed for the two newly created base maps *Business Application and Domain Cluster Map with Interfaces* and *Business Application and Domain Cluster Map with Technical Details*. I-55 was adjusted by replacing the class Component with `BusinessApplication`. I-Patterns I-66 and I-67 were added due to V-Pattern *Business Application and Domain Cluster Map with Technical Details*. I-Pattern I-66 is added in a reduced form. `Connector` and `ConnectorRole` were removed and the association classes `Usage` and `AbstractUsage` were turned into normal uses associations. I-82 was integrated as information since this I-Pattern is needed for visualizing `Interfaces` in the two base maps *Business Application and*

Domain Cluster Map with Interfaces and *Business Application and Domain Cluster Map with Technical Details*. I-82 was modified to the extent that the class `BusinessObject` and its relationship to `Interface` were omitted and furthermore the attribute `usedTransferProtocol` was added. Besides the I-Patterns listed in the EAM Pattern Catalog, two newly created I-Patterns were integrated into the information model. **I-a** (section 3.2.3.1) and **I-b** (section 3.2.3.2) extend the concept of domains and are required for creating visualizations according to V-Pattern *Business Application and Domain Cluster Map with Technical Details*.

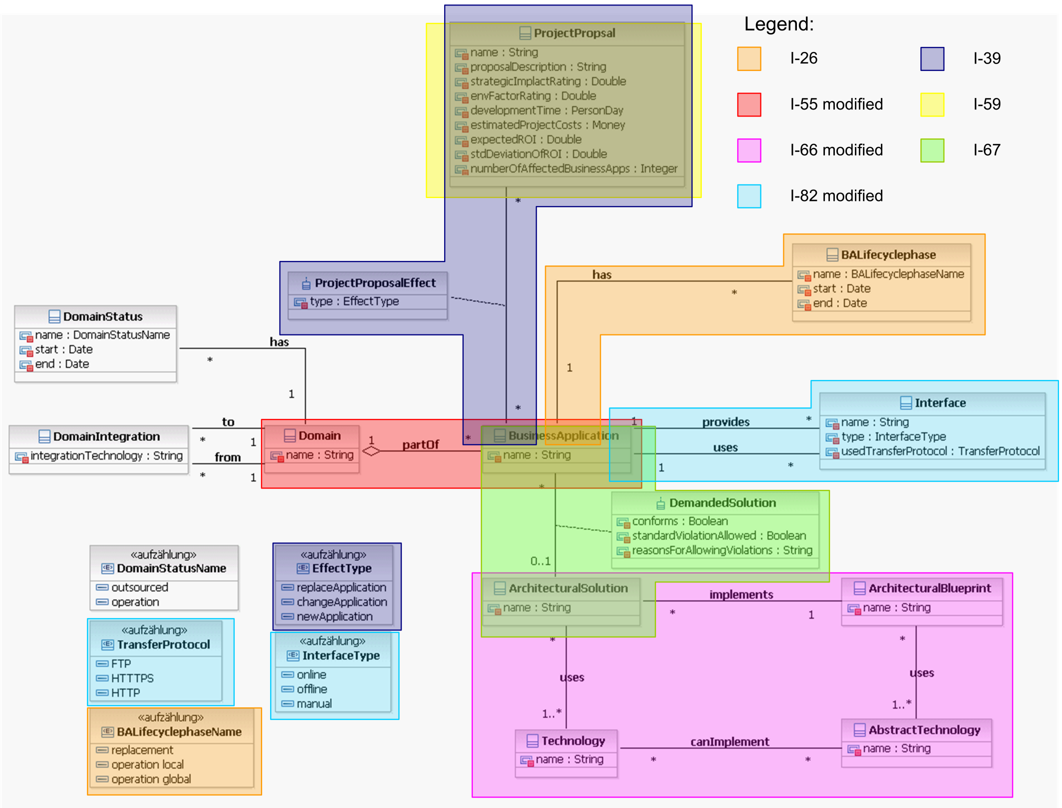

Figure 3.10: Organization-specific information model

3.3 Resume and outlook

The objective of this report was to document a part of the project proposal decision process at a functional service provider with the aid of EAM patterns. The scope of the project was limited to the subprocess of identifying and visualizing architecture elements affected by project proposals. Therefore, in the course of the project the theoretical fundamentals as well as the organization-specific project proposal decision process were investigated at first. As next, M-Patterns, V-Patterns and I-Patterns applicable to the process in general were identified by using the EAM Pattern Catalog (see long list in 3.3). Subsequently, the chosen EAM patterns were sorted out in accordance with the relevance to the project proposal decision process established at the functional service provider (see short list in 3.3). Furthermore, new EAM Patterns were designed. This includes the M-Pattern *Identifying architecture elements affected by project proposals* (M-a) (see Section 3.2.1) as well as four V-Patterns, which are described in Section 3.2.2. The creation of new I-Patterns also became necessary (see Section 3.2.3). Finally, the set of relevant I-Patterns from the EAM Pattern Catalog and the newly developed I-Patterns were integrated to an organization-specific information model in Section 3.2.4.

The additionally developed EAM patterns were so far only assessed by the advisor of this report. For reasons of quality assurance, it might be reasonable to present the newly developed EAM patterns to the EAM pattern community and have those patterns revised by several experts and practitioners. It might also be interesting to investigate further subprocesses of the project proposal decision process and develop EAM patterns.

Appendix

Short and long list of selected EAM patterns

Id	Name	Long list	Short list
M-24	Project Portfolio Management	X	
M-25	Monitoring of the Project Portfolio	X	
M-26	Decision for Project Approval	X	
M-3	Management of Homogeneity	X	X
M-4	Management of Blueprint Conformity of the Application Landscape	X	X
V-35	Proposal Impact Table	X	
V-37	Effects of a Project Proposal on the Application Landscape	X	X
V-38	Effects of a Project Proposal on Technologies	X	
V-39	Effects of a Project Proposal on the Application Landscape (detail)	X	X
V-57	Expected Proposal Effects	X	
V-59	Financial Project Portfolio Overview	X	
V-60	Strategic Project Portfolio Overview	X	
V-61	Technical Project Portfolio Overview	X	
I-35	Project Proposal Targets	X	X
I-38	Changing Basic Technologies	X	
I-39	Planned Proposal Effects	X	X
I-57	Proposed Changes	X	X
I-59	Project Proposal	X	X

Table 3.1: Long and short list of EAM patterns supporting the project proposal decision process

Bibliography

[Bu08] Buckl, S. et al.:. *Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008).* Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2008.

[Ch09] Chair of Informatics 19 Technische Universität München:. *Lecture: Software Engineering for Business Applications - Master Course.* online. 2009.

[Er06] Ernst, A. M. et al.:. *Using Model Transformation for Generating Visualizations from Repository Contents – An Application to Software Cartography.* Technical report. Technische Universität München, Chair for Informatics 19 (sebis). Munich, Germany. 2006.

[FAW07] Fischer, R.; Aier, S.; Winter, R.:. *A Federated Approach to Enterprise Architecture Model Maintenance.* In *Enterprise Modelling and Information Systems Architectures - Concepts and Applications, Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA 2007), St. Goar, Germany, October 8-9, 2007.* pages 9–22. St. Goar, Germany. 2007.

[Pf08] Pflügler, K.:. *Evaluation and Extension of the EAM Pattern Catalog in a German Insurance Company.* Bachelor's thesis. Fakultät für Informatik, Technische Universität München. 2008.

[Wi07] Wittenburg, A.:. *Softwarekartographie: Modelle und Methoden zur systematischen Visualisierung von Anwendungslandschaften.* PhD thesis. Fakultät für Informatik, Technische Universität München, Germany. 2007.

CHAPTER 4

eLearning in medicine

Alexander Kaletsch (kaletsch@in.tum.de),
Dorian Proksch (dorian.poksch@mytum.de),
Tursinie Teuntchou (tursiniet@yahoo.fr),
and Christoph Theel (theel@in.tum.de)

Abstract:
The integration of information systems is a difficult and error-prone process because many aspects have to be considered. In order to deal with th aforementioned complexity, a consistent documentation is important. This article describes the integration of the three systems *eLearning*, *MediTUM*, and *Casus* at the Technische Universität München. To get a user's perspective on the integration of the three systems a survey about the usage behavior was conducted and evaluated. The result was that some improvements regarding the usability are desired. Also, an enterprise architecture management (EAM) pattern was identified and adapted to the integration process to show a best practice approach of documenting the integration of systems. In addition, the conducted survey was generalized to create the new EAM pattern "Survey based analysis of system integration". This pattern helps to successfully evaluate the integration process from a user perspective.

4.1 Motivation

4.1.1 Situation

The Technische Universität München (TUM) uses three information systems to manage their teaching capability, in 2004 the central platform *eLearning* was introduced at TUM. More than 19.000 users, students and lecturers use *eLearning* in their daily work. More than 600 lectures and other events are administrated through the platform. In order to satisfy the specific needs of medicine students and their lecturers the *eLearning* system was technical integrated with two other systems: the *MediTUM* system, a campus management system located at the medicine department, and Casus, a platform which enables case-based learning for students.

The interaction between the three systems is quite complex because different interface types are used. Furthermore, there are some organizational deficiencies like the non-existence of students and lecturers accounts in TUM's information systems. Medicine students, which are in their first four semesters, haven't got access to TUM's information systems but need to use *eLearning* for managing their lectures. Also physicians, who are not employed at TUM, have no access to TUM's systems, although they are supposed to hold and administer lectures via *eLearning*.

The *IntegraTUM*[1] team led the integration project of the three systems and is still continuing to improve provided the services through a better integration. *IntegraTUM's* goal is the creation of a user-friendly infrastructure for information and communication at TUM by reducing costs at the same time. Therefore, redundant systems should be eliminated through the integration process.

4.1.2 Project goal

At the present time usage rates of medicine students for the *eLearning* system are very low, although the systems *MediTUM* and *Casus* are already integrated. Therefore, our project team was asked to identify the causes for this problem. We were asked to examine the integrated systems from a technical point of view. This would enable *IntegraTUM* to gain a precise overview of the interactions between the respective systems and their customers. We also wanted to find out, if there are any doubts or objections against the systems or their integration, which prevents the productive usage of the systems and their interplay.

4.1.3 Course of the project

We started the project with an analysis phase to gather requirements and to get familiar with the systems. It became clear that our customer wanted visualizations modeling the integrated systems and their customers' usage. Within the main project phase after the external kickoff-presentation, we had weekly meetings with our project partner in order to get feedback for every model we designed and the next steps we were going to take. This customer-focused approach enabled us to accurately respond to our customer's needs and demands. We decided to split our modeling phase into four stages: At first, we modeled the systems' integration on a very abstract. Secondly, we showed the user's interaction with the systems. At third, we had a detailed look of the authentication procedures at the interacting systems and the used protocols. Finally, we built and conducted an online survey where we asked students and lecturers for their experience with the three systems.

4.1.4 The EAM perspective on the project

One of the central tasks of the project was to document the integration of the three systems *MediTUM*, *eLearning*, and *Casus* because no complete documentation exists. We used the EAM Pattern Catalog (EAMPC) provided by the Chair for Software Engineering for Business Information Systems of the TUM as a reference work. This catalog contains best-practice solutions for EAM problems and provides guidelines for a consistent documentation of enterprises' architectures. Therefore, our documentation is adopted to these guidelines to avoid ambiguities.

Another important task was to create a survey to ask users about their perception of the integration of the applications in their every day work. This task is connected to the *demand management* process considering the model of processes for managing the application landscape of [Wi07] (see Figure 4.1). It is a process, which captures new demands and forms an entry point for new initiatives [Ch09]. These demands are first undocumented and unstructured. That is why demand description documents are

[1] See http://portal.mytum.de/iuk/integratum/index_html for further information about the IntegraTUM project.

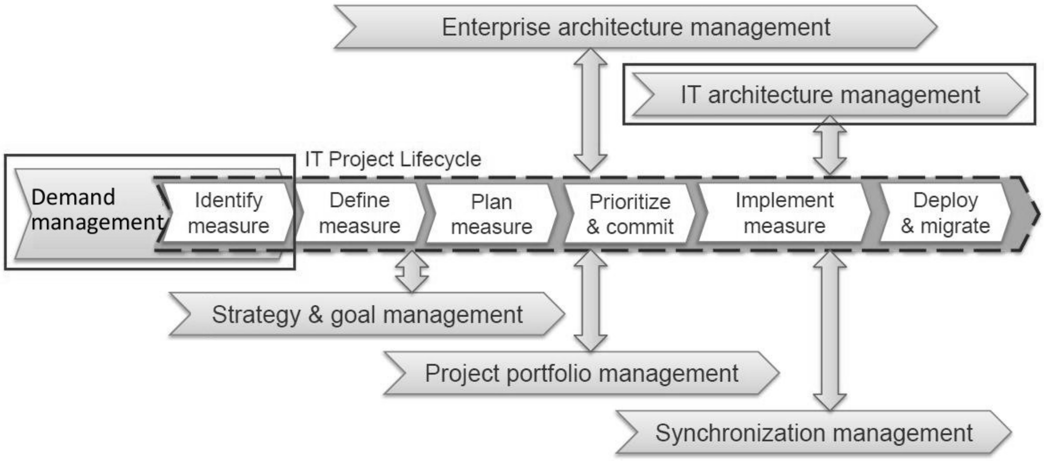

Figure 4.1: Processes for the management of application landscapes according to [Wi07]

created in this process, which consolidates these demands. Based on these documents, proposals are built, which are the final output of this process. The user survey in our project result in new demands as derived from the user satisfaction with the current system integration. This gives users the opportunity to propose new demands or currently unfulfilled ones. These demands could then be used to create new project proposals. The *demand management* is initiated by IT in our case. However, realized demands should result in optimized business processes and improved usability for the user.

4.2 Project context

The system landscape at TUM has changed a lot in the last years. Objectives of these changes was inter alia the goal to achieve a centralized management system. So far the different systems have been connected to exchange the necessary data in order to enable the students to access their data on the respective systems. Until 2008, the *UNIVIS* system and the *myTUM-platform* have been the two major systems next to *CLIX platform*, known as *eLearning system*. In 2009, *UNIVIS* and *myTUM* were replaced by *TUMOnline*, a centralized student organization system.

Within *TUMOnline* not only students are administrated. The system contains all data necessary for the whole university staff and external staff. Regarding the medicine faculty, these are all employees of TUM, students, doctors of the clinical center, external doctors, affiliates of the Ludwig-Maximilians-Universität (LMU), and guests. As all these users do have different rights in the different teaching systems at the TUM. *TUMOnline* uses the *TUM Identity Management* (TUM-IM) to verify the different users in these systems and control their access to the needed data. Figure 4.2 shows the connection of *TUMOnline* and the *TUM IM* with the three considered systems *MediTUM*, *eLearning*, and *Casus*. During a new registration at *TUMOnline* a new account is created automatically within *eLearning*, including the setting of the correct user rights. Within *eLearning* the user has the necessary rights to administrate courses. Tutors can enter new courses and the needed material, set appointments, provide information, and more. Students can apply for the different courses and access data provided by the tutors.

Figure 4.2 schematically shows the interconnection of the relevant systems via data exchange methods. Further, the function executed by the teaching management systems are denoted. In the following an overview over the three systems is given:

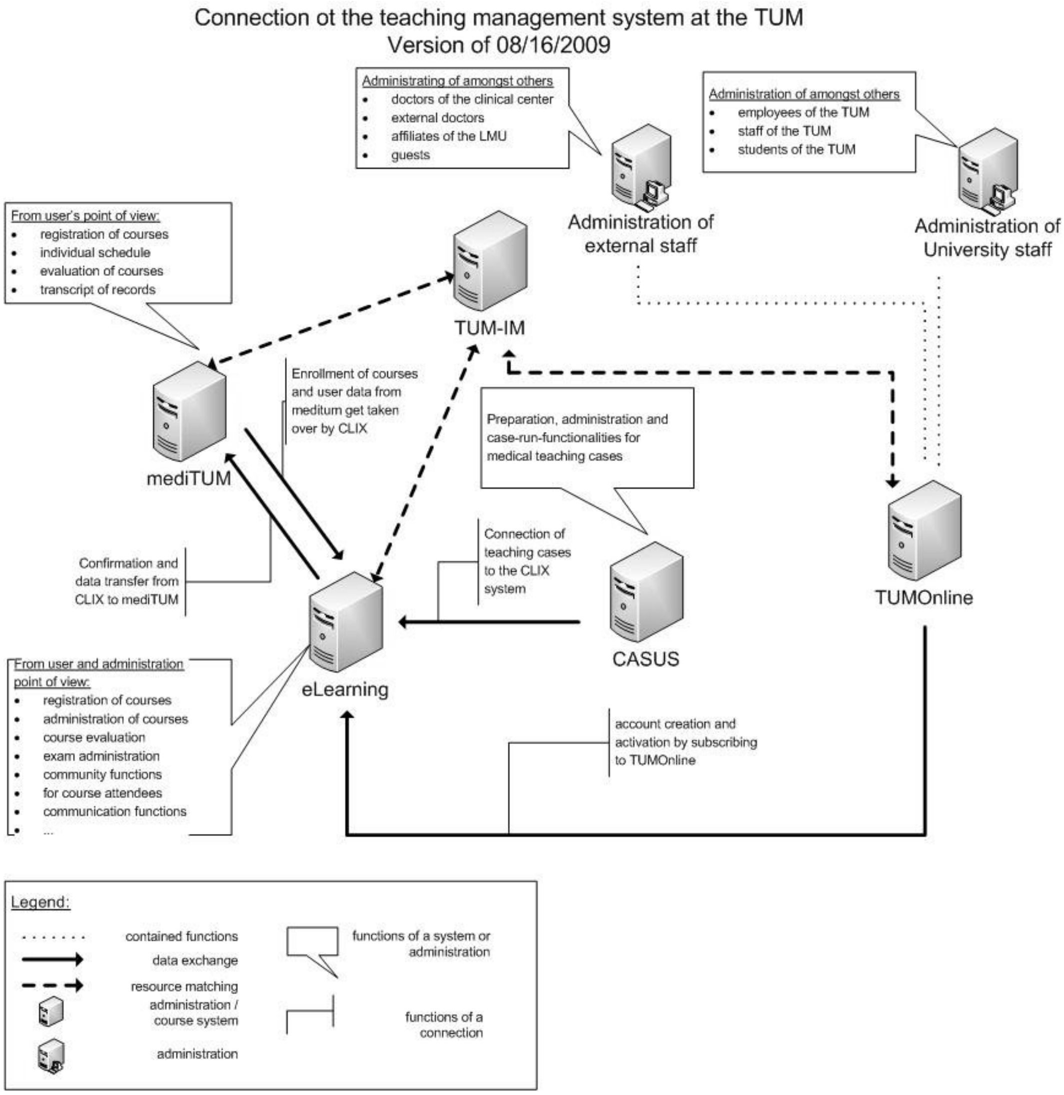

Figure 4.2: Connection of teaching management systems at the TUM

eLearning

ELearning is the central system for data transfers between *Casus* and *MediTUM*. From the user's perspective *eLearning* is used for the assessment of the courses. From the administration's point of view it controls and manages the courses and the registrations for them. It manages the users for each course and controls the usage of the *Casus* interface as well as it receives *Casus* meta data for each file, which is initiated manually.

While finishing a Casus case the system sends back the case data including the overall score of participants. As *eLearning* is not only accessible for TUM members but also for members of the LMU and others, *eLearning* also controls the usage rights for *Casus* via *Shibboleth*, an middleware for federated identity-based authentication and authorization infrastructure based on the Security Assertion Makeup Language. As *MediTUM* students need to access *Casus* cases by using *eLearning* the evaluation of the cases has to be transmitted not only from *Casus* to *eLearning* but also to *MediTUM*. eLearning therefore automatically transmits *Casus* data to *MediTUM* after a case has been closed.

Casus

The Casus system was added to the TUM teaching systems as it offers the possibility to conduct case studies. These are referenced to the studies contained in the different courses in *eLearning* and *MediTUM*. Prepared and administrated by tutors, these cases can be run by students. As not only registered members of *MediTUM*, but also guests and further participants need to access *Casus* cases, the system was connected to *eLearning* and can be linked to the according courses. This way, all the users of *TUMOnline*, which controls the whole member administration, can use *Casus* cases easily. A *SCORM/AICC-HACP* respectively a *Shibboleth* interface is used for the authentication of the users to perform data transmission. This interface manages the user identification via the *TUM-IM* as well as other identity management systems from other establishments.

MediTUM

Within the *MediTUM* system several functionalities got implemented, which are only used by the medical students. Therefore, the system was not phased out, but was connected to *eLearning* to maintain the existing functionalities for the students and tutors. The users have to register themselves independently from *TUMOnline* or *eLearning*. When a new course is registered in *MediTUM* this course gets automatically registered in *eLearning*. The users who already assigned for that course get also registered within *eLearning* and get access to the provided data. Within *MediTUM* the students do have additional options, which are not provided in *eLearning*, such as creating an individual schedule or the creation of a transcript of records. *MediTUM* was connected to *eLearning* via JMS-Middleware and ERP2CLIX interfaces, which secure the connection and also verify the systems among each other. This connection runs completely automatically. As the communication runs one way with *Casus*, *eLearning* sends and receives data from *MediTUM*. The data consists of the notification and the rating of finalized *Casus* cases. Instead of saving the ratings *eLearning* stores the cases that has been run before. This transmission is basically necessary for the final grade of the course. So the grades are only saved within *MediTUM*. *eLearning* receives metadata of courses and course assignments, which are automatically transferred into the database. When logging into *eLearning*, the courses already got integrated to the systems and the users, who registered for a course in *MediTUM* also are registered in *eLearning*. An additional secondary registration is not necessary.

4.3 Capturing user demands

We conducted an online survey to evaluate the user's perspective of the integration of the three systems. The survey contains 18 questions classified in seven groups. These are text based questions as well as multiple choice questions. Every multiple choice question was supplemented with a text area, which additionally allowed the respondent to enter his opinion. In total 102 responses were received until the completion of this paper. 68 of these were full responses and 34 were incomplete. Even some of the incomplete responses were useful to gain information on the customer's behavior, so they were partially integrated into the analysis. The survey was online for six weeks.

4.3.1 Participants

As the audience of the survey as well as the *eLearning's* user base consisted of mostly German native speakers the survey was conducted in German. The survey was promoted at the *eLearning system* and our project partner supplementary promoted the survey via e-mail to students and lecturers already using the *eLearning* system. Furthermore, the student council was asked to send out the survey among the medical students and lecturers. Therefore, the survey was promoted to a large but unknown number of medical students.

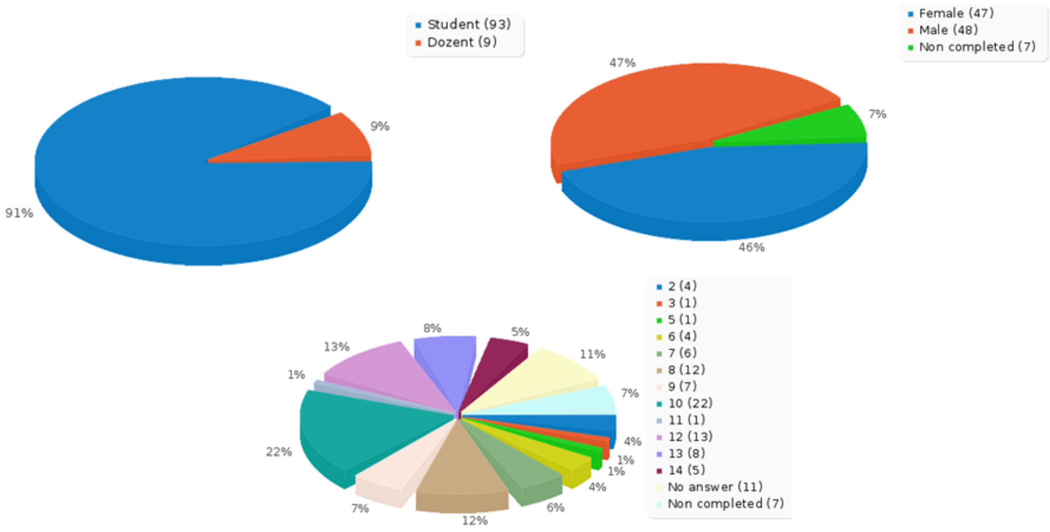

Figure 4.3: The survey's participants[2]

As shown in Figure 4.3 the respondents were mainly students with genders and semester of study distributed equally.

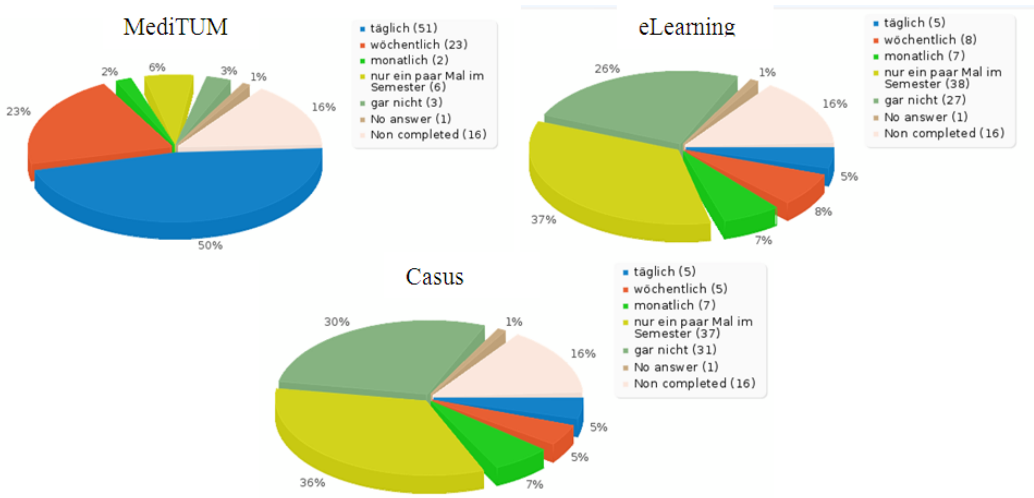

Figure 4.4: Usage rates of the systems analyzed[3]

Furthermore, the systems' usage rates were evaluated. While *MediTUM* is nearly used daily or at least weekly by the respondents, *Casus* and *eLearning* are only used on a monthly basis.

[2]Translation: Student = student and Dozent = lecturer

4.3.2 Questions for MediTUM

The first question in this group was about "which functions do you use most at *MediTUM*?". 68 people gave a valid response to this question. The most popular functions are the download of lecture materials (70 %), the curriculum (63 %), and the grade module (47 %). Also, news and information (32 %) as well as lecture evaluations (23 %), and the forum (23 %) play an important role at *MediTUM*. The calendar (10 %), the module booking system (7 %), and the *MediTUM* support (10 %) were also mentioned. The Question "Which functions do you miss at *MediTUM*?" was answered by 46 respondents. Even though there were some individual wishes of new functions the most respondents were quite satisfied with the functions already provided, although some claimed that the overall organizational structure could be laid out clearer.

4.3.3 Questions for eLearning

69 answers were given to the question "do you think that enriching a lecture with online materials is useful in order to improve teaching?". Nearly all respondents claimed "yes". Just a few respondents warned that the online materials should not replace practical parts of the education and therefore preferred more video and practical staff on the platform.

Asked for problems while using *eLearning*, 54 respondents answered in total. 24 among them had never suffered problems with *eLearning* while 11 didn't use it at all. Two problems were listed often: the low speed of the website and its complicated structure.

Furthermore, the lecturers were asked, if they suffered issues while creating lectures with *Casus* cases and how they sense handling the system. Three out of four sensed it simple and one said it is feasible, none felt that the administration is too difficult. However, been asked for new functions and improvements, the respondents still wanted the system to be simplified and more straight forward in usage.

4.3.4 Questions for Casus

The attendees were asked "how they sense the usage of the *Casus* system?". 13% sensed it very easy in usage, 27% as easy or adequate and only 5% as hard or too hard. Contrastingly, they felt that the transition from *eLearning* into *Casus* is complicated or even too complicated (28%). 25% said it is adequate and only 19% thought it is easy or very easy.

4.3.5 Concluding questions

The participants were asked for common wishes, recommendations, and critics from their point of view. We received only 28 valid answers. The most represented wishes were: "bigger amount of lectures", "easier to use", and "integration of student body (e.g. lecturer notes)".

When asked "how well are you supported by the current IT-systems in medicine" over 80% felt that the systems make life easier (cf. Figure 4.5).

Finally, we asked about the satisfaction of the systems' integration. Over 60% appreciated the integration, 20 % were not sure and only very few (6%) disliked it (cf. Figure 4.6).

[3] Translation: täglich = daily; wöchentlich = weekly; monatlich = monthly; nur einmal pro Semester = once per semester; gar nicht = never

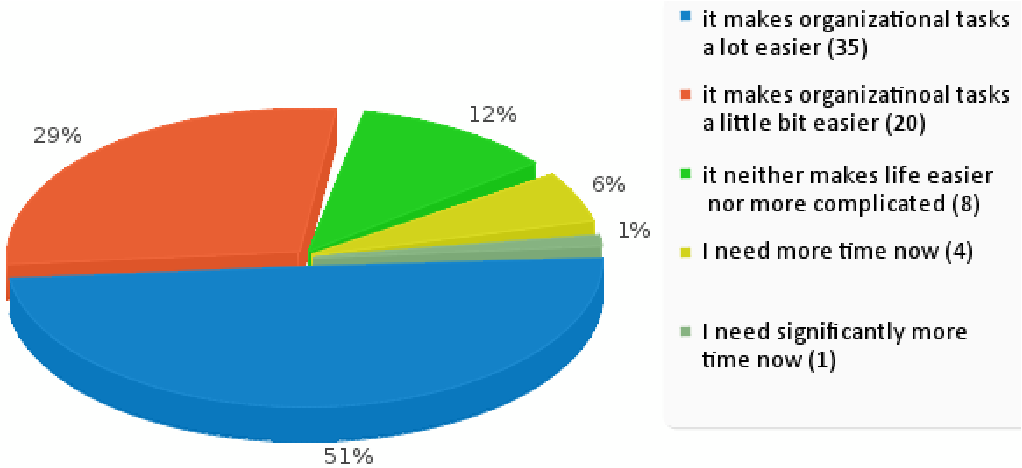

Figure 4.5: Revealed support through using the IT-systems in medicine studies

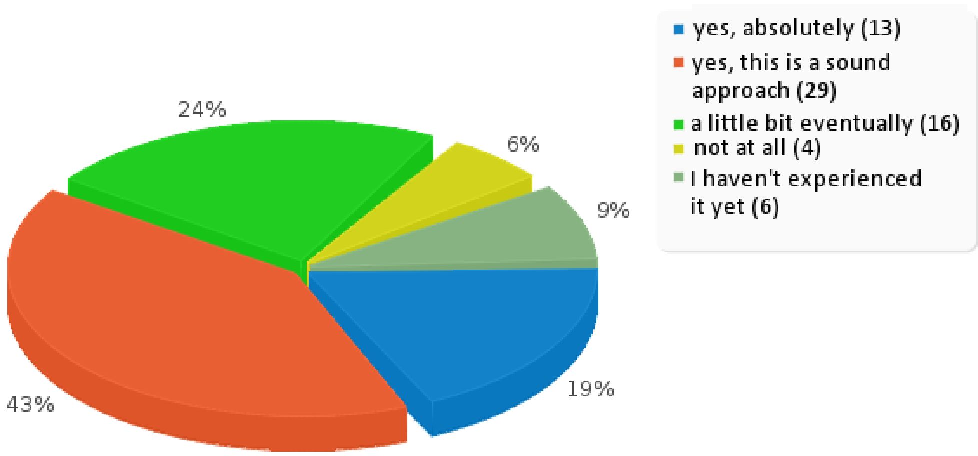

Figure 4.6: surplus of the system integration

4.4 The usage of the EAMPC

To get a consistent documentation of the systems, we used the EAMPC provided by the chair of Software Engineering for Business Information Systems (sebis) at TUM. We used the M-patterns "*Management of Business Objects*" (V 48) and adapted it to be usable in managing the integration of information systems. In addition, we created a new pattern "*Survey based analysis of system integration*" to provide a method for the questionnaire based evaluation of system integrations.

4.4.1 Management of business objects in context of the integration of information systems

The M-Pattern *Management of Business Objects* (M 19, [Bu08]) addresses different concerns regarding the dependencies and usage of business objects. We defined a business object together with our project partner as "every data collection object exchanged throughout the applications". A business object can be used by many applications of the application landscape. Therefore, business objects are relevant to EAM. The *Management of Business Objects* is especially important in an integration process of systems to identify redundancies and to get a starting point for integrating the applications.

Example

The three systems *MediTUM*, *eLearning*, and *Casus* were integrated. Now it is important to know which system has access to which business object and which business objects are exchanged (this refers to concern C-51 of the EAMPC: *Which Business Objects are used or exchanged by which Business Applications or Services?*). This knowledge could help planning the further integration process.

Context

After an integration of multiple information systems a documentation of the exchanged business object is required. This enables monitoring of the integration process and planning further integration process of the information systems.

Problem

You want to obtain an overview about the business objects exchanged between the systems to identify redundancies and to get a starting point for planning the next steps for further integration. The important question is: How do you document the business objects exchanged by your systems?

Solution

We discussed with our project partner how to define a business object. Then, we identified the business object using the relevant literature ([SWS09, S.08]), Subsequently, we described the systems and their interaction and reviewed our finding with our project partner. Finally, we were able to relate the business objects to the systems. We used a *cluster map* to visualize the business object flow. It is important to consult your project partner frequently throughout the process to get a complete list of business objects.

4.4.2 Cluster map visualizing business object flows between information systems

We used *Cluster Map visualizing Business Object Flows between Business Applications*, which is described in the V-Pattern 48 [Bu08]. This cluster map has symbols for business objects and systems and is able to illustrate the direction of the information flow between systems and information objects. The original cluster map also describes the type of interface between the systems and the business objects. Valid values are *manual*, *online*, and *offline*. In our case, we only have online interfaces, so we didn't use this classification. Also, we used a variant of the cluster map, which includes persons. This enables us to include different stakeholders of the system. A cluster map of our system is shown in Figure 4.7.

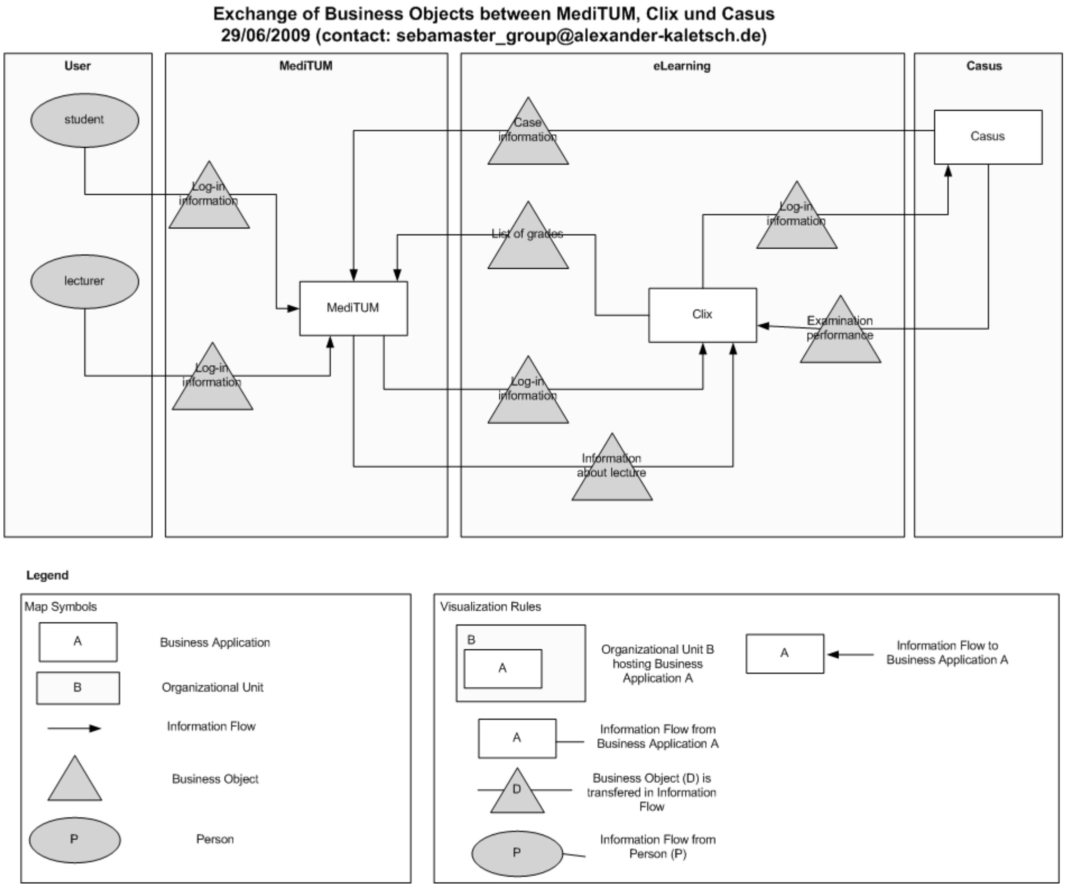

Figure 4.7: Exchange of business objects between MediTUM, eLearning, and Casus

Implementation

Views according to this viewpoint can be created manually by any drawing tool, like e.g. Microsoft Visio. However, a manual creation is time consuming and error prone. Therefore, we recommend a tool, which can automatically create such visualizations.

Consequences

Completeness versus readability: When you have got a huge application landscape, it might be very difficult to read a cluster map representing the whole application landscape. A solution could be to split the overall cluster map into smaller pieces where you can model the exchange of objects in detail.

Belonging versus planning: You have to select your focus first when using this pattern: Do you want to model the as-is state or the planned state? If you want to include both aspects, it will have negative effects on the readability of the resulting visualization. A solution would be to use two different layers to model the respective information.

4.4.3 Survey based analysis of system integration

The M-Pattern *Survey based analysis of system integration* is concerned with the evaluation of a system integration from a user perspective. If you have completed the system integration and you want to find out if the integration is not only successful from a technical perspective, but also from a user's point of view. This is especially important to collect requirements for a further integration process and therefore relevant to EAM.

Example

The three systems *MediTUM*, *eLearning*, and *Casus* were integrated. Now it is important to know if the integration was successful from a user's perspective. In order to determine this, a survey should be conducted asking questions on the usage and the opinion on the system integration.

Context

After an integration of two or more systems the user has to deal with a *new* system. It is important to know, if the users profit from the integration of the systems. If this is not the case the integration has not been useful. Conducting a survey can reveal the users opinion on the integration.

Problem

You want to evaluate the integration process to determine, if the integration of systems was successful and get a starting point for planning a further integration steps. The important questions are: "Has the integration process been successful from a user's perspective?" and "What parts of the system could be improved?"

Solution

We talked to our project partner to determine which questions are important to reveal the success of the integration. Also, we discussed who should participate in the survey and how we could reach them. We then created a proposal for the survey, which was reviewed and refined by our project partner. A good way to reach the relevant participants is to contact them directly throughout the integrated systems. However, if that is not possible you maybe have a user database you could reach via e-mail. Furthermore, a good idea is to use a non-technical promoter to motivate the users to participate in the survey. In our case, we used the student council to promote our survey. Following are some exemplary questions:

How often do you use the specific system? This question could provide you with hints how successful the integration of systems was. If the system is not used frequently the integration might have not been very successful. However, it is useful to have statistics about the usage from the old systems to compare the results and determine, if the usage has increased, decreased, or stayed at the same level.

Which functions do you use most at a specific system? The answers to this question tell you what the most important systems are and what systems maybe don't provide any relevant functionality. This could be helpful for a further integration process.

Do you profit from the integration of the systems? This is the key question because you get the subjective opinion on the system integration from the users. If a big part of the users don't profit from the integration the integration might not have been very successful.

Implementation

We used the open-source software *LimeSurvey*[4] for the implementation of our survey. However, there are a lot of different tools for conducting surveys like *polldaddy*[5] or *2ask*[6]. Important factors for the selection of a survey tool are the complexity to use it and the ability to export the results in common formats like *Microsoft Excel*.

Consequences

Completeness vs. answering time: The users would probably not answer too many questions. If the survey takes too long to be filled out, it is likely that the user will abort it after a specific time. Therefore, you should only ask the most important questions and keep the text and descriptions short.

All users vs. user group: Sometimes it is not necessary to use all users of the system as participants for the survey, whereas, only a small group profit from the integration. Then, it is better to ask only this group because the whole user base could falsify the results of the survey.

4.5 Outlook and acknowledgement

Considering the models we created and the results of our survey, we can state that the integration of the systems is widely done in a technologically sense. A single-sign-on method is implemented and the operators were already aware of their systems configuration. Now, they are also endowed with detailed graphical models, which allow them to gain a better overview of their systems.

The survey showed that the integration of the systems is currently not well adopted by the users and some systems are only rarely used. This results from the fact that on the one hand the integration is quite new and not known to all users by, on the other hand the users think that some parts of the systems are too difficult to use.

The outcome of this project will be further evaluated by the IntegraTUM team and then used in their own research in order to improve the integration of *eLearning*, *MediTUM*, and *Casus*. There will also be other surveys in order to improve knowledge about the user's needs.

The new EAM pattern *Survey based analysis of system integration* was created by generalizing from the accomplished project. However, it must be evaluated, if it is also useful for other integration projects before it can be included into the EAMPC.

We want to thank all participants of our survey. Also, we want to express our gratitude to our industry partner and the IntegraTUM team for the great support we received in every aspects of our project.

[4] LimeSurvey can be downloaded from http://www.limesurvey.org/.
[5] Polldaddy can be found at http://polldaddy.com/.
[6] 2ask can be found at http://www.2ask.net/.

Bibliography

[Bu08] Buckl, S. et al.:. *Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008)*. Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2008.

[Ch09] Chair of Informatics 19 Technische Universität München:. *Lecture: Software Engineering for Business Applications - Master Course*. online. 2009.

[S.08] S., G. et al.:. *eLearning als Teil einer serviceorientierten Hochschulinfrastruktur*. *DeLFI 2008*. pages 65–76. 2008.

[SWS09] S, G.; W, H.; S., R.:. *Hochschulübergreifendes E-Learning: Technische Realisierung und Datenschutz*. *Proceedings der 9. Internationalen Tagung Wirtschaftsinformatik*. 2009.

[Wi07] Wittenburg, A.:. *Softwarekartographie: Modelle und Methoden zur systematischen Visualisierung von Anwendungslandschaften*. PhD thesis. Fakultät für Informatik, Technische Universität München, Germany. 2007.

CHAPTER 5

Modeling and Documentation of IntegraTUM's System Landscape

David Ciechanowicz (david.ciechanowicz@mytum.de),
Philipp Donie (philipp.donie@mytum.de),
Tomislav Pravidur (tomislav@mytum.de),
and René Tiede (rene.tiede@mytum.de)

Abstract:
The IT landscape in university environments is often characterized by complex and heterogeneous systems. All the faculties, schools, and departments belonging to the same university often can act on their own authority. However, the goal to provide a single interface and standardized services to customers might exist. In the context of an integration and consolidation initiative at Technische Universität München (TUM), the project *IntegraTUM* has been established. This led to new challenges, e.g. how to document and manage the different components. It would be beneficial to identify and document all the responsibilities, processes, systems, data flows, and services. This article presents an approach to introduce and apply enterprise architecture management methods, to solve the aforementioned problems. The approach contains a comprehensive review of existing documentation and the development of an information model. The results are visualized by specific system/software maps.

5.1 Motivation

This article is about a project as part of the master course *Software Engineering for Business Applications (SEBA)* at the Technische Universität München (TUM). The objective of the project is to identify or develop a solution for specific EAM concerns in a practical setting. The project ist conducted in cooperation with participating companies by using methods of *Enterprise Architecture Management (EAM).* The client of the project *Modeling and Documenting of IntegraTUM's System Landscape* is the Leibniz-Rechenzentrum (LRZ), IT service provider for the TUM.

The remainder of the article is structured as follows: Section 5.1 and 5.2 of this report list the identified problems and describe the developed approach. The IT architecture of the LRZ's IT landscape, designed during the gathering of information at the beginning of the project, are described in Section 5.3. Section 5.4 and 5.5 depict the modeled software maps as well as the information model and describe their development. Section 5.6 elaborates on the SoCaTool of the Chair for Informatics 19 (sebis), which allows the automated generation of software maps. The last section, conclusion and future work, considers the project's results and provides an outlook on future work.

The structure of this document fits exactly the real proceeding of the performed project and reflects the project course. The decision for this structure was made to avoid redundancies. Some of the designed software maps are used during the solution of several concerns.

5.1.1 Project overview

This project belongs to the long-term program named *IntegraTUM* at the LRZ. *IntegraTUM* is separated into nine subprojects. One of them deals with establishing a central service directory. The main goal of this proposition is to provide a central identity and access management for the TUM. This covers various aspects, like a central identity repository, single sign-on, and centralization of services. A user should be granted access to any available IT service of *IntegraTUM* with only one single user account.

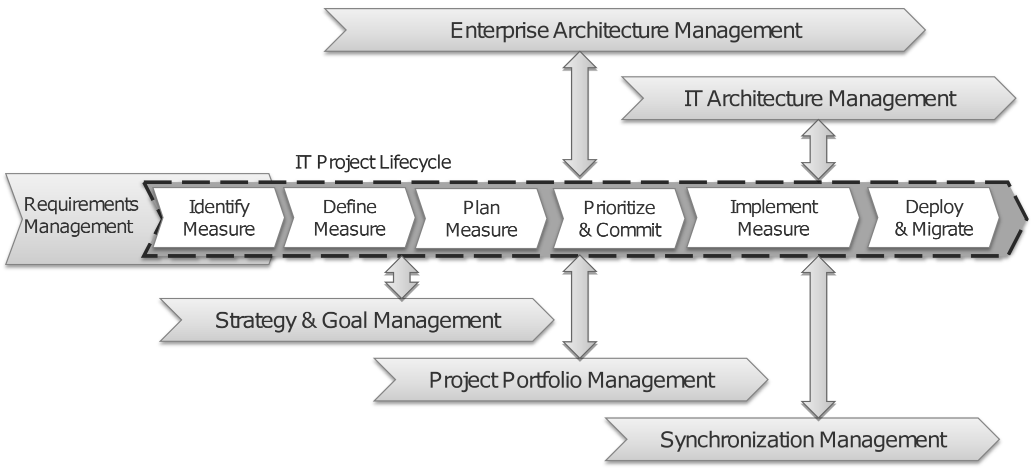

Figure 5.1: IT project lifecycle with enterprise level management processes.

The activities of the project can be located in the framework of Wittenburg [Wi07] as shown in Figure 5.1 in the process of EAM.

5.1.2 Problem definition and concerns

The described problems and concerns were identified in close cooperation with the responsible persons of the LRZ. In relation to the *EAM Pattern Catalog* of the sebis chair most of the identified concerns, e.g. C-33: *Which applications are used by which organizational units*?, can be assigned to the EA Management topic *Application Landscape Planning*. "Application Landscape Planning is concerned with planning and analyzing the structure and evolution of the application landscape, focusing on current, planned, and target landscapes." [Bu08]. Some concerns, e.g. C-98: *What is the impact of the shut-down of an infrastructure element? What other elements of the application landscape are affected?*, can be assigned to the EA Management topic *Infrastructure Management*. By means of existing documentations and project records the status quo of IntegraTUM was determined. By further analysis of this information and interviewing the responsible stakeholders problems could be identified and delimited.

This project was initiated due to the appearance of different problems, which can be summarized as a lack of transparency. To be able to control the IT landscape of the LRZ, it is necessary to develop a comprehensive and complete documentation of its landscape. The affected IT landscape of the LRZ consists of components like *systems*, *target systems*, *source systems*, and *services*. Systems and target systems perform the same tasks but the difference is that an end-user cannot interact with a system, whereas he can interact with a target system. A source system is an element from which data is loaded into a central data pool. A service is an IT supported, self-contained functionality, which is offered by a system. It exchanges data by means of interfaces and complies specific features to other systems or users. Further components are *data flows*, *responsibilities*, *processes*, and different *interdependencies* between these components. A missing general view on this IT landscape entails several problems. The troubleshooting is very complicated because of missing knowledge about which system causes which problem. Furthermore, it is often unknown who is responsible for which system and consequently the point of entry and contact in case of trouble. Problems also occur in trainings and workshops, which should import knowledge on the IT landscape.

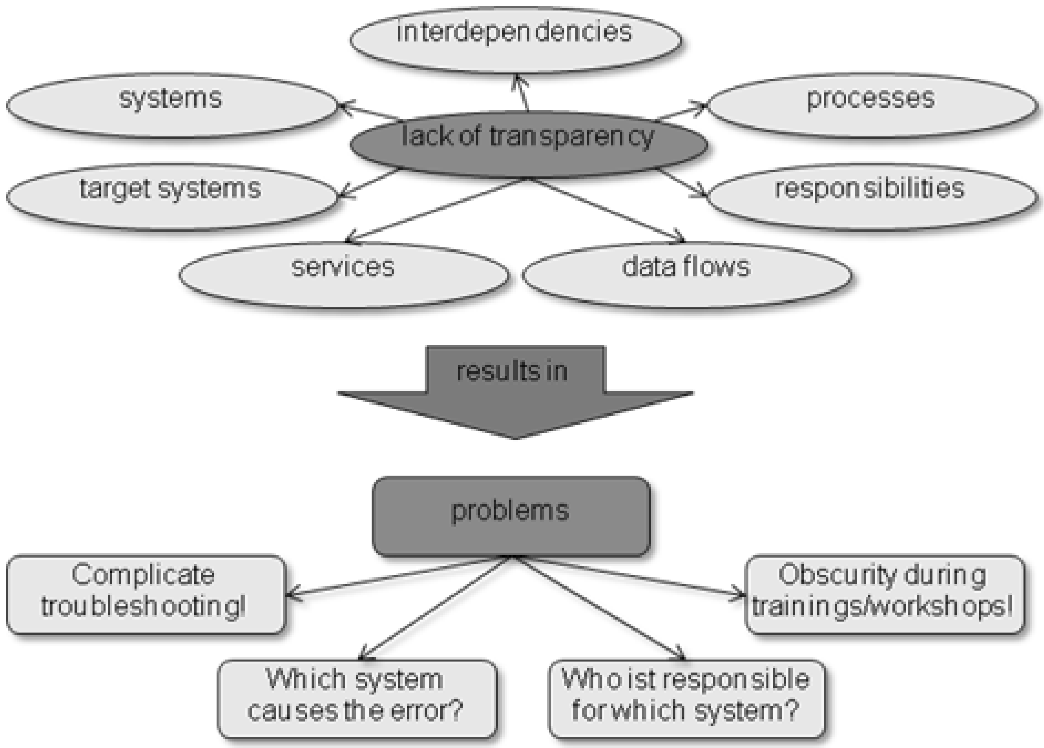

Figure 5.2: Problems and concerns of *IntegraTUM*.

5.2 Problem approach

Before explaining the approach the final project plan is exemplified. First, the project allocation was performed by the sebis chair, in the way that every project was assigned to four students. The second step was the kick-off meeting followed by the requirements analysis with the LRZ. In this phase of the project environment, status, concerns, requirements, and goals were acquired and analyzed. After the project was defined and first basic deliverables were developed these results were prepared for the intermediate presentation and introduced to the other participants of the SEBA lecture and their industry project partners. After fulfilling this first milestone the solution development as the main part of the project took place. This step is described in the next paragraph. After finding and implementing a solution for the LRZ's concerns the final presentation of the project was prepared. The last five points of the project plan apply to the project report, which was created in a first version, reviewed, and finally finished.

		Vorgangsname	Dauer	Anfang	Ende	Vorgänger	Ressourcennamen
1		**Whole Project**	**88,02 Tage**	**Mi 06.05.09**	**So 30.08.09**		
2	✓	**Project Allocation**	**0,2 Tage**	**Mi 06.05.09**	**Mi 06.05.09**		
3	✓	Adoption of Project Responsibility	0,2 Tage	Mi 06.05.09	Mi 06.05.09		David;Philipp;René;Tomislav
4	✓	**Kick-Off Meeting**	**0,1 Tage**	**Fr 08.05.09**	**Fr 08.05.09**		
5	✓	Acquisition of Environment, Goals & Status	0,1 Tage	Fr 08.05.09	Fr 08.05.09	3	David;Philipp;René;Tomislav
6	✓	**Status & Requirements Analysis**	**6 Tage**	**Sa 09.05.09**	**So 17.05.09**		
7	✓	Analysis of Environment, Goals & Status	6 Tage	Sa 09.05.09	So 17.05.09	5	David;Philipp;René;Tomislav
8	✓	**Preparation of the Intermediate Presentation**	**2 Tage**	**Mo 18.05.09**	**Di 19.05.09**		
9	✓	Processing: Client & Concerns	1 Tag	Mo 18.05.09	Mo 18.05.09	7	Tomislav
10	✓	Processing: Status & Architecture	1 Tag	Mo 18.05.09	Mo 18.05.09	7	David
11	✓	Processing: Pattern Matching	1 Tag	Mo 18.05.09	Mo 18.05.09	7	Philipp
12	✓	Processing: Time Scheduling	1 Tag	Mo 18.05.09	Mo 18.05.09	7	René
13	✓	Creating, Shaping & Finalizing the Intermediate Present:	1 Tag	Di 19.05.09	Di 19.05.09	9;10;11;12	David;Philipp;René;Tomislav
14	✓	**Conduction of the Intermediate Presentation**	**0,05 Tage**	**Mi 20.05.09**	**Mi 20.05.09**		
15	✓	Presentation	0,03 Tage	Mi 20.05.09	Mi 20.05.09	13	David;Philipp
16	✓	Discussion & Questions	0,02 Tage	Mi 20.05.09	Mi 20.05.09	15	David;Philipp;René;Tomislav
17	✓	**Solution Development**	**38 Tage**	**Do 21.05.09**	**Sa 11.07.09**		
18	✓	Finding & Balancing of Possible Solutions	19 Tage	Do 21.05.09	Di 16.06.09	16	David;Philipp;René;Tomislav
19	✓	Modelling of the Software Patterns	19 Tage	Mi 17.06.09	Sa 11.07.09	18	David;Philipp;René
20	✓	Introduction to the SoCa Tool	3 Tage	Do 02.07.09	Mo 06.07.09	19	David;Philipp;René;Tomislav
21	✓	Working on the Information Model	6 Tage	Mo 06.07.09	Sa 11.07.09	20	David;Philipp;René
22	✓	**Preparation of the Final Presentation**	**3 Tage**	**So 12.07.09**	**Di 14.07.09**		
23	✓	Creating, Shaping & Finalizing the Final Presentation	3 Tage	So 12.07.09	Di 14.07.09	21	David;Philipp;René;Tomislav
24		**Conduction of the Final Presentation**	**1 Tag**	**Mi 15.07.09**	**Mi 15.07.09**		
25		Presentation	0,02 Tage	Mi 15.07.09	Mi 15.07.09	23	David;Philipp;René
26		Discussion & Questions	0,02 Tage	Mi 15.07.09	Mi 15.07.09	25	David;Philipp;René
27		**Preparation of the Elaboration**	**20 Tage**	**Do 16.07.09**	**Mi 12.08.09**		
28		Creating, Shaping & Finalizing the Elaboration	20 Tage	Do 16.07.09	Mi 12.08.09	26	David;Philipp;René;Tomislav
29		**Due of the Elaboration**	**0,02 Tage**	**Do 13.08.09**	**Do 13.08.09**		
30		Handover of the Elaboration	0,02 Tage	Do 13.08.09	Do 13.08.09	28	David;Philipp;René
31		**Review & Feedback**	**0,1 Tage**	**Mo 24.08.09**	**Mo 24.08.09**		
32		Review & Feedback at sebis	0,1 Tage	Mo 24.08.09	Mo 24.08.09	30	David;Philipp;René;Tomislav
33		**Finalizing the Elaboration**	**5 Tage**	**Di 25.08.09**	**Sa 29.08.09**		
34		Creating, Shaping & Finalizing the Final Elaboration	5 Tage	Di 25.08.09	Sa 29.08.09	32	David;Philipp;René
35		**Due of the Final Elaboration**	**0,02 Tage**	**So 30.08.09**	**So 30.08.09**		
36		Handover of the Final Elaboration	0,02 Tage	So 30.08.09	So 30.08.09	34	David;Philipp;René

Figure 5.3: Final project plan.

This paragraph specifies the solution development, listed in the project plan, as main part of the project. The identified concerns of the LRZ were approached as shown in Figure 5.4. The first step was to identify the problems as mentioned in paragraph 5.1.2. There was insufficient documentation about the system landscape, the used target systems, the location where the target systems are installed, and the responsibilities for the systems and target systems available. Thus the second step was the gathering of necessary information about the IT landscape of the LRZ. Even if this task sounds trivial it was the most important part of the project and it took most of the time. The project's goal was the comprehensive design of a model depicting the IT landscape, which is only possible with a sufficient set of correct information. Consequently, all hidden information had to be identified and integrated with assistance of the stakeholders of the LRZ. Beside further assistance from other employees of the LRZ, the stakeholders provided the major part of the information, which had to be arranged and modeled as part of the current, planned, and target architecture. Due to bit by bit perceived questions and concerns the three processes – collecting necessary information, develop comprehensive system architecture, and design software maps and information models – as shown in Figure 4, had to be repeated frequently. The basis of information as well as existing models had to be updated during this proceeding. In the final phase of the project the problem rose that some information of landscape components lacked and nobody knew the person in charge. Due to the time pressure in this situation the only possibility was to define the current information status as complete. Only this proceeding allowed the development of the planned software maps for depicting different views of the LRZ's IT landscape. Nevertheless, due to the missing information, the software maps should be completed at a later date. The last two steps represent the modeling of architectures, software maps, and the information model. Complete models of the existing IT architecture were designed to be able to create an information basis for the software maps. Resting on these architectures, the planned software maps and the information model were developed as described in Section 5.4 and 5.5.

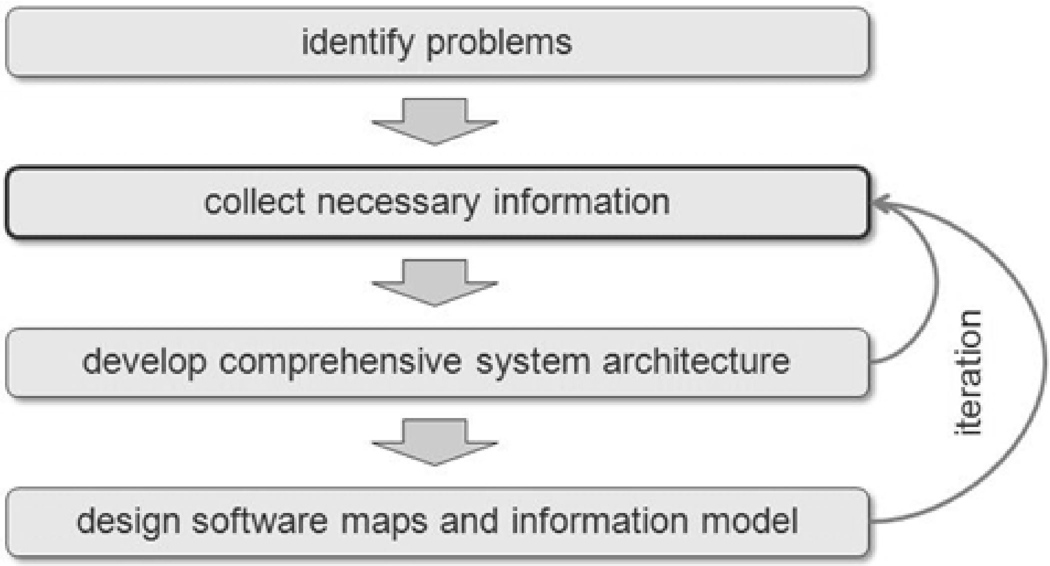

Figure 5.4: The project phases

5.3 System landscape

In this section an overview on the states of the system landscape at different points in time is given. At the outset, the as-is architecture as available at the project of the beginning is shown. Based on the problems, which occur, the vision for the end of the project is desired. The section ends with the current as-is architecture and a description of the gap analysis.

5.3.1 As-is architecture as of june 2004

Figure 5.5 depicts the as-is architecture before the project start in June 2004. It consists of the users, which are surrounded by different target systems. The user had to sign-on with a specialized identification for each target system he/she wants to use. Every target system in this situation is a stand-alone system, which does not necessarily need to be connected to any other system. If a new target system had to be integrated, this could be done with little effort and users could easily log-in with a new identification. Common data could be shared with a direct or indirect connection between the target systems.

The disadvantage of this solution is that there is no central data pool for common data of all target systems. In addition, the user had to sign-on to each target system with a different identification. The objective of the target architecture is to eliminate this weaknesses and to achieve a flexible, non-redundant, easy-administrable, and user-friendly system landscape.

5.3.2 Target architecture – a vision for october 2009 as of june 2004

In Figure 5.6 the target architecture developed at the beginning of the project in 2004 is depicted. Today, it should be seen more as a vision than as a real architecture because it turned out that there are drawbacks, which are described later.

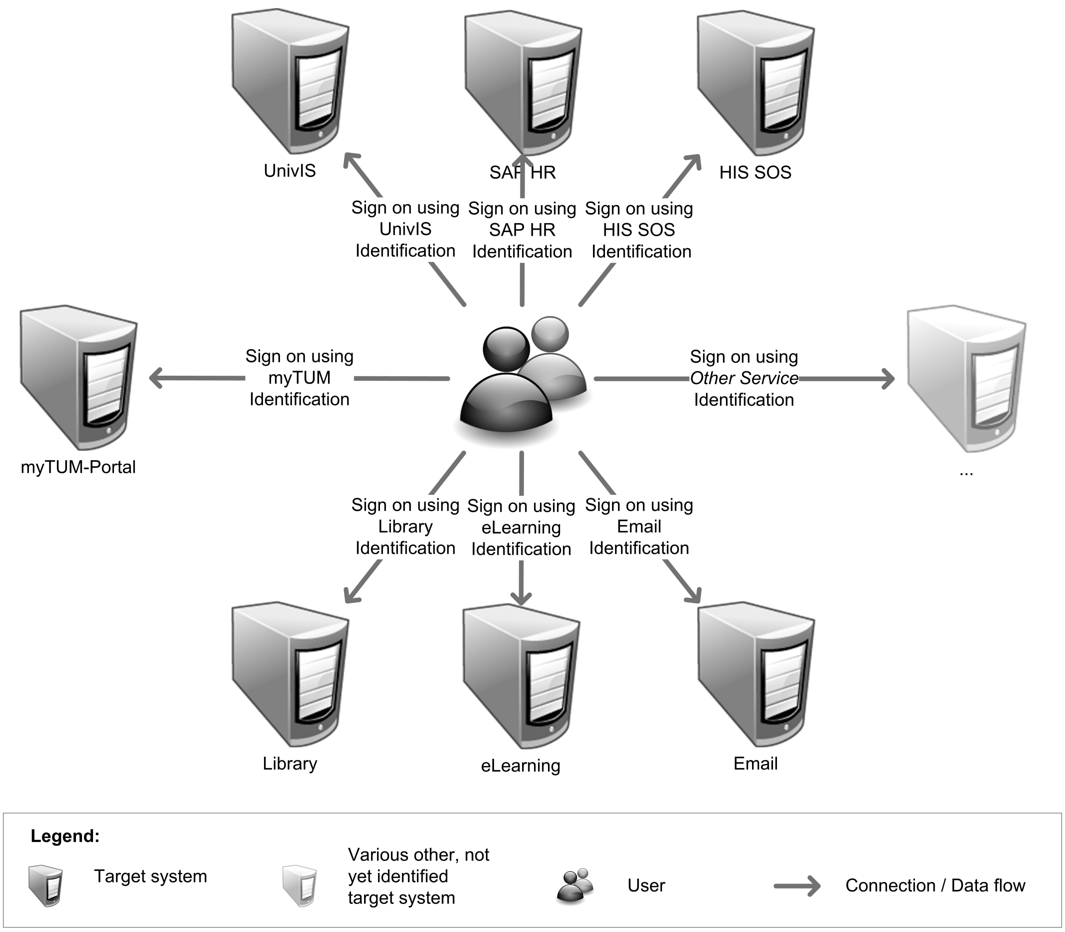

Figure 5.5: IntegraTUM: As-is architecture – The as-is architecture in June 2004
Last modified: 22nd of July 2009

The shown system landscape is sliced in the layers (a) data sources, (b) data pools, and (c) target systems. In (a) the two systems *SAP HR* and *TUMonline* exchange data. In the second layer only one central data pool should exist but because of its peculiarities *myTUM* has its own central data pool. In this envisioned architecture other target systems can have their own central data pool as well. The data from layer (a) is live-mirrored to layer (b). The target systems from layer (c) send their queries to the central data pool (b), which forwards the result again to the target systems (c). In addition, it is possible for a system to communicate directly with the source systems on layer (a) the way e.g. *eLearning* does. This happens whenever a target system requests data that is only used by this target system. Hence, the central data pool stores only data, which is used by more than one system. The user signs on with a generic identification (e.g. MWNID) to every target system, which then queries the sign-on request to the central data pool, which confirms the authentication of the user.

The advantage of this architecture is the possibility to flexibly integrate new target systems. Data which is used by more than one system is saved non-redundant in the central data pool. In addition, the user only needs one login name and appropriated password for all target systems, which results in a user-friendly system landscape. There is a possible disadvantage regarding the performance and availability bottleneck in this architecture. The reason for classifying this architecture as a vision and not as a real architecture is that it is not possible to realize it in exactly this way. The remedy for these problems is described subsequently.

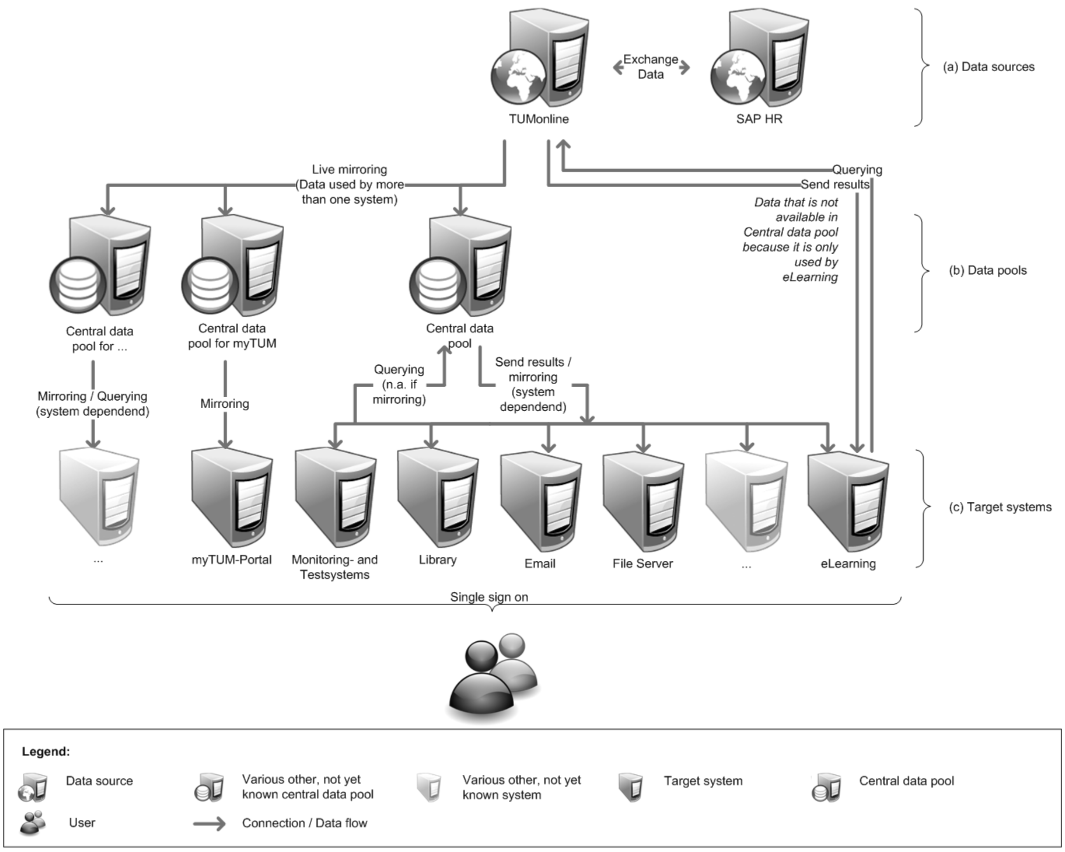

Figure 5.6: IntegraTUM: Target architecture – A vision for the as-is IT system architecture Last modified 22nd of July 2009

5.3.3 As-is architecture as of june 2009

Figure 5.7 depicts the as-is architecture in June 2009. Since this date is close to the end of the project the architecture can be considered as the final realized architecture.

Analogously to the architecture shown in Figure 5.6, the architecture in Figure 5.7 is divided in three layers (a) data sources, (b) data pools, and (d) target systems and has one additional layer (c) systems and load balancer. The latter includes the authentication server, the application satellite, the administration satellite, and the load balancer. All these systems are laid out redundantly to prevent a total failure in inter-system communication. In opposite to Figure 5.6 the target systems do not communicate directly with the central data pool but only through the layer (c). Every target system sends its queries to the load balancer, which forwards it to the responsible system. The authentication server has only the functionality of authenticating users. Similarly, the administration satellite treats administration queries only. The application satellite takes care of every other query that comes in from a target system. The authentication server, the administration, and the application satellite are holding the data from the central data pool or TUMonline, respectively. Thus, the target systems are not dependent on the systems of layer (a) or (b). Not all target systems communicate totally or only partly with other systems from layer (a), (b), or (c). These systems are shown in the bottom right corner of Figure 5.7. In the first case, the system is not part of the system landscape of IntegraTUM and only mentioned for interest. In the second case, a target system can only use the authentication server to provide the advantage of single sign-on and holds the user data redundantly. In this case the user signs-on with a generic identification to every target system.

The realized architecture solves the problems mentioned in the previous section of performance and availability bottlenecks through introduction of redundantly laid out systems and a load balancer.

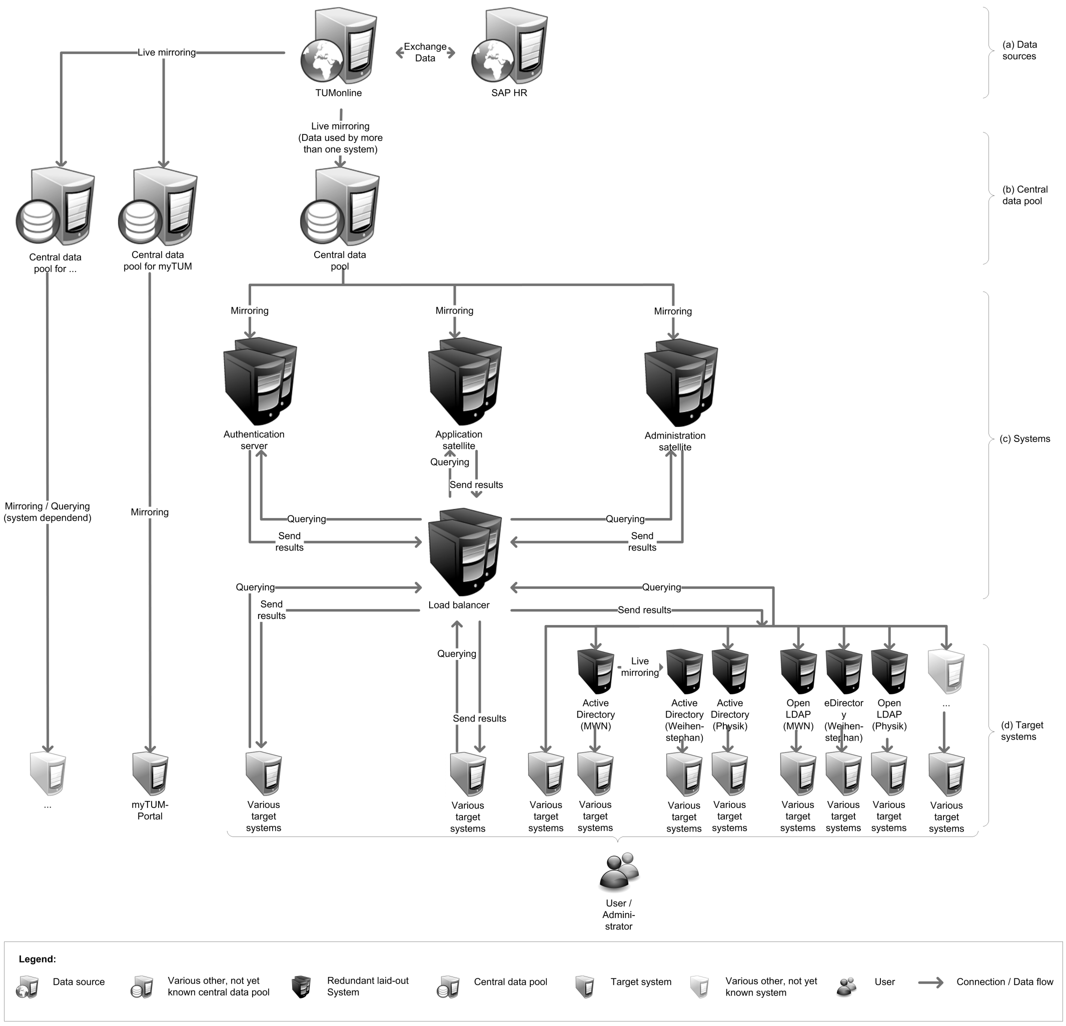

Figure 5.7: IntegraTUM: As-is architecture – The as-is architecture in June 2009
Last modified: 22nd of July 2009

5.4 Software maps

Software maps are used to provide a closer understanding both of the underlying architecture and the systems. Beyond that, they provide insights to relationships, interactions, responsibilities, or any other relevant information that is modeled. [Wi07]

This presents the software maps that were created and used in the project. They are categorized in two types: Cluster maps and cartesian maps, which are described in the following sections.

5.4.1 Cluster maps

In cluster maps components with one or more identical attributes are grouped to so-called *logical domains*. These compounds can be connected to show a relationship of any kind, for example hosted at, located at, exchanges information, or logically grouped. Domains can be nested into other domains. Thereby, cluster maps are generated with minimal size. Therefore, the positioning of the particular components is not specified expect for a nesting positioning. Individual rules can be defined in the legend. [Wi07]

The following three cluster maps, describing the IntegraTUM project, are explained exemplary in the next paragraphs.

5.4.1.1 Locations hosting systems

The cluster map in Figure 5.8 aims at the concern C-62: *What are the domains[1] of the application landscape?* of the EAM Pattern Catalog of the sebis chair (see [Bu08]). As recommended in the catalog for the related M-Pattern[2] M-20: *Management of Business Services and Domains* a cluster map according to the V-Pattern[3] V-55: *Component Cluster Map* was used to display the domains. More precisely, the map shows the locations where the systems and target systems of the IntegraTUM project are deployed at. The single locations are represented by gray outer squares, for example the datacenter LRZ in Garching. All systems are colored light gray, all target systems dark gray. Every system and target system is exactly deployed at one location. In the software map this is visualized as a nesting relationship. For instance the system TUMonline is hosted at LRZ Garching.

5.4.1.2 Systems' responsibilities

Because there is currently no concern on responsibilities for systems contained in the EAM Pattern Catalog C-62: *What are the domains of the application landscape?* with the related M-Pattern M-20: *Management of Business Services and Domains* and the included V-Pattern V-55: *Component Cluster Map* fits best again. [Bu08]

The cluster map shown in Figure 5.9 has an equivalent structure to the previous one in Figure 5.8. The light gray respectively dark gray rectangles display the systems or target systems. In contrast the outer squares define responsible institutions. An institution is responsible for all systems and target systems located in its particular outer square. Responsible in this context means to take over all actions concerning support and maintenance to prevent or solve problems in availability, occurrence of errors, bugs, and failures. The LRZ for example is responsible for the system TUMonline.

5.4.1.3 Interaction and data flows between systems and target systems

The software map in Figure 5.10 belongs to the concern C-98: *What is the impact of the shut-down of an infrastructure element? What other elements of the application landscape are affected?* of the EAM Pattern Catalog. The concern C-62: *What are the domains of the application landscape?* plays a subordinate role. As there are not all target systems depicted it can also be seen as a solution to concern C-34: *How does the long-term vision, the target of the application landscape, look like?*. Since the focus is more on the hierarchy and interdependency of the infrastructure than on possible consequences a cluster map is the optimal choice of representation. [Bu08]

[1] Domains can be seen as equivalent to locations in the context of TUM

[2] "A Methodology Pattern (M-Pattern) defines steps to be taken in order to address the given concerns" [Bu08]

[3] Viewpoint Pattern (V-Pattern) proposes a way to present relevant data[Bu08]

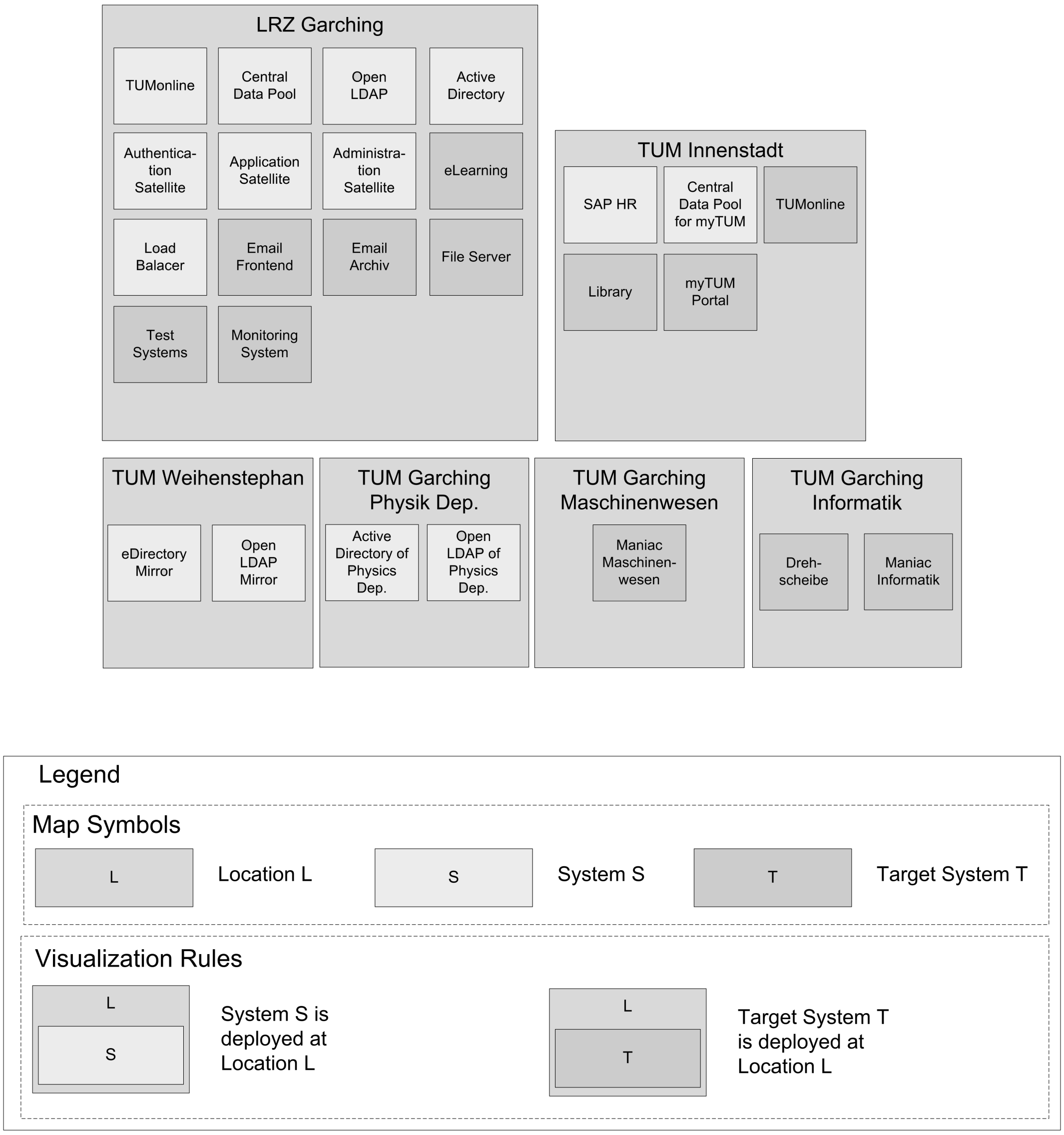

Figure 5.8: IntegraTUM: Infrastructure – Cluster map: Locations
Last modified: 27th of August 2009

The cluster map of interaction and data flows depicted in Figure 5.10 resembles the architecture landscapes. It focuses on the connections between systems and target systems. It provides a more technical view on the infrastructure than the aforementioned cluster maps.

The orange and purple rectangles represent the systems and target systems again. A specialty in this case is that not all target systems are displayed. Instead they are bundled in the appropriate purple rectangle labeled with "...". This was done to ease the readability because of the high number of target systems. All target systems clustered in one system interact with it, meaning there is a flow of operating data. The interactions concerning the flow of all master data between systems and target systems are displayed as (labeled) arrows. The labels specify the type of interaction. As there are many target systems connected with only few systems, a bottleneck can occur easily. To prevent the system landscape from a jam if one or more systems fail, they are construed redundantly. This is

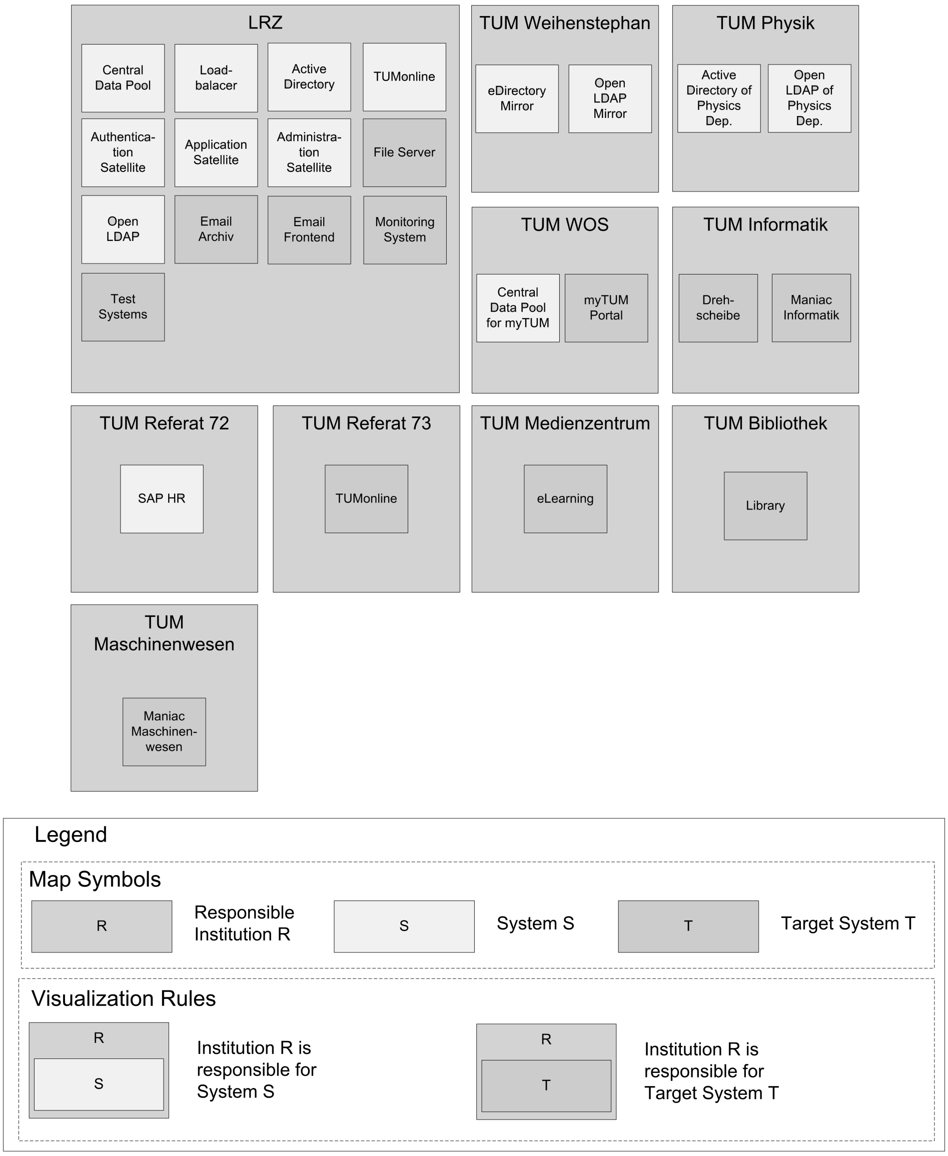

Figure 5.9: IntegraTUM: Infrastructure – Cluster map: Responsibilities
Last modified: 27th of August 2009

shown in Figure 5.10 as identically labeled black squares. A load balancer automatically switches between the redundant systems to enable ideal performance.

5.4.2 Cartesian maps

Every cartesian map consists of two axes: The x- and y-axis. Each of them is divided into intervals that define for example periods of time, business units, or divisions. All components on these axes are classified with at least one interval for each axis. [Wi07]. Axes can be freely defined but in general there exist two types of cartesian maps: process support maps and time interval maps.

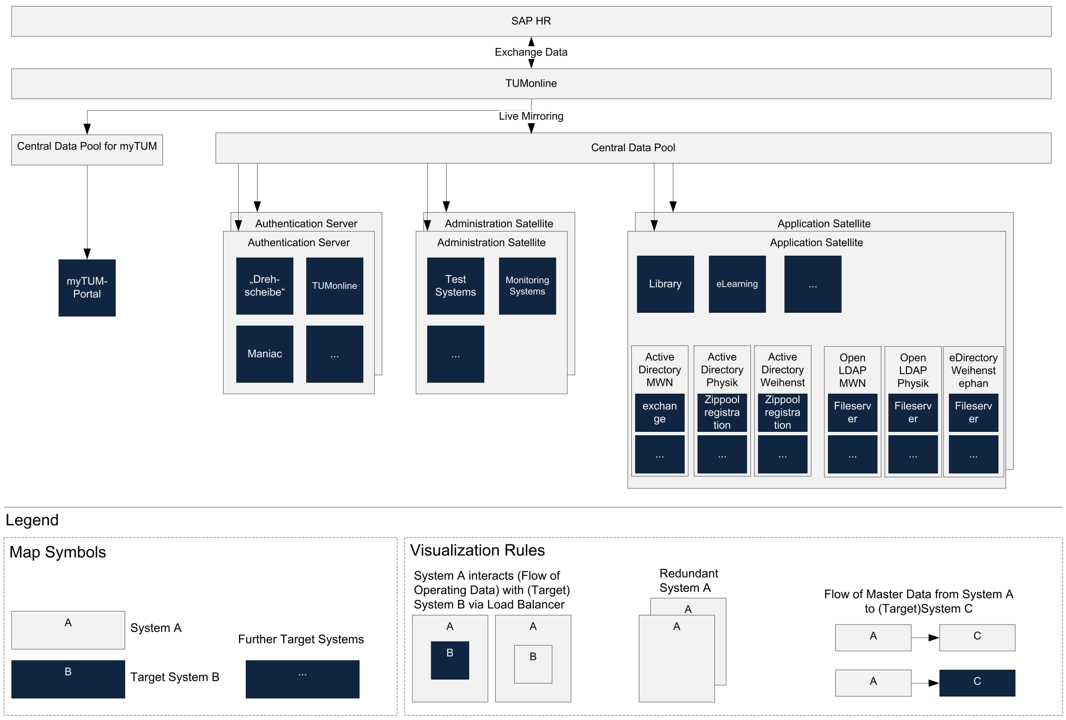

Figure 5.10: IntegraTUM: Infrastructure – Cluster Map: Interaction between different Systems
Last Modification: 18th of July 2009

5.4.2.1 Target systems

The main idea of this cartesian Map is to consolidate all information about particular target systems and services in one view. Obviously, several concerns are directed: C-33: *Which applications are used by which organizational units?*, C-86: *Which business applications are hosted by which organizational unit?* (whereas the usage and hosting of a target system or service is mainly done by persons of the same institutions), C-62: *What are the domains of the application landscape?*, and C-98: *What is the impact of the shut-down of an infrastructure element? What other elements of the application landscape are affected?*[Bu08]. There is a big variety of M-Patterns and included V-Patterns dealing with those concerns. The decision was made to design a cartesian map as shown in Figure 5.11.

The aim of the cartesian map in Figure 5.11 is to display the connected systems and responsible institutions for target systems and services. Especially for purposes of coordination, change management, or de-escalation of problems this map may positively contribute to all stakeholders. In cases of technical problems cartesian maps can help to rapidly determine responsible persons, physical locations, and connections of systems and target systems.

The black rectangles of the x-axis of the map show the systems of the IntegraTUM project. They are grouped by domains in gray rectangles that represent the physical location of the systems such as TUMonline that is deployed at the LRZ in Garching. Responsible institutions in dark grey rectangles divide the y-axis into parts. The gray rectangles represent the target systems while the gray rounded rectangles visualize services. Target systems or services, as defined in Paragraph 5.1.2, displayed in one column with a particular system interact directly with it. An institution is responsible for all services and target systems in its row. The TUM Verwaltung for instance is responsible for the target system TUMonline that is directly connected and interacts with the Authentication Satellite.

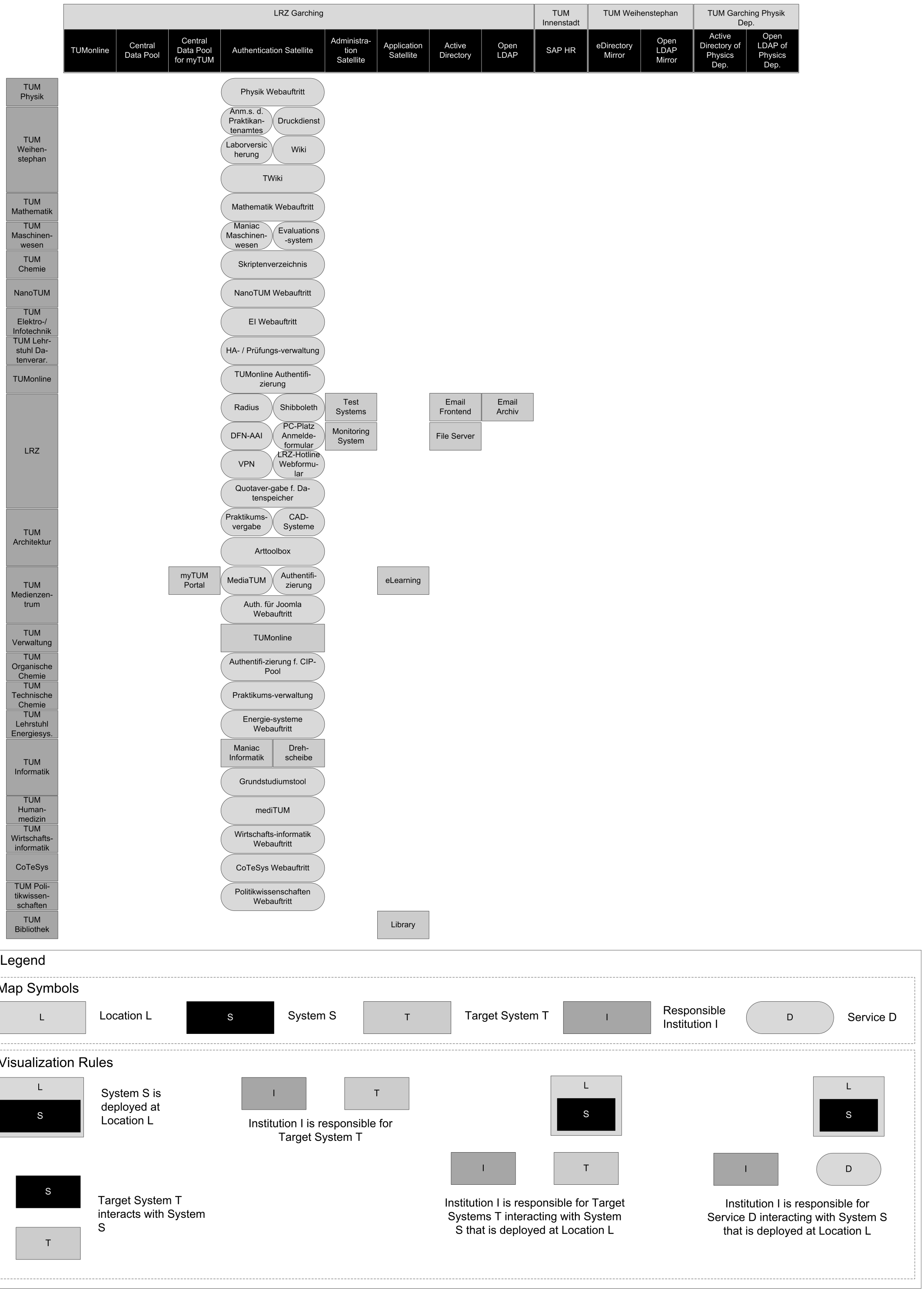

Figure 5.11: IntegraTUM: Infrastructure – Cartesian map: Target systems
Last modified: 27th of August 2009

5.4.2.2 Student lifecycle

The approach on the software map displaying the student lifecycle mainly focuses on the second part of the concern C-98: *What is the impact of the shut-down of an infrastructure element? What other elements of the application landscape are affected?*[Bu08]. A cartesian map was considered for the best representation.

The cartesian map in Figure 5.12 has the student lifecycle as x-axis. The y-axis represents the systems of the IntegraTUM project. The map visualizes the horizontal and vertical integration of target systems. For a horizontal integration two or more target systems are consolidated into one that is used in more than one process. Thus, media disruptions and technical interfaces can be avoided. Vertical integration aims at integrating data and services from many systems into one target system.

The information shown in this cartesian map is important to define *frozen zones* in which no updates or other changes are allowed for the specific systems and target systems. By this, periods of time can be determined to realize changes of the infrastructure with minimal impact to availability.

The rectangles dividing the y-axis show the systems, the rectangles in the midle show the target systems of the IntegraTUM project. The arrows on the x-axis define the processes a student undergoes throughout his studies. Every process except the last one has a successor process. This is visualized by two processes represented next to each other. A student does not have to undergo all of these processes. Obviously not all students are able to pass all exams so they skip the process of final degree (*Studienabschluss*). A target system displayed under one particular process is used in the respective process. Systems and target systems represented in one line are connected and interact with each other.

The target system TUMonline, for example, is mainly used to support the processes Matriculation (*Immatrikulation*), attending courses (*Veranstaltungen besuchen*), taking exams (*Prüfungsleistung erbringen*), and final degree (*Studienabschluss*). In this period of time there is a high interaction with the systems SAP HR, TUMonline, Central Data Pool, and Authentication Satellite.

Due to a lack of information about the services this process support map will be completed in a further project with the LRZ by gathering more information about the IntegraTUM's system landscape and adding the worked out services, systems, and target systems to the student lifecycle map.

5.5 Information model

The software maps shown previously do not have separated information bases. All information is integrated in the visualization itself, thus they have to be adjusted with every change performed to the infrastructure. This can be complex and time consuming. Therefore, it is recommended to separate the information from the visualization. This is achieved with the information model. The information model is an abstract schema (*Meta model*) of all displayed components and objects of the previous software maps. It visualizes their classes, attributes, and relations.

The demand was made to ease the handling and reduce the effort after a change of IntegraTUM's infrastructure to a minimum. Thus, the aim was to design an information model that can be used for all software maps of the previous sections. Therefore, the model shown in Figure 5.13 is based on several I-Patterns of the EAM Pattern Catalog: *I-30*, *I-32*, *I-48* and *I-68*.[4]

Six classes are needed to model the IntegraTUM project: location, system, target system, responsibility, service, and process. Each of these classes has individual attributes. Additionally, the *Support-Relationship* and the *GeneralizedSystem* class are designed allowing to model the software maps of

[4] "An Information Model Pattern (I-Pattern) supplies underlying models (the abstract syntax) for the data visualized in one or more V-Patterns." [Bu08]

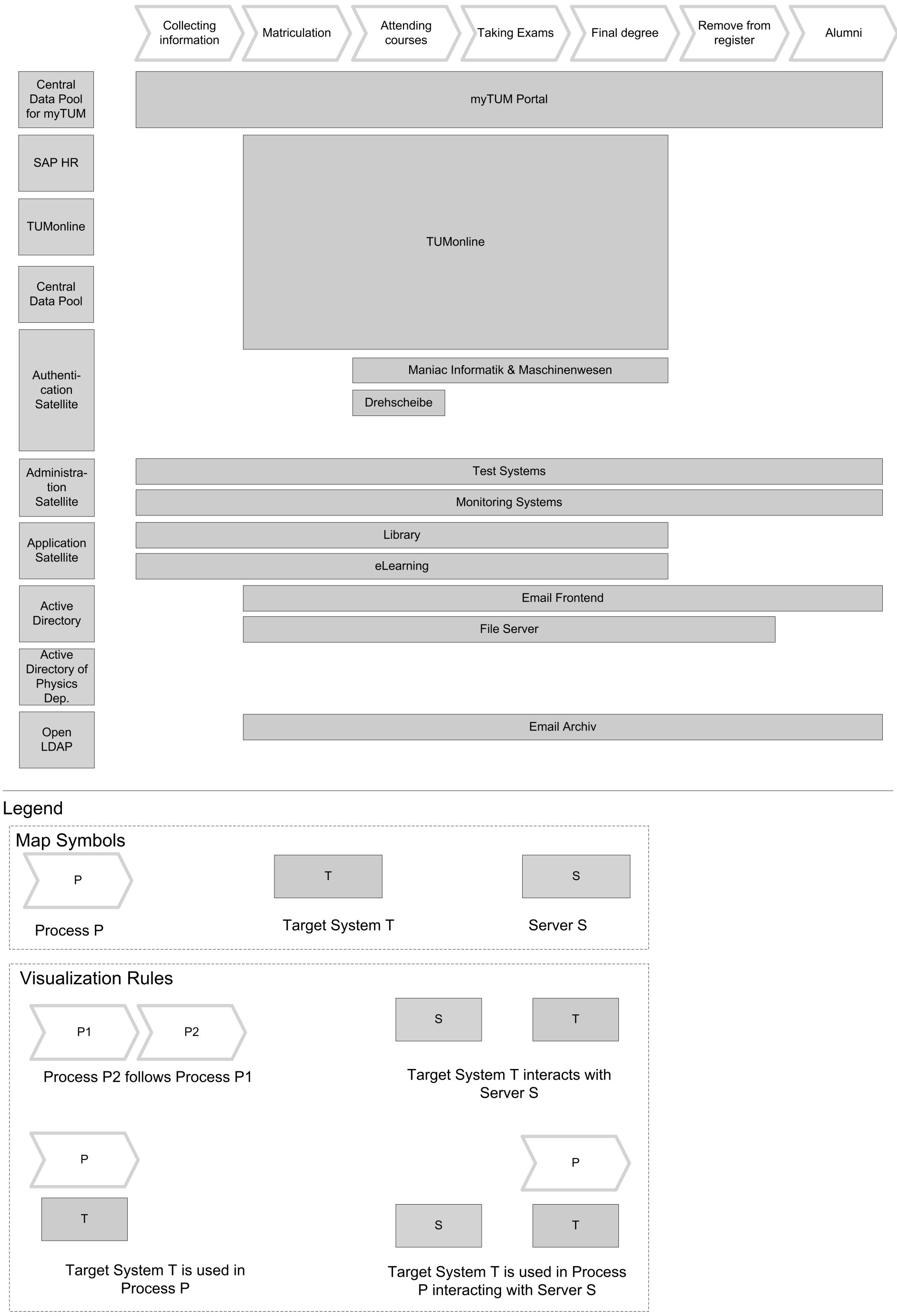

Figure 5.12: IntegraTUM. Infrastructure – Cartesian map: Student lifecycle
Last modified: 27th of August 2009

the IntegraTUM project. The GeneralizedSystem class stores all data concerning both the systems and target systems. This eases the handling and manipulation of particular data records. All classes have specific attributes. A location for instance is defined by a *location ID (LID)*, a name, and an address. The classes are connected by labeled lines describing the type of relationship. All six classes

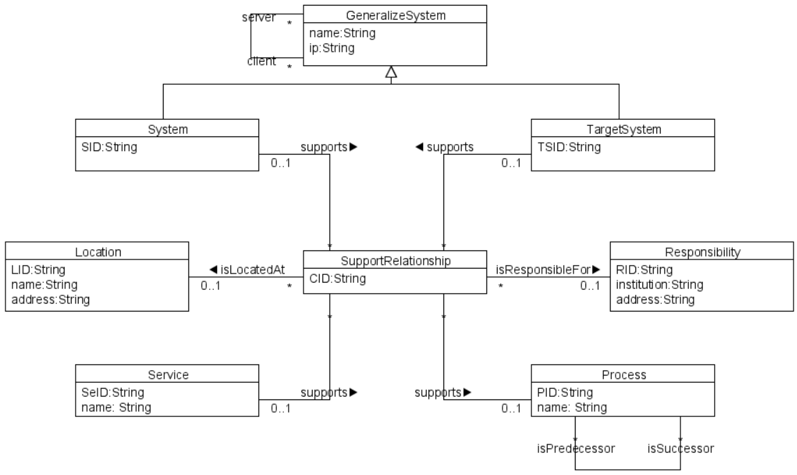

Figure 5.13: IntegraTUM: Information ModelLast modified: 27th of August 2009

are connected with the SupportRelationship. In this class the combination of connections between all classes are defined and stored. If a system for example is deployed at a location there is a data record saved in the SupportRelationship containing the *system ID (SID)* and the location ID (LID). The lines also have multiplicities that define the allowed number of corresponding classes. In this case all classes can be stored in any number of correlations but a relationship does not have to have all classes in it. Systems can also interact with other systems in a many-to-many relationship and processes can be successor of any number of other process.

With the help of the information model and the information of the infrastructure, stored in an Excel XML file, it is possible to automatically generate software maps for instance with the SoCaTool[5] from the sebis chair of the TUM.

5.6 SoCaTool – generating software maps

Maps for system landscapes can be created manually by a drawing tool (e.g. Microsoft Visio) or by some modeling tool, which uses the information stored according to an information model to automatically generate a visualization of the system landscape (e.g. sebis SoCaTool). The SoCaTool has been developed at the sebis chair and is OpenSource software. Initially, it was a special tailored tool for software cartography, i.e. to generate software maps. In the meantime, the functionality has been extended to cover visualizations of data modeled according to an object-oriented information model [Is09]. In this context, the SoCaTool in version 5.1 has been evaluated and introduced to the client to investigate its potential use.

The SoCaTool shown in Figure 5.14 separates the information from the visualization model and conducts the one-sided transformation from one model to the other automatically (see [Bu07] for more details). This implies that only with a little interaction by the user the SoCaTool generates the specified maps out of the data the user stores in a file. The data (entities and associations) is stored in a Microsoft Excel XML file, which has to be created manually by the user. In the following step, the user also defines the information model by creating an Ecore-XMI file. The generated maps can

[5]http://wwwmatthes.in.tum.de/wikis/sebis/sycatool

be exported to various file formats. The availability of three different views (helicopter, legend, and layers) makes it easy to navigate through the data available.

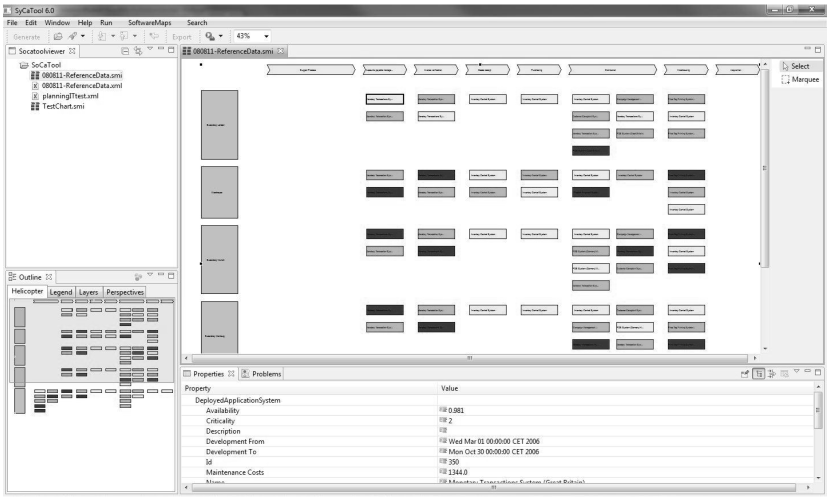

Figure 5.14: Screenshot of the SoCaTool

The creation of the information basis needed by the SoCaTool was obstructed due to a lack of time. Therefore, the process of building the needed data for using the SoCaTool was not further investigated. The client has recognized the possible added value generated by using the SoCaTool but decided to realize this in a further project.

5.7 Conclusion and future work

The project *Modeling and Documentation of IntegraTUM's System Landscape* can be seen as success because the goals of the project have been achieved, especially as client's expectations were fulfilled. At the end both, the client and the project team, benefit from the implementation of the project. The project team gained an insight in the work and the architecture used by the client. The main tasks the contractor had to solve were defining the problem, proposing an approach to tackle the problem, documenting the system landscape, and creating some software maps out of a self-created information model. The client received a sound documentation of the system landscape used in IntegraTUM. The documentation consists of models of the system landscape as well as the generated software maps including the information model. The information model is the basis for the software map. Furthermore, the client obtained an impression how to improve his work further by means of the SoCaTool and the process support maps for the student life cycle.

The biggest obstacle throughout the project was the collation of information. There was only insufficient documentation about the system landscape, the used target systems, the location where the target systems are based, and the responsibilities for the systems and target systems available. This shortcoming was reduced due to the fact that the client supported the project with great dedication and effort. The client was always available and delivered the necessary data in-time.

The generated software maps will support the client at his service desk for instruction and troubleshooting, service level agreements, as well as further development and migration. The client is looking forward to introduce the SoCaTool in the context of a next project with the sebis chair. Another further project will follow up the idea of student lifecycle maps as they were only schematically introduced within this project.

Bibliography

[Bu07] Buckl, S. et al.:. *Generating Visualizations of Enterprise Architectures using Model Transformation (extended version). Enterprise Modelling and Information Systems Architectures – An International Journal.* 2(2):3 – 13. 2007.

[Bu08] Buckl, S. et al.:. *Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008).* Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2008.

[Is09] Chair for Informatics 19 (sebis), T. U. M.:. *sebis: SyCaTool.* `http://wwwmatthes.in.tum.de/wikis/sebis/sycatool` (cited 2009-08-16). 2009.

[Wi07] Wittenburg, A.:. *Softwarekartographie: Modelle und Methoden zur systematischen Visualisierung von Anwendungslandschaften.* PhD thesis. Fakultät für Informatik, Technische Universität München, Germany. 2007.

CHAPTER 6

Development of candidate patterns for the activities "Envision EA" and "Plan EA" from the Enterprise Architecture Management Process

Mariana Mykhashchuk (mariana.mykhashchuk@mytum.de),
Ivailo Tsvetkov (ivailo.tsvetkov@mytum.de),
Nicolai Zaidman (nicolai.zaidman@mytum.de)

Abstract:
The design of the Enterprise Architecture (EA) management process in an enterprise is difficult due to both the complexity of the process and its environment.There are a lot of different approaches in the literature trying to reduce this complexity through establishing a structuring framework for EA management functions. In this paper, the relevant framework as proposed by the Chair of Software Engineering for Business Information Systems (sebis) at the Technical University of Munich is used as a basis. This framework introduces distinct activities that are relevant for a continuous and holistic EA management function. For two of these activities, namely "Envision EA" and "Plan EA", this paper provides additional information on how to implement. In doing so, each activity is complemented with individual management concerns of relevance, which are further mapped to supporting viewpoints as well as corresponding information models. In this vein, the suggested solution follows a pattern-based approach towards EA management, but also relies on best-practices as presented in The Open Group Architecture Framework (TOGAF).

6.1 Motivation

The survival of the enterprise in an environment with ever growing complexity and increasing change dynamics requires the ability to adapt the IT to new market situations. A way to achieve the required alignment between business and IT is implementation of Enterprise Architecture Management (EAM) [Bu09a]. For the purposes of this work we are using the following definition of EAM found in [Er06]:

> EA Management is a continuous and iterative process controlling and improving the existing and planned information technology (IT) support for an organization. The process not only considers the IT of the enterprise, but also business processes, business

goals, strategies, etc. are considered in order to build a holistic and integrated view on the enterprise. The goal is a common vision regarding the status quo of business and IT as well as of opportunities and problems arising from these fields, used as a basis for a continually aligned steering of IT and business.

Due to the importance of EAM for the enterprise various frameworks have been developed to support the implementation of EAM. In their survey [Bu09b] describe that the existing frameworks are extensively used either as sources of inspiration or as basis for implementing an EAM function in an enterprise. In the latter case, nevertheless, the frameworks are mostly adapted to the specific requirements of the using enterprise. According to our understanding, a key finding of this survey is that the large part of practitioners believe the existing frameworks are too theoretical and impossible to be implemented in a business environment. This leads to the assumption that the industry needs a detailed methodology of how to implement an EAM framework.

Responding to the needs of the industry the Chair of Informatics 19 at the Technical University of Munich proposes a framework for structuring a EAM function. In this framework, the following activities constituting an EA management function are introduced:

- Envision EA
- Plan EA
- Document EA
- Analyze EA
- Enforce EA

It has to be noted, that no specific ordering is defined over the different activities, as they can be combined in various ways, where even a concurrent execution of different activities is possible.

In this work, we propose methods for performing the activities "Envision EA" and "Plan EA" as found in existing EAM literature. In presenting these methods, we follow the notion of EAM patterns [Er08]. Therefore, three different constituents for implementing the activities are described, namely Methodology patterns (M-Pattern), Viewpoint patterns (V-Pattern) and Information model patterns (I-Pattern). In [Bu09b] we have found evidence that the most popular EAM framework in the industry is TOGAF. That's why we have used that framework as a reference point in respect to EAM for the most part of this work.

The remainder of the paper is structured as follows. Section 6.2 is concerned with the preliminary activity of stakeholder identification, which should precede the establishment of an EA management process. Section 6.3 takes a look at Envision EA and concerns from the domains of Business Architecture, Information Systems Architecture and Technology Architecture. Section 6.4 introduces the topic Plan EA and the related concerns and methodology. Section 6.5 summarizes the findings of this paper and provides direction for future work.

6.2 Stakeholder identification

In order to establish the target business architecture, the enterprise strategy has to be defined as well as the business goals, requirements and constraints. To be successful in this endeavor, it is necessary to know the relevant EA management stakeholders as well as their concerns. Below, we present a method, viewpoints and a metamodel that can be used to identify and manage stakeholders. These three constituents are themselves presented as pattern, although they are no EA management patterns but merely patterns for governing the EA management process.

In this work we understand the term *stakeholder* in accordance with [Pr96] as follows:

> Stakeholders are individuals and organizations that are actively involved in the project, or whose interests may be positively or negatively affected as a result of project execution or project completion.

Example

An enterprise wide process as EA management has many different stakeholders. Each of them has a specific set of requirements and expectations, which should be satisfied by the EA management process.

Context

The business goals, drivers and constraints as well as the corporate strategy and mission have been already derived in previous steps from the envisioning the target business architecture. In order to get a more in-depth understanding of the EA management process to set up, it is essential to identify key stakeholders, their concerns and requirements.

Problem

The whole project of setting up an EA management process is at risk, if the key stakeholders, their concerns and requirements are not identified correctly, because the resulting EA would be irrelevant for them. If the EA is irrelevant to the key stakeholders, it has no reason to exist. Put in a question, the problem under consideration is: how to develop the future EA, so that it reflects all concerns and requirements of key stakeholders?

Solution

The method *Stakeholder Management* (see Fig. 6.1) addresses the problems described above by identifying and documenting the key stakeholders of the EA management process.

The whole process is decomposed of three steps, namely *Identifying stakeholders*, *Identifying stakeholder's impact* and *Identifying stakeholder's concerns and viewpoints*. The exact nature of these steps is detailed in the following.

Identifying stakeholders
The EA framework TOGAF [Th09] proposes a method for identifying stakeholders, which consists of the following two steps:

1. Get the corporate organigram
2. Start from the top of the organigram asking and brainstorming on the questions:

   Who gains and who loses from the EA?

   Who controls change management of processes?

   Who designs new systems?

   Who will make the decisions?

   Who procures IT systems and who decides what to buy?

   Who has specialist skills the project needs?

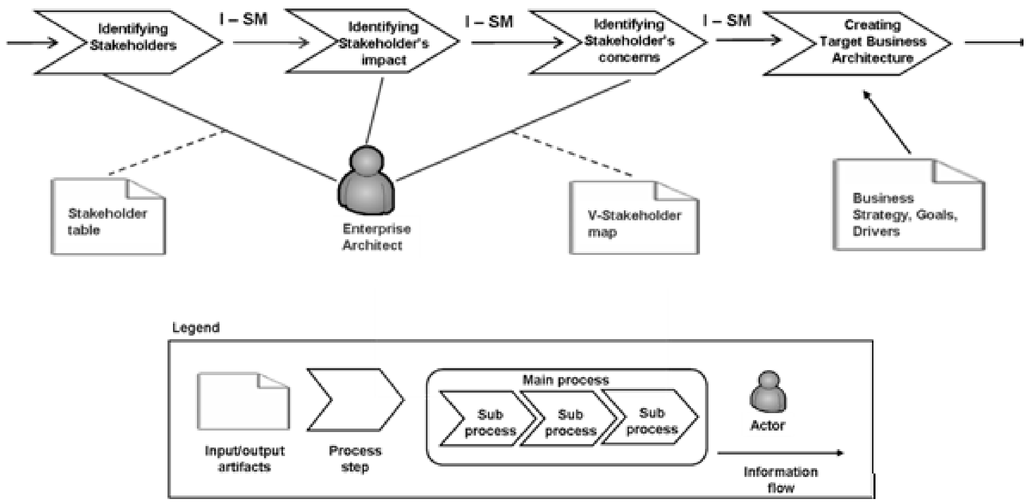

Figure 6.1: Stakeholder Management Process

Who has influence?

Classify the stakeholders according to Figure 6.1.

The answers to the above questions help to identify potential stakeholders of the EA management process. Based on this superset of candidates, a classification of the stakeholders in respect to their level of interest in the EA management project and to their decisive power in the organization is used to refine the stakeholder list. Table 6.1 shows a *stakeholder power grid*, which can be used to classify stakeholder candidates in the aforementioned way.

		level of interest	
		low	*high*
decisive power	*low*	minimal effort	keep informed
	high	keep satisfied	key players

Table 6.1: Stakeholder Power Grid. Source [Th09]

Using the power grid, for example *Key Players* can be identified, who are stakeholders that have a big interest in realizing the EA management project and they also have a high decisive power like CxOs.

Identifying stakeholder's impact
In this second step the *level of commitment* of each stakeholder is documented. According to [Th09], a tabular checklist as shown in Table 6.2 can be used to analyze each stakeholder's impact.

Identifying stakeholder's concerns and viewpoints
In the third and last step, the analysis should be further extended in direction of the relevant viewpoints and the major concerns (cf. Table 6.3)

Consequences

Better acceptance and less stakeholder resistance - when the key stakeholders can see that their concerns and requirements are taken into account, they are more willing to accept the EA management

Stakeholder	Ability to disrupt change	Current under-standing	Required under-standing	Current commit-ment	Required commit-ment	Required support
CIO	High	Middle	High	Low	Middle	High
CFO	Middle	Low	Middle	Low	Middle	High
...	...	...	...	...	...	...

Table 6.2: Stakeholder’s impact on EAM project. Source [Th09]

Stakeholder	Involvement	Class	Relevant Viewpoints	Concerns
CIO	The stakeholder is interested in the high-level drivers, goals, and objectives of the organization.	Keep satisfied	Business foot-print, Organiza-tion Chart	Which applica-tion systems are used by which organizational units?
...	...	...	...	...

Table 6.3: Analyzing stakeholders concerns and viewpoints. Source [Th09]

project. Also their resistance should be less, because they have direct benefit from the EA management project.

Stronger support from key stakeholders - when the key stakeholders realize that supporting the EA management project helps them to achieve their own objectives and goals, they are going to support the EA management project even more willingly.

See Also

In order to gain a quick overview of the stakeholders and their position and role in the EAM process a visualization following the structure of the *Stakeholder Map* (see Figure 6.2) should be considered. Detailed descriptions of Tables 6.2 and 6.3 as well as examples can be found in [Th09].

6.2.0.3 Visualization: Stakeholder Map

This visualization allows to represent the information gathered during the Stakeholder Management process. Figure 6.2 show an example of a portfolio-matrix like diagram that classifies the key stakeholders according their decisive power and level of interest. The required support form the stakeholders for the EA management project is represented in three different sized circles (small size = low support required; middle size = middle support required; big size = high support required). The stakeholder’s ability to disrupt change is represented in a color scheme (green = low ability to disrupt change; yellow = middle ability to disrupt change; red = high ability to disrupt change).

Underlying the visualization, the information according to the following metamodel must be gathered. This information is further used to support the aforementioned method for stakeholder management. Figure 6.3 shows the corresponding metamodel.

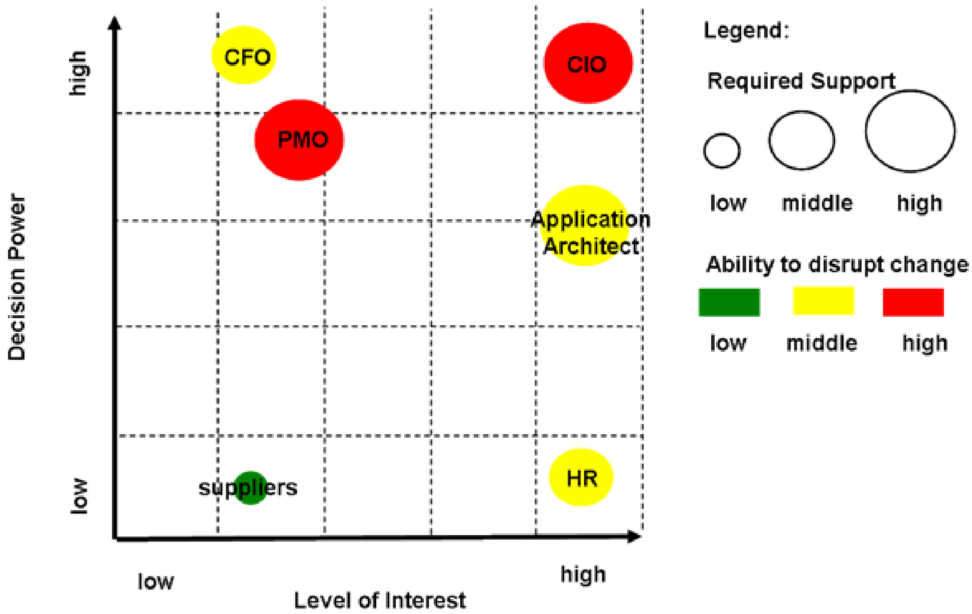

Figure 6.2: Stakeholder Map

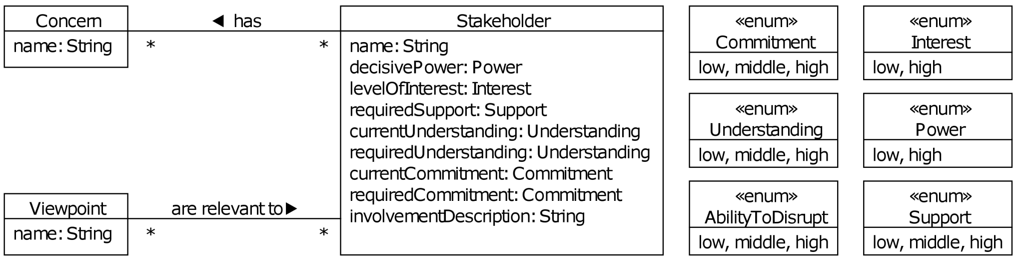

Figure 6.3: Stakeholder Information metamodel

6.3 Envision EA

Envision EA is an iterative process that is often used to start an EA management process. According to [Th09], it has different objectives like:

- ensuring that the project that starts the implementation of the EA management process gets the required recognition from top management,
- defining the scope of the EA, and
- validating the business strategy, goals, drivers and constraints.

Figure 6.4 shows a conceptual model for the Envision EA activity, which according to [Th09] gets input of three categories:

- Organizational model for EA - defines the organization, roles, responsibilities and constraints that are needed to support the successful implementation of the architecture framework.

- Architecture framework - the architecture framework that is going to be used, needs to be tailored on the specific needs of the enterprise. This includes the definition of common vocabulary and synchronization with existing project and process management frameworks.
- Architecture repository - acts as a storage for all enterprise architecture projects

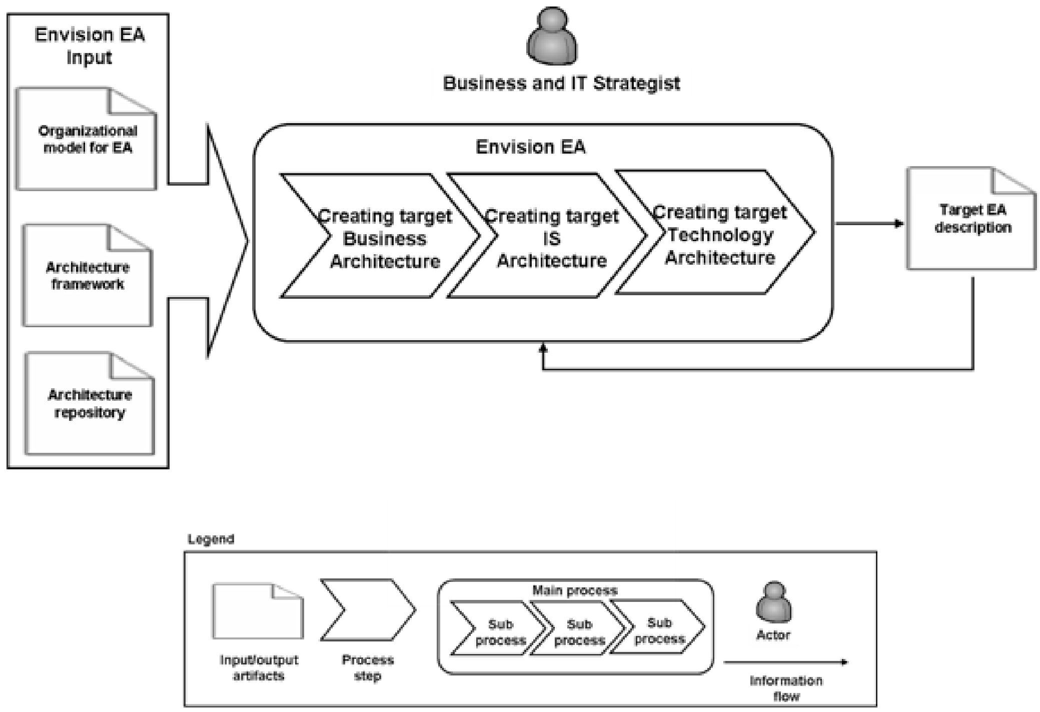

Figure 6.4: Envision EA

The Envision EA activity is further divided into three sub-processes. The first of them delivers the description of the target Business Architecture. It serves as input for the sub-process that creates the description of target information systems architecture. The output of this process itself feeds the last sub-process that creates target technology architecture. In this work, we focused on the sub-processes of creating the target information systems architecture and creating the target technology architecture, which are detailed below.

6.3.1 Envision EA – target information systems architecture

In this section, we discuss a method for developing the target information systems architecture of an enterprise from a previously defined business architecture. The importance of the target IS architecture not only can been derived from TOGAF, but also from a lot of other authors [Ke07, En08, De06]. Resorting to a definition of the target IS architecture given by Engels in [En08], we define this architecture as the collection of application systems, which a company uses for organization and development in its business.

6.3.1.1 M-Pattern candidate – Creating Target Information Systems Architecture

The M-Pattern candidate *Creating Target Information Systems Architecture* helps to describes the development of target information systems architecture.

Example

In order to make the new strategy of an enterprise possible, some old application systems have to be replaced by new ones using the modern JAVA-based technology. The application landscape of the enterprise, which became very heterogeneous and complex in last few years, has to be reworked in this context. To guide this rework, a new target architecture needs to be defined in a methodical approach.

Context

This M-Pattern can be used by an enterprise, which is interested in developing its business strategy oriented information systems architecture.

Problem

If each IS is developed in isolation, their structure and interplay will most likely not correspond to the IT strategy of the enterprise. This will result in non-compliance with the business strategy. Consequently, the business objectives cannot be achieved optimally. How do I act in order to develop the business oriented target information systems architecture?

The following forces affect the solution:

Business strategy of the enterprise: Which IT components should be used in order to realize the business strategy? Do the currently used infrastructure components contribute to the realization of the business strategy?

IT strategy of the enterprise: How should the enterprise's IT strategy be changed in order to realize the business strategy?

Solution

The M-Pattern candidate *Creating Target Information Systems Architecture* is a proposal to solve the problem described above. As shown in the Figure 6.5, the development of the information systems architecture requires the definition of the business architecture (see e.g. [Th09]). In this context, the following artifacts are important: architecture principles, organizational architecture and process architecture.

Within the process step **Creating Target Data Architecture** the cross-enterprise definitions of the information objects are defined. The enterprise architect is the person who is responsible for the execution of this step. This process step is based on the data architecture as described by TOGAF [Th09].

In context of **Creating of Domains Knowledge**, domains are identified using the knowledge about (core) business process. A domain provides a basis for grouping the constituents of the application landscape following functional aspects and is used to decrease the complexity [En08]. In this step the participation of a person with domain knowledge, e.g. the business architect, is essential.

In the process step **Creating Target Application Systems Architecture** the application systems are assigned to the application domains. This process step based on the application architecture how described by TOGAF [Th09].

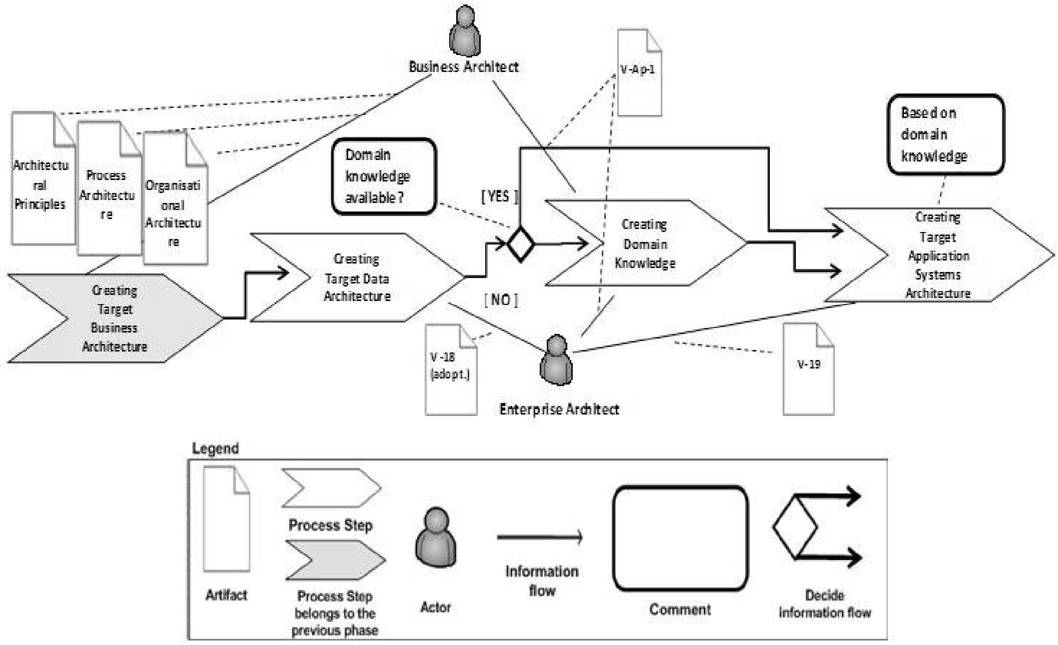

Figure 6.5: M-Pattern candidate "*Creating Target Information Systems Architecture*" with the previous process step

Consequences

Methodical approach: The development of the target information systems architecture follows a methodical process oriented approach.

Common understanding: Common understanding of the information objects as basis for a successful communication.

Orientation on existing IT strategy: The development of individual architecture components is oriented towards existing IT strategy.

See also

The V-Pattern candidate *Application Domain Map* (see Section 6.3.1.2) is useful in order to demonstrate how domains can be used in context of information systems architecture. In addition the following V-Patterns from [Ch09] can be used: *Business Services provided by Application systems and used by Processes* (V-19), *Service-based Business Process Support Map* (V-18) extended by information on data objects for each business service.

6.3.1.2 V-Pattern candidate – Application Domains Map

This V-Pattern provides a way to visualize the existing domains knowledge as well as a mapping between domains knowledge and application systems.

Problem

How to group individual elements of the application landscape?

Solution

The V-Pattern candidate *Application Domains Map*, shown in Figure 6.6, addresses the problem described above. Some application systems used to provide the same service can be grouped within an application domain, as e.g. ”Purchase”. The application domains supporting coherent services, in turn, can be combined in domain layers, as e.g. the domains ”Purchase” and ”Planning”, which both support the core business of the enterprise.

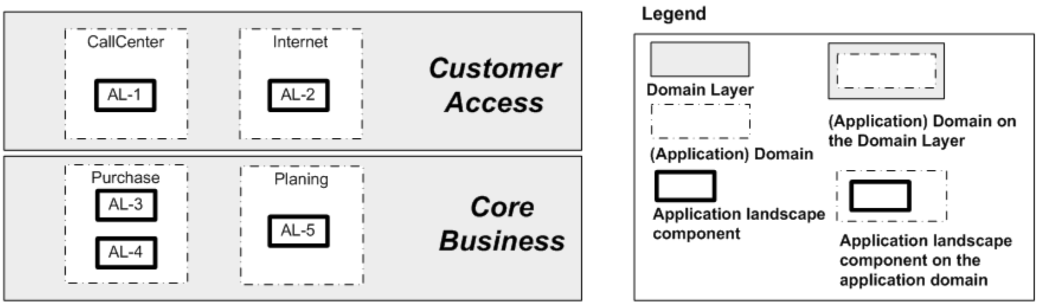

Figure 6.6: V-Pattern candidate "*Application Domains Map*". Source: Adapted from [En08]

See also

The visualized information is based on the I-Pattern candidate *Envisioning an Information Systems Architecture.*

6.3.2 I-Pattern candidate – envisioning an information systems architecture

This I-Pattern candidate provides a way to store information about information systems architecture.

Problem

Which information about the elements of the enterprise architecture and their relationships is relevant with regard to the target information systems architecture?

Solution

The Figure 6.7 shows the I-Pattern candidate Planning Information Systems Architecture addressing the problem described above.

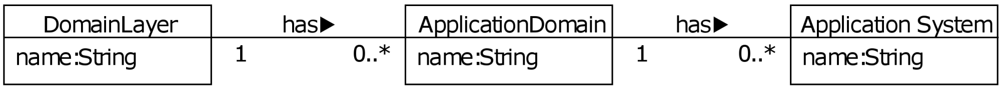

Figure 6.7: I-Pattern candidate "*Planning Information Systems Architecture*"

6.3.3 Envision EA – target technology architecture

After the target information systems architecture had been defined, the technology architecture as the collection of technical components used to make operation of the application systems possible can be designed.

The M-Pattern "*One Infrastructure*" (M-45) as well as the V-Pattern "*Infrastructure Usage Diagram*" (V-56) and "*Architectural Solution and Technology Mapping*" (V-23) can be used in this context. The I-Pattern depicted in figure 6.4 covers the technological components and can also be used in this case as underlying information model.

6.4 Plan EA

The EA management pattern candidates included in this section of the article describe another activity of the Enterprise Architecture Management process, namely Plan EA. According to our solution, the activity is addressed by one M-Pattern, two V-Pattern and one I-Pattern candidates. Figure 6.8 shows a pattern map visualizing the relationships between the pattern candidates. In the following, pattern candidates are referenced by their names.

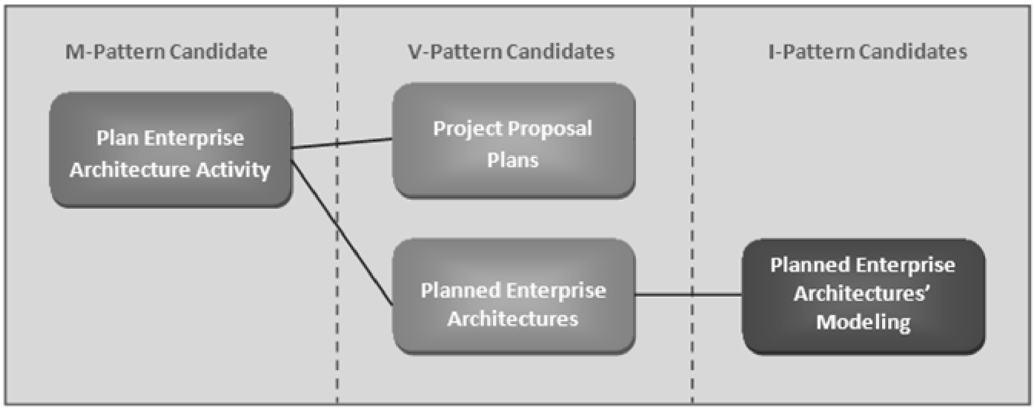

Figure 6.8: Pattern Map for Plan EA activity

6.4.1 M-Pattern candidate – plan enterprise architecture activity

M-Pattern candidate *Plan Enterprise Architecture Activity* defines and manages the Plan EA activity. The main focus lies on development of planned EAs, which are to contribute to the transition from current EA to target one. In this respect, the planned EAs support a smooth architectural transition.

Example

The marketing department of a large mobile network operator has discovered considerable increase of demand on 3G services since the beginning of the year. Consequently, a *3G Network Implementation Project* is proposed in enterprise-wide demand management. Thereby, a decision on how to coordinate the project proposal with the enterprise strategy has to be taken.

Context

Most large enterprises have to deal with continuously changing external factors, such as market conditions, new technological developments, etc. Therewith emergent operational goals should be coordinated with strategic ones.

Problem

You have to decide, whether a project proposal corresponds to your enterprise strategy. How do you align the project proposals with the enterprise strategy and deliver the feasible group of projects, which takes into consideration the strategy of the company as well as its operational business or technical needs?

The following *forces* influence the solution:

- **Strategic versus operational needs**: the rapid change of the market situation and business environment can require the need of quick reaction of "operational" management leaving no time for the strategy adaptation. How can the enterprise strategy be developed along to the relevant operational projects?
- **Comparability**: How can a single project be compared with the architecture, the enterprise is aiming at?
- **Feasibility**: How can the feasible combination of strategic and operative projects be built?

Solution

M-Pattern candidate *Plan Enterprise Architecture Activity* addresses the problems described above. But before introducing the methodology, input and output factors of the Plan EA activity are defined:

- Current Architecture (mandatory),
- Strategic Project Proposals (optional), and
- Operational Project Proposals (optional).

Strategic project proposals are generated by the Analyze EA activity in order to transform from the current (or planned) to the target architecture. Operational project proposals are delivered by Demand Management in order to cover the operational needs. As output of the Plan EA activity, the planned EA proposal descriptions are generated, which are feasible and correspond to the target architecture.

The whole Plan EA activity is divided into two steps, see Figure 6.9. The first process step *Project Proposal Plans' Construction* generates project proposal plans on the basis of strategic and operational project proposals. In doing so the dependencies between the project proposals are taken into account, and thus, the delivered by the step project proposal plans are feasible. Project proposal plans as well as current architecture are considered in the next process step *Planned EA's Modeling*, which derives planned enterprise architectures.

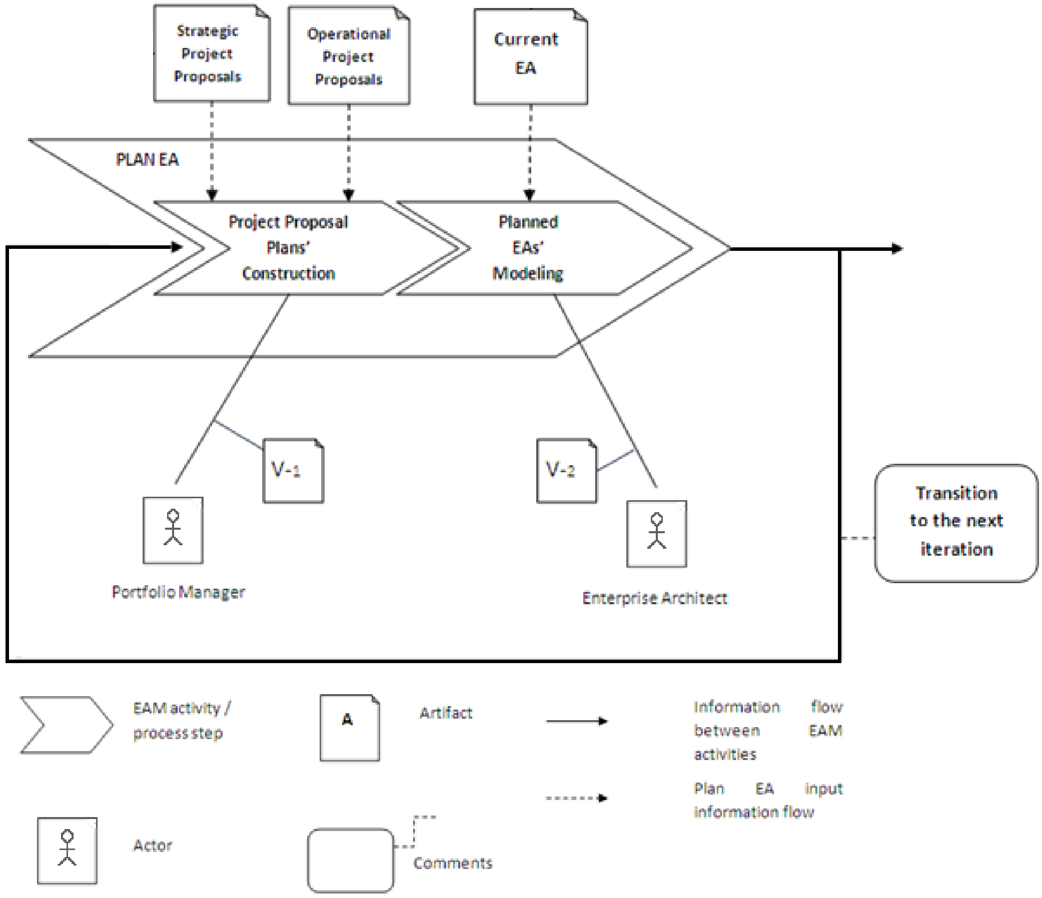

Figure 6.9: M-Pattern candidate "*Plan Enterprise Architecture Activity*"

The architectures generated during the first iteration may also be feasible, still they have not been checked for correspondence with the target architecture, which however, should be verified due to incoming operational project proposals. So planned enterprise architectures are to be forwarded to the Analyze EA activity, which performs the gap-analysis between each of them and the target architecture. For each planned EA, a number of strategic project proposals is delivered which could fill the gaps. Output from the Analyze EA activity is forwarded again to the Plan EA activity, which restarts with a new iteration.

Consequences

Strategic versus operational needs: in order to consider both strategic as well as operational project proposals these types of project proposals are taken into account while project proposal plans' construction.

Comparability: as the task of direct comparison between a single project proposal and an architecture is quite difficult and not certainly precise, the planned enterprise architectures are modeled, which enables the comparison between two architectures instead of comparing a project with architecture.

Feasibility: the feasibility is understood in the sense of dependencies between the projects. Feasibility is ensured by making plans, which take the dependencies between project proposals into account.

See also

In order to support the implementation of the M-Pattern candidate Plan Enterprise Architecture Activity the following two V-Pattern candidates, *Project Proposal Plans* and *Planned Enterprise Architectures* should be considered (see Section 6.4.2).

6.4.2 V-Pattern candidates

This section introduces V-Pattern candidates: *Project Proposal Plans* and *Planned Enterprise Architectures.*

6.4.2.1 Project proposal plans

Problem

You want to get an overview about all project proposal plans, modeled on the basis of all possible feasible combinations of strategic and operational project proposals, to make modeling of planned enterprise architectures more transparent and clear, as well as to keep all the threads of project proposals in mind. Moreover, all the possible feasible project proposals threads should be taken into consideration.

How do you visualize all the project proposal plans in a transparent but still complete manner?

Solution

V-Pattern candidate *Project Proposal Plans* consists of a table, which contains all the created project proposal plans (see Table 6.4). Each plan consists of groups of project proposals, which can be implemented either concurrently or sequentially. Table columns depict time dimension in a sense of first possible iteration, in which the corresponding projects can be carried out. Groups of project proposals within a single iteration can be executed concurrently.

The number of different project proposal plans depends on the number of projects which have no prerequisites and can be implemented immediately as well as on the depth of dependencies between the project proposals.

6.4.2.2 Planned enterprise architectures

Problem

You want to get an overview about all planned architectures as well as about the current architecture and the project proposal threads, which transform the architecture from current state to corresponding planned one.

How do you visualize all the planned enterprise architectures with corresponding project proposal plans in a transparent but still complete manner?

# Plan	Current iteration	Next iteration	Iteration after next
1	-		
2	P1		
⋮	⋮		
16	P1, P2, P3, P6		
17	P3	P5	
18	P1, P2	P4	
⋮	⋮	⋮	
30	P1, P2, P3, P6	P4, P5	
31	P3, P6	P5	P7
32	P1, P3, P6	P5	P7
⋮	⋮	⋮	⋮
35	P1, P2, P3, P6	P4, P5	P7

Table 6.4: V-Pattern candidate "*Project Proposal Plan*"

Solution

V-Pattern candidate *Planned Enterprise Architecture* consists of a table, which contains the entire planned enterprise architectures as well as corresponding project proposal plans, which transform the current architecture to the planned one (see Figure 6.10). Table rows represent sequence numbers of planned architectures. In each row the current architecture as well as the project proposal plan and the corresponding planned Enterprise Architecture proposal are represented.

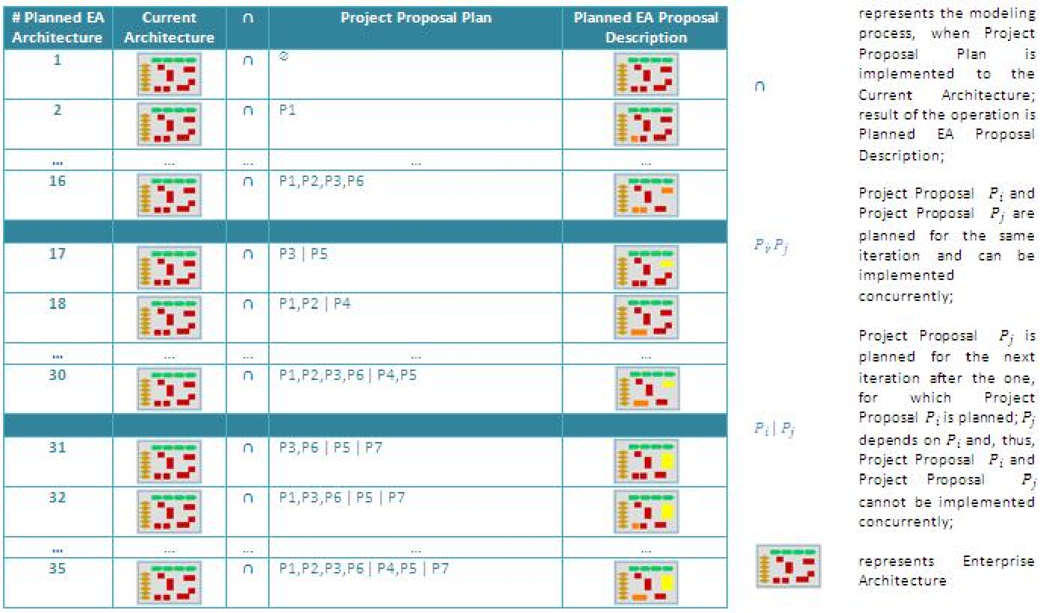

Figure 6.10: V-Pattern candidate "*Planned Enterprise Architectures*"

6.4.2.3 See also

These V-Pattern candidates may be useful while using M-Pattern candidate *Plan Enterprise Architecture Activity* (see Section 6.4.1). The visualized information is based on I-Pattern candidates *Project Proposal Plans' Construction* and *Planned Enterprise Architectures' Modeling* (see Section 6.4.3).

6.4.3 I-Pattern candidate – planned enterprise architectures' modeling

Current section introduces I-Pattern candidate *Planned Enterprise Architectures Modeling*, which supports a corresponding step of M-Pattern candidate *Plan Enterprise Architecture Activity.*

6.4.3.1 Problem

You want to know more about the utilization of project proposals within project proposal plans and project proposal's as well as project proposal plan's structure. In addition, you are interested in all artifacts used by Plan EA and dependencies between them.

How should an information model look alike, to be able to store information about all artifacts used by the Plan EA activity and dependencies between them?

6.4.3.2 Solution

The solution of the problem described above is based on I-Pattern *Planned Enterprise Architectures Modeling* (see Figure 6.11), while details on the EA structure are omitted for reasons of readability.

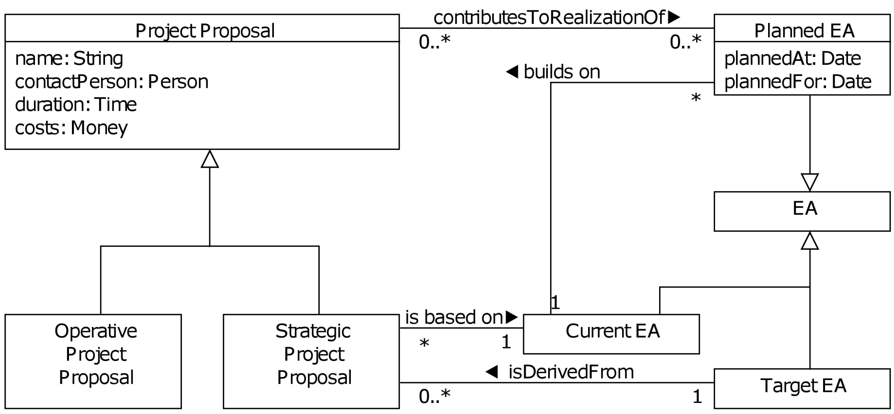

Figure 6.11: I-Pattern candidate "*Planned Enterprise Architectures Modeling*"

6.4.3.3 See also

I-Pattern candidate *Planned Enterprise Architectures Modeling* is closely related to V-Pattern candidate *Planned Enterprise Architectures* (see Section 6.4.2) and provides M-Pattern candidate *Plan Enterprise Architecture Activity* (Section 6.4.1) with the necessary information model.

6.5 Outlook and acknowledgment

This article shows a pattern-based approach to realize the activities Envision EA and Plan EA as part of EA management. For both EA management activities the concerns were determined and possible solutions in the form of M-, V- and I-Pattern candidates were suggested. Our solution is partially based on established frameworks for EA, namely The Open Group Architecture Framework and the EAM Pattern Catalog [Ch09].

Complementing the EA management patterns, we further developed a set of methods, visualizations and metamodel patterns for stakeholder identification that contributes to the field of EA management governance. Admittedly, all proposed solutions have not yet been applied in a practical setting. Nevertheless, we could discuss them with EA management expert from a consulting company, Mr Christian Winterhalder, to whom we also express our gratitude. We would also thank our advisors, Ms Sabine Buckl and Mr Christopher Schulz, who provided our research and gave helpful advices. In prospect the practical testing of the suggested solutions is planned, during which it will be checked, whether the pattern candidates, offered in our solution, find acceptance in industry and join the ranks of the EAM pattern catalog.

Bibliography

[Bu09a] Buckl, S. et al.:. *EAM Pattern for Consolidations after Mergers.* In *SE 2009 – Workshopband.* Kaiserslautern. 2009.

[Bu09b] Buckl, S. et al.:. *State of the Art in Enterprise Architecture Management 2009.* Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2009.

[Ch09] Chair for Informatics 19 (sebis), Technische Universität München:. *EAM Pattern Catalog Wiki.* http://eampc-wiki.systemcartography.info (cited 2009-08-06). 2009.

[De06] Dern, G.:. *Management von IT-Architekturen (Edition CIO).* Vieweg. Wiesbaden. 2006.

[En08] Engels, G. et al.:. *Quasar Enterprise – Anwendungslandschaften serviceorientiert gestalten.* dpunkt.verlag. Heidelberg, Germany. 2008.

[Er06] Ernst, A. M. et al.:. *Using Model Transformation for Generating Visualizations from Repository Contents – An Application to Software Cartography.* Technical report. Technische Universität München, Chair for Informatics 19 (sebis). Munich, Germany. 2006.

[Er08] Ernst, A.:. *Enterprise Architecture Management Patterns.* In *PLoP 08: Proceedings of the Pattern Languages of Programs Conference 2008.* Nashville, USA. 2008.

[Ke07] Keller, W.:. *IT-Unternehmensarchitektur.* dpunkt.verlag. Heidelberg. 2007.

[Pr96] Project Management Institute:. *A guide to the project management body of knowledge.* PMI Publishing Division. 1996.

[Th09] The Open Group:. *TOGAF "Enterprise Edition" Version 9.* `http://www.togaf.org` (cited 2009-07-10). 2009.

CHAPTER 7

Comparing the Usage of a Content Management System in the Radio and Television Archives of a Public Broadcasting Company

Alexander Bartsch (bartsch@in.tum.de),
Steffen Brugger (steffen.brugger@mytum.de),
Kolja Rödel (roedel@in.tum.de),
Sören Ruttkowski (ruttkows@in.tum.de),
Tobias Schlachtbauer (schlacht@in.tum.de),
Daniel Schmidt-Loebe (daniel@schmidt-loebe.de),
Torben Volkwein (volkwein@in.tum.de),
Alexandru Zerva (zerva@in.tum.de)

Abstract:
Todays companies are in the increasing need of changing their historically grown system landscape, to realize synergies as well as being able to offer new products and services to the market. Therefore, the first step in changing the landscape is to obtain an overview. High level views of the systems and their interconnections like visualizing business processes and the interfaces between the core systems help to achieve that goal.

In this paper, we describe an approach to analyze the core processes and interfaces in a public broadcasting company. This approach analyzes of the core content management system in two departments, in the television archive as well as in the radio archive. Even though the core system itself is the same system in both departments, different demands led to a diverse use of the system and the present need to identify possible synergies between the two historically grown systems. Many companies have similar situations; and the following approach may help others to cope with the problem of analyzing their system landscape in order to discover improvement possibilities through synergies.

7.1 Introduction

Today, information technology(IT) in general is becoming more and more important for companies all over the world. It is often the IT, which is the limiting part towards the introduction of new products or the reduction of costs. Therefore, the system landscape has to be efficiently managed, a commonly accepted means to manage the IT adn business architecture from a holistic perspective is enterprise architecture (EA) management. The first step in managing the EA is to understand it. Without suitable information about the processes, the systems, and their dependencies the enterprise architect cannot decide how to change the landscape in order to introduce new products or how to become more cost efficient without interfering with everyday business. In this paper, the usage of a content management system within the environment of a radio and television broadcaster for the general public is examined exemplarily to describe an approach to identify and analyze core parts of the system landscape in respect to synergy potential.

Before discussing the theoretical foundations as well as the project approach and development, some general information about the company itself and its environment is provided. The organizational structure and functions of the company under consideration are determined by a legal foundation, which makes the structures inherently rigid and hampers quick and flexible changes. The company is embedded in a network of other public broadcasting organizations throughout the country, which have all been working independently in the past. Only in the more recent years some cooperation has begun, mainly due to the fast transmitting channel of the internet. In the company under study about 3,000 employees and several freelancers are working together to provide input material for the radio and television channels.

This study took place in the archive business units of both the radio and the television departments. During everyday work, the main objective of the unit is to archive the broadcasted radio and television programs respectively. This way, parts of a program can be reused in future broadcastings for cost efficiency. In television, everything being broadcasted is automatically archived, while only certain parts or programs of the radio are being stored. The radio archive is also responsible for digitalizing old as well as importing new unsend content. Similar activities in the television archive are outsourced.

Within the company the two departments of radio and television broadcasting, as well as the single channels in each of the departments are working relatively independent and have chosen their own software over time, depending on individual needs and preferences. Due to this independence, the content management system, which was introduced in the television archive in 2004 and in the radio archive in 2005, has developed quite differently. Even though the same system was chosen in the radio archive to realize synergies, this goal has yet not been achieved.

This report is structured as follows: the main part of the paper starts with the theoretical foundations of EA management patterns, software cartography, and the Enterprise Architecture Management Pattern Catalog (EAM Pattern Catalog). Following, the methodology of analysis of the software landscape is described. Special attention is paid to the business processes using or operating on the core content management system under study. In addition, the views on the system landscape as used in the analysis are described, as are patterns for certain kinds of visualizations in the context of a system landscape with special regard to the processes and interfaces associated with the content management system. Above all, the findings of the analysis are presented as an example of a result of the proposed approach. Finally, an outlook is given towards further research, connected to other interesting patterns of the EAM Pattern Catalog.

7.2 Theoretical foundations

In computer sciences, a commonly accepted definition of architecture is "fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution" [IE00]. Further, EA can be defined as the coherent and holistic architecture of an enterprise, consisting not only of IT but also of the business; it is built of components with tight relations and connections to each other. An EA should be stable and useful and can be divided in layers and crosscutting functions [Wi07]. Figure 7.1 shows an integrated information model for EA.

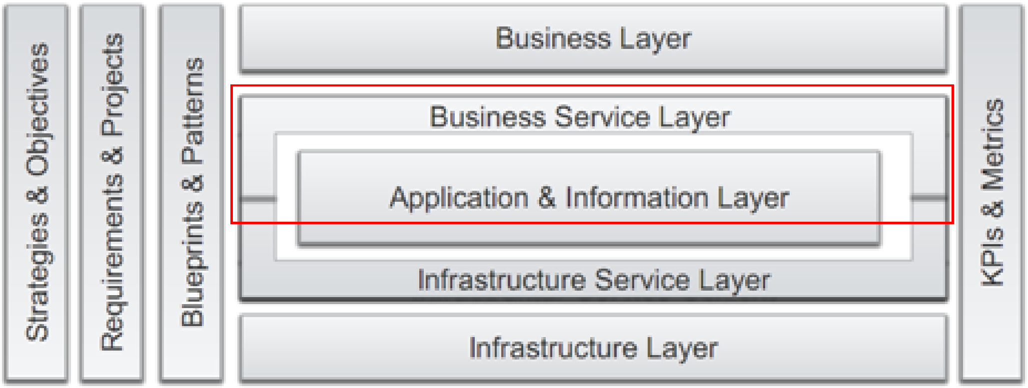

Figure 7.1: Layer and crosscutting functions as structure of an EA

In this respect "EA management (EAM) is a continuous, iterative, and self maintaining process seeking to improve the alignment of business and IT" in an enterprise [Bu09b]. The foundation of this process is the holistic perspective on the enterprise; this way, one tries to establish and ensure transparency.

As an enterprise's application landscape is a complex system consisting of hundreds or even thousands of business application systems, the documentation, evaluation, and planning of this applications landscape is a major challenge for the enterprise. For documenting the different states (as-is, planned, to-be landscape) of an EA, software maps are a useful approach [Er06]. A software map is a graphical model of an application landscape, which provides a visualization in accordance with the interest of the stakeholders. Important aspects for modeling an enterprise's application landscape are the abstraction from IT details as well as emphasis on business aspects and relations to other important constituents of the enterprise.

One way to establish EAM is by using EAM patterns. A pattern is a general, reusable solution to a common problem in a given context. One definition of a pattern, which defines patterns in the context of architecture of buildings but can be applied for EA as well, is the following of Alexander et al.: "Each pattern describes a problem, which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice" [AIS77]. A key part of patterns is that they are rooted in practice. Thus, patterns do not provide a purely theoretical solution for a certain concern, but one that could be observed by looking at what people do, observing things that work, and then looking for the core of the solution [Fo02].

7.2.1 Software cartography

EA landscapes are complex and intangible structures. Furthermore, the backgrounds and objectives of the people creating, utilizing, financing, and managing the landscape are considerably different. In order to facilitate the communication between these stakeholders, intuitive and precise visual maps are very helpful [Bu09b]. Software cartography provides models and methods for describing, evaluating, and designing application landscapes. The visual representations (models) of landscapes in software cartography so called software maps.

When using a model, the general concept (of a model) has to be taken into account. Key characteristics of a model according to Stachowiak [St73] are:

- Models are always models of something, namely portraits or representations of natural or artificial originals, which can be models themselves.

- Models commonly do not capture all attributes of their corresponding original, but only those, which seem to be relevant for the model creator and/or model user.
- Models are not clearly mapped to their originals; they are a replacement of the original.

A problem arising from the usage of simple drawn diagrams is that the visualizations inherently have no well-defined semantics that determines what the individual graphical symbols in the visualization mean. This leads to visualizations, which can only be interpreted correctly and understood by the author of the drawing [Er06]. Thus, the focus of software cartography is on modeling and not on drawing or painting. Therefore, the use of modeling languages with defined notations, syntax, and semantics is not only important, but also necessary.

A good software map possesses a title field and a legend in addition to the map field. The title field contains title, author, creation date, and contact information (person, role or group). As software maps visualize different states at specific points in time, every software map has time references. The legend explains the map symbols, the symbol variables and the visualization rules. As an example, common map symbols, like rectangles, are used differently in different models. A legend is necessary to provide clarity about the used semantics [Wi07]. See Wittenburg[Wi07] for a list of software map types.

7.2.2 EAM patterns

As with many other disciplines and fields of study, recurring problems in EAM can be addressed using a pattern-based approach. There is a great number of definitions for patterns (cf. [AIS77, Bu96, Ga94]), but they all point out similar key characteristics.

The most noticeable thing about a pattern is that it always is composed of three specific parts. First, the context is described. This helps the user identify under what circumstances the pattern in question was established. Second, the problem is laid out. Related to the described context, the problem is usually a recurring or often appearing one. By combining the information about the particular problem in the specific context, the potential user of the pattern can decide, if the pattern can be applied in his particular case. Third, a solution is proposed. The pattern offers a well-proven approach to solve the addressed problem. Depending on the pattern and the problem at hand, the solution can be a series of simple and straightforward steps, but it can also grow to be a complex generic scheme of components, responsibilities, and interactions, which needs to be adapted and applied to the encountered problem.

To sum it up, a pattern is a three-part representation of a general, reusable solution to a commonly occurring problem in a given context. In addition to this general description, an EAM pattern also includes a number of specific details, like conflicting forces and stakeholders, consequences of use, or other social, technical or economic factors. In addition, EAM patterns are usually discovered as "working solutions", and are thus strongly connected to practice.

7.2.3 EAM pattern catalog

Because EAM takes the holistic perspective of the enterprise into account, it often involves the analysis of complex situations, where questions about concerns, stakeholders, viewpoints, information availability, and necessary activities arise. In such a case, it is on the one hand very easy to lose sight of the actual problems, and on the other hand it is very difficult to be sure that every aspect of the general situation is being addressed correctly. A structured approach is needed in order to be able to stay focused on the goals and reduce the overall complexity of the situation.

The EAM Pattern Catalog [Bu08] is a collection of a number of patterns, which provide additional detail and guidance to existing EA frameworks, thus enabling the step-by-step establishment of EAM. The Pattern Catalog focuses on identifying the connections between individual concerns of the higher management, management methodologies, supporting viewpoints, and information models. A model of this approach can be seen in Figure 7.2.

While individual concerns are identified and represented as such, methodologies, viewpoints, and information models are EAM patterns of different. They describe problems and solutions related to the concern or pattern that precedes them or interacts with them. The structure is hierarchical and displays the elements in a fashion that encourages a top-to-bottom approach. Once a specific concern is identified, the user of the EAM Pattern Catalog can follow the connections to one or more Methodology Patterns (M-Patterns), which represent patterns for problems commonly associated with the identified concern. These M-Patterns lead in turn to specific V-Patterns, which contain information about specific visualizations, which may be helpful in the context. Finally, the visualizations are based on information models, which are presented in the I-Patterns at the very bottom of the hierarchy.

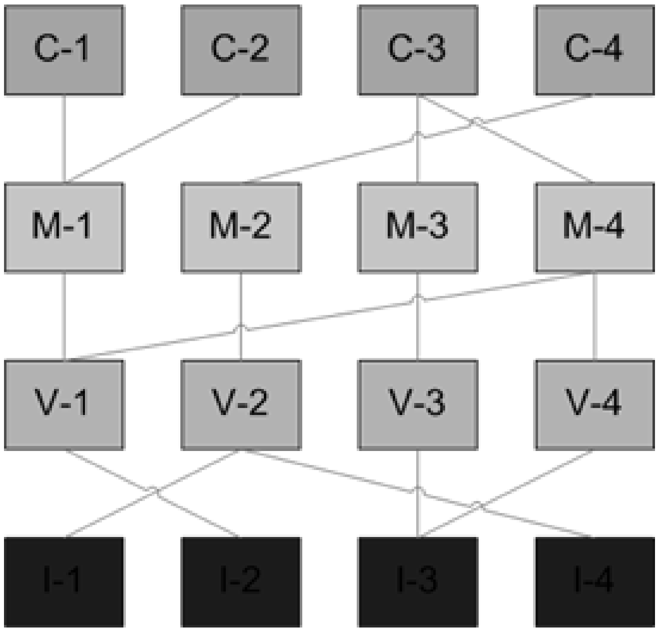

Figure 7.2: Example of EAM concerns, M-, V- and I-Patterns and their connections

The EAM Pattern Catalog version 1.0 contains 43 concerns, 20 M-Patterns, 53 V-Patterns, and 47 I-Patterns. This information hints towards an important aspect of this model: the relations between concerns, M-, V-, and I-Patterns are very rarely one to one, i.e. a number of concerns can share the same M-Pattern, which can then be visualized with the help of different V-Patterns, which in turn are based on one or more I-Patterns. This allows the Pattern Catalog to remain flexible and to provide a broad spectrum of suggestions in connection with a specified concern.

Another noteworthy advantage of the Pattern Catalog is that it combines established best practices from both practitioners and the academic society. The selection of patterns and concerns is based on an extensive online questionnaire and study done by the SEBIS Chair of Informatics of the Technische Universität München, as well as general literature and other research projects. This makes the Pattern Catalog a promising candidate for the creation of a common ground for EAM.

7.3 Project approach and development

7.3.1 Context and scope

Some 5 years ago, the broadcasting company in Bavaria (Germany) introduced a new application in two autonomous business units. Both business units are archives – one for radio and one for television - and the new application was a content management system (CMS), which could store, manage, and retrieve data. After the initial deployment, in 2004 for television and in 2005 for radio, each of the two business units used, developed, and maintained the application independently.

The newly founded IT-strategy department of the company now wants to find out how the CMS is used in each business unit in detail, in order to identify possible synergistic effects or other improvements regarding the installed system.

7.3.2 The problem and concerns

A first glimpse at the IT-strategy department showed that the CMS is used in very different ways in the two business units. So in a first kick off meeting the following project goals for the team had been defined:

1. Creation of a visualization of the three main business processes – input, processing, and output – of each business unit with a process modeling tool.
2. Visualization of the interfaces between CMS and other applications
3. Comparison of the main business processes from both business units
4. Suggestions for improvement (possible EAM tool, process optimization)
5. Documentation of the used methodology and results

The overall goal of these individually defined project goals was to get a comprehensive overview of the business processes support provided by a particular core application, in our case the CMS. After the analysis phase, we are able to show which activities of the chosen business units are executed with the help of the application under investigation and which activities are using other applications and why. The results and the gained knowledge are the foundation of suggestions for future improvements.

7.3.3 Analyzing multiple system installations for synergy potential

Because work needed to be done simultaneously in both business units, the team was divided into two smaller teams, one for each business unit. The teams met regularly for synchronization and progress updates. The general approach for this process was a three-phase model, as shown in Figure 7.3.

7.3.4 Project development

7.3.4.1 Phase 1 – information gathering

Following the approach presented in Figure 7.3, the teams started their work by interviewing different employees from the respective business units. Furthermore, an additional general orientation tour facilitated the understanding of the overall situation. After the first interviews and meetings, the teams decided to represent the processes in each archive department using a *life cycle model.* This representation depicts the way a specific type of content – in this case also called an "essence" – runs

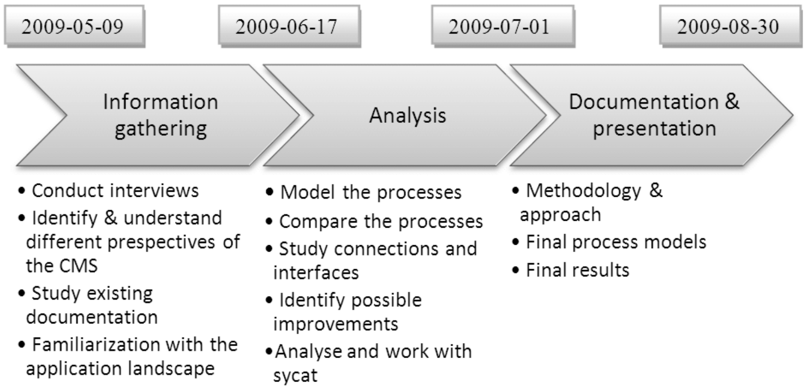

Figure 7.3: Three-phase project approach

through the three main processes input, processing, and output. This way, the use of the CMS in each phase of the life cycle could be closely observed.

The main sources of information were interviews with different employees involved in each process and sub-process in the two archive departments. Our contact persons provided a list of potential interview partners and helped us to arrange the necessary appointments. The interviews were purposely unstructured, because we wanted each employee to have the freedom to describe the processes he/she is involved in from his/her own perspective, thus highlighting the parts of the workflow that are important to him/her. Although it may be argued that this approach may return very subjective results, we are confident that the important interactions with the system will still be identified, with the added bonus value of the highly explorative character of the interviews.

The gathered information was mainly used to model the processes with the processing modeling tool, which will be described in Section 7.3.4.2. Throughout the interviews, the interview partners were asked to describe their daily routine, e.g. to explain for which business process they are responsible for or are involved in, how they do their work, what supporting systems they use, and to what extend.

In addition to the interviews, the teams analyzed available material documenting the current application landscape and the interaction between individual systems. These documents were subsequently received and provided valuable insights into the current situation of the IT landscape, which already pointed towards a very different development of the CMS in the two business units.

At the conclusion of the first phase, the two teams discovered that each business unit had a different focus in the life cycle model of an essence. The radio archive focused mainly on input processes, offering a broad spectrum of input possibilities; processing remained relatively straightforward, and a great part of the administrative work related to output was passed on to other departments. This implied that the number of business processes belonging to the input category was very large. The television archive however concentrated on the processing and enrichment of their essences and their potential output; this was due to the need to process the essences in great detail and because the archive department also handled the management of the usage rights of their essences. This meant that their focus was on the processing and output categories.

7.3.4.2 Phase 2 – snalysis

Due to the differences between the process categories in each business unit, the teams decided to focus on a certain number of comparable processes, which were to an extent shared by both the radio and the television archive. These will be shown in Section 7.4.1. The teams then proceeded to the analysis of their findings in relation to the concerns that were expressed by the two business units of the broadcasting company. To this end, a pattern-based approach was chosen.

Pattern-based proceeding

As previously mentioned, a significant part of the student project dealt with examining the core processes and understanding the role of the CMS. To realize that in a structured manner, a pattern-based proceeding was chosen. The applied patterns were selected from the EAM Pattern Catalog of the chair for Software Engineering for Business Information Systems (sebis) at Technische Universität München [Bu09a]. The use of the Pattern Catalog was motivated – apart from the need for a structured proceeding – by the idea to reuse existing knowledge about EAM and its inherent problems. A hierarchical method was employed to identify the appropriate patterns. First, the present concerns of the broadcasting company were detected (see Section 7.3.2). Then, these concerns were mapped to available concerns of the Pattern Catalog. This resulted in two mainly suitable concerns:

- Which Business Applications are used by which Organizational Units? (C-33)
- Which Business Processes are supported by which Business Applications? (C-87)

From that point, the EAM Pattern Catalog provided applicable patterns, which led in each case one level deeper in the pattern hierarchy (see Figure 7.4).

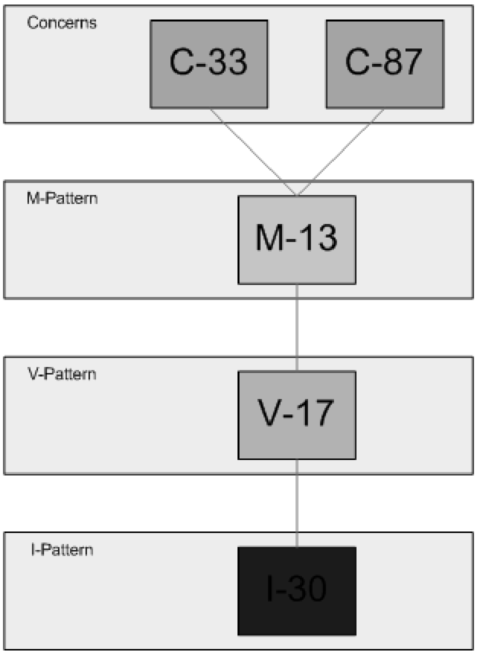

Figure 7.4: Applied patterns in hierarchical order

As a logical consequence, the M-Pattern *Analysis of the Application Landscape* (M-13) was applied. It specifies a methodology to achieve a static view on the application landscapes of an enterprise, regarding mainly how and to what extent business and IT are aligned. To visualize the results of the analysis, the V-Pattern *Process Support Map* (V-17), was used. It is directly linked to M-13 according to the Pattern Catalog and shows relationships between business applications, business processes, and Organizational Units. V-17 is based on the I-Pattern *Process Support* (I-30), which describes how information is organized around processes. Figure 7.4 illustrates the described pattern application. It has been created on the basis of Figure 1.2 of the Pattern Catalog [Bu09a].

Beside these elements of the Pattern Catalog, additional visualizations have been created during the project. These were necessary because process support maps are focused on the structure of architecture rather than on the flow of information between the different systems. As the need for a process-oriented illustration was obvious (according to the goals of the project) and the broadcasting company wished to obtain a visualization, which fitted the existing ones, the process modeling tool of a consulting company was chosen to create the process diagrams. It is noteworthy to say that there are patterns in the Pattern Catalog, which are very close to the process modeling tool's models. These patterns and other patterns, which could also have been relevant in this case are described in Section 7.5.2.

Implementation and portrayal

For a structured graphical representation of the information, two programs were used: the aforementioned process modeling tool and Microsoft Visio.

This process modeling tool, which is in use at the broadcasting company, employs a proprietary database to store information about business units, processes, and the systems supporting them. The tool itself is a process design tool, so naturally its main focus is around designing and managing business processes. The basic elements of a process designed with the tool are organizational units (function areas, displayed as gray, arrow-like shapes), steps of a business process (rectangles), and information elements (modified rectangles). Tool-based process models are created from the perspective of the user who executes that process; this entails that the process is made up of a series of consecutive – sometimes also parallel – steps or actions, which take place in certain function areas, while exchanging different information elements. The processes cannot switch from one function area to another, without an information object being exchanged between the two. An exemplary visualization of a process model can be seen in Figure 7.5.

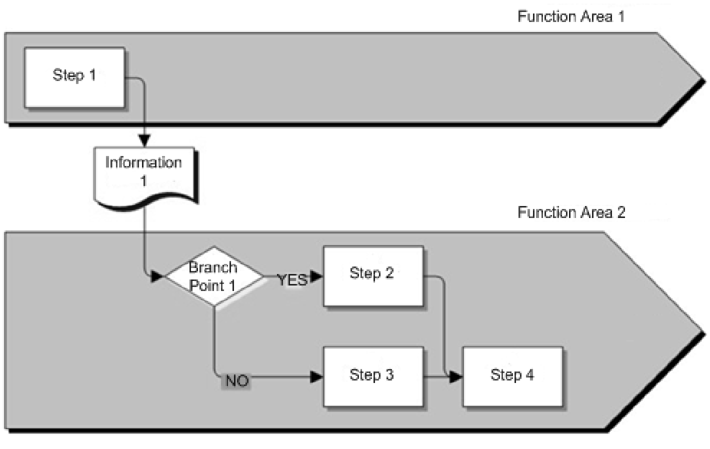

Figure 7.5: Basic process model elements and process layout

The process modeling tool, although appreciated for its relative flexibility and existing widespread use in the company, has one important weak point: the relationship between IT system and process is not displayed. This information could be entered in the database and each step could have an assigned supporting system, but this could not be made visible directly. The only way to get this information in relation to a process was to export spreadsheets from the process modeling tool. These files contained all the necessary information, but the possibilities of representing it in a way that would fit our pattern-based approach were very limited.

Confronted with this problem, the teams initially suggested altering the perspective of the process modeling tool in such a way that would allow supporting IT systems to be represented alongside the function areas. This option was however rejected by the broadcasting company, because they feared that the same representation for both function areas and supporting systems would create confusion with regard to which parts of the process are executed by the employee and which parts actually take place automatically in a computer application. Additionally, such a modification would conflict with the company policies for process modeling tool's models and would further complicate matters. This option was thus abandoned.

After careful consideration, it was decided to stick to the standard models of the process modeling tool and use the visualizations that could be derived from the V-Patterns to obtain a comprehensive model of the landscape around the two CMS systems.

For the process support map Microsoft's Office program Visio 2003 was used. Visio is a very powerful drawing and modeling tool, which offers a great number of templates and shapes that can be freely used on the canvas. By providing a detailed legend to our process support maps, in accordance with the requirements of EAM patterns, we ensured that the meaning of each shape and the arrangement of different elements would be clearly conveyed to the reader.

7.3.4.3 Phase 3 – documentation and presentation

In this third and final phase of the project, the teams concentrated their efforts on reviewing and correcting selected processes and diagrams, with the assistance of the partners at the broadcasting company. The results of the project, which will be described in Section 7.4 of this paper, were compiled into a series of final presentations, for both the broadcasting company and the sebis chair at the Technische Universität München. After the results were presented, work began on the documentation of the team's activities, approach, and results in this project. This paper is a structured report on the project at the broadcasting company and represents an important part of the overall documentation.

7.4 Results

This section presents the results obtained by each team for their respective business unit. As previously mentioned, it was important that the results of the two analyses are directly comparable. Thus, it is considerably easier to draw the proper conclusions and provide useful suggestions for the future evolution at the end of the project.

7.4.1 Process overview – radio archive and television archive

As with many other large enterprises, there is a great number of (business) processes that are involved in the daily operation of the broadcasting company. In order to be able to achieve concrete results the two teams each focused on the process directly or indirectly connected with the CMS in each business unit. This area of focus was selected, due to the short time frame of the projcet and on demand of the broadcasting company.

During the first and second phase of the project, each team identified a relatively large number of processes in each archive department. For the radio archive, the following different processes of the taypes input, processing, and output were identified:

Input

- Digitalization
  - E-Music Digitalization
  - L-Music Digitalization
  - Tape/Vinyl Digitalization
- Import of essences
  - IDAS Import
  - Digital Sampling
  - Standard Import
  - Word (DigAS)
  - Productions (DigAS)
- Import and review of basic metadata
  - Audiopass Import
  - Sequoiapass Import
  - Cuepoint attachment
  - DigAS Metadata

Processing

- Formal capture and indexing
  - Documentation Word
  - Documentation Music
- Compilation of digital rights
- Scan extra material
- Edit audio file (Sequoia)

Output

- Research preview of digital archive contents in the Medienbroker
- Orderes (including digitalization if needed)
- Export digital archive contents to DigAS for streaming & production
- FTP Export

For the television archive, the identified processes were the following:

Input

- Digitalization of Euronews material

- Import
  - Prepare material import
  - Import sending material
  - Import Euronews material
  - Import thematic archive material
  - Import inter-archive material

Processing

- Add Metadata
  - Maintenance and Technical Metadata
  - Formal Metadata
  - Content Metadata
  - Rights Metadata

Output

- Research
- Order essence
- Transfer essences
- Prepare inter-archive material
- Content preparation for 35 different end-systems

Out of these identified processes, a number of specific processes, which were similar in both archive business units were chosen. Thus, a better comparison of the use of the CMS could be achieved. This also gave the teams the chance to focus on those specific processes and study them in detail. The processes that were chosen for the comparison were selected by the partners from the broadcasting company, according to similarity criteria, which were defined and set by them.

The processes chosen for further analysis for the radio archive were:

Input: Digitalization (E-Music and L-Music), IDAS Import, Standard Import, Digital Sampling

Processing: Documentation Word, Documentation Music

Output: Research, Preview of digital archive contents in the Medienbroker, Export digital archive contents to DigAS for streaming & production

For the television archive, the following processes were selected for further investigation:

Input: Prepare material import, Import sending material, Import Euronews material

Processing: Add formal Metadata, Add content Metadata, Add rights Metadata

Output: Research

For the chosen processes the teams carefully reviewed the information at hand and organized additional interviews and question and answer sessions if more information about the processes was needed. After every detail was clarified, the processes were modeled in the process modeling tool according to the previously described company modeling policies and guidelines. See Appendix B an example of a modeled process.

Although the graphical models in the process modeling tool were proved useful for a simple and easy to understand depiction of the business processes, supplementary visualizations according to the identified EAM pattern, detailing on special aspects also need to be created. For these models, information about which parts of the processes are supported by which IT system was required. This information existed in the database of the software tool, although it was not possible to display it in the process models. The spreadsheet export function of the process modeling tool was used to create a table of processes and used systems (cf. Figure 7.6).

		System											
	Processes	EAC	Internet Explorer	CMS	Medienbroker	Microsoft Outlook	Microsoft Word	MUSAD	Notepad	Production system	Sequoia	Windows Explorer	WOSAD
	Overview radio archive												
Input	**E-Music Digitalization**	X		X	X	X					X		
	L-Music Digitalization	X		X	X	X							
	Digital sampling		X		X								
	Standard import			X		X						X	
Processing	**Documentation Music**			X		X	X	X	X	X	X	X	
	Documentation Word			X		X	X			X	X		X
	Archiving from Documentation Word		X		X					X			
Output	**Research in Medienbroker**		X		X								

Figure 7.6: Processl modeling tool export of processes and supporting systems

A similar table was created for the television archive. With the help of these tables, a mapping of the process to its supporting systems and applications was possible. Furthermore, the process support maps (V-Pattern 17) could be designed based on these tables.

7.4.2 Process support maps

Process support maps have a two-dimensional cartesian layout (cf. citeWi07b). Business processes are represented on the x-axis in a logical sequence. On the y-axis, the actor involved in the process is represented. Following the intersection of actor and process, one can find the supporting application that assists the actor during that process. Figures 7.7 and 7.8 show the process support maps for the radio and television archive, respectively.

7.4.3 Comparison of radio and television archive

After all visualizations were created, the analysis and comparison of the utilization of the CMS in two business units could start.

Perception and metadata

The first difference that was noticed was the way that the CMS was perceived in each archive department. As shown in two process support maps, the CMS system is seen as a "whole", a single entity in the radio archive, which is involved in every process. This is different for the television archive, where the CMS is seen as an aggregation of subsystems, like CMS Catalogger, CMS Transfer Monitor,

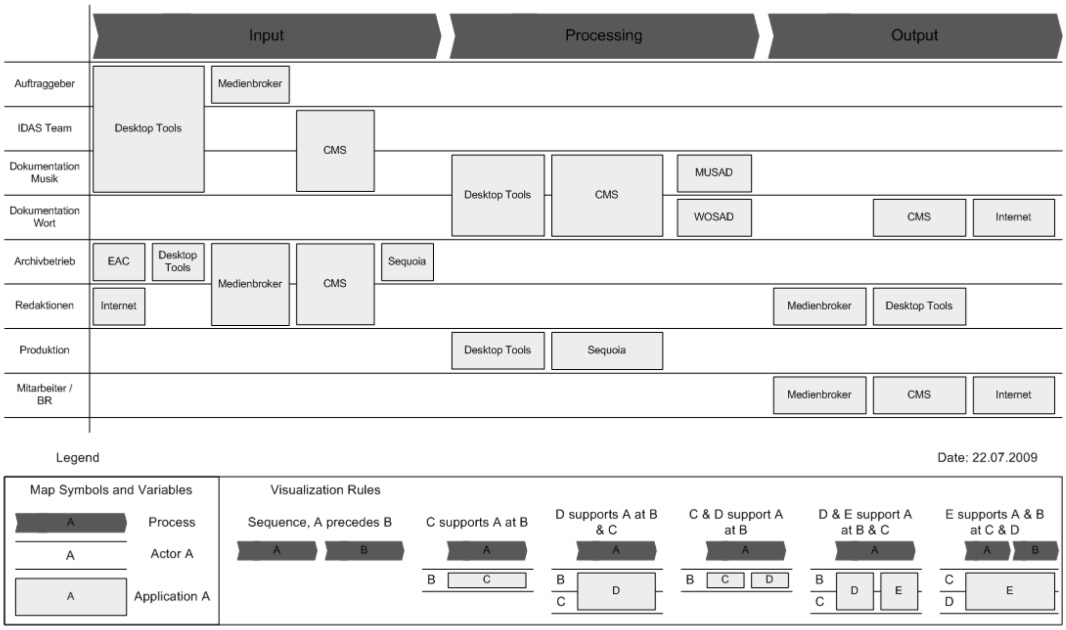

Figure 7.7: Process support map for radio archive

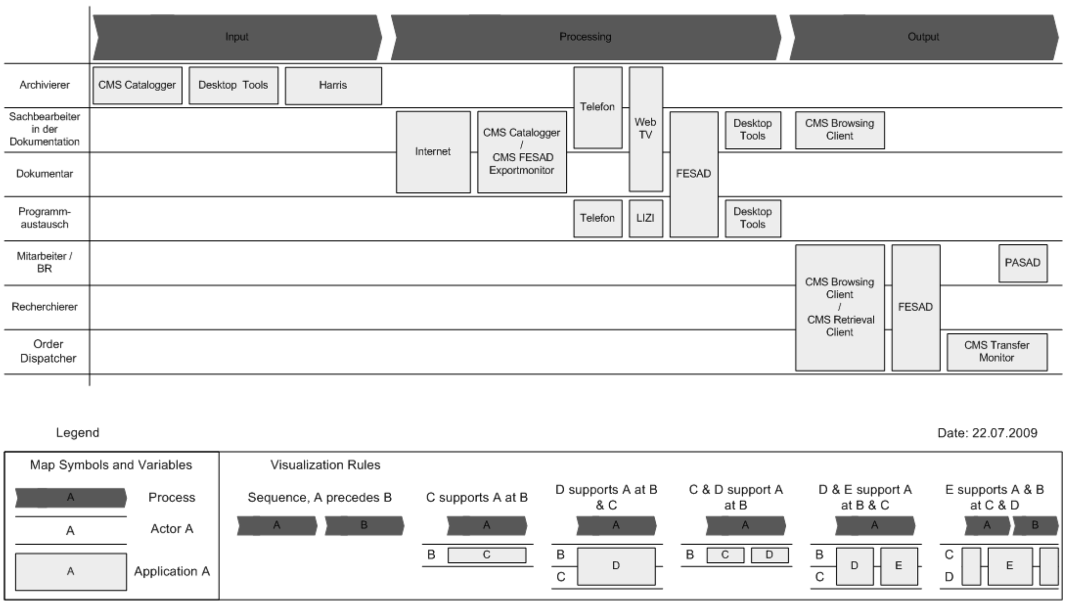

Figure 7.8: Process support map for television archive

etc. The CMS is thus more intensively used in the television archive, partly because it is the main database for meta information (metadata on essences). Metadata is primarily stored and maintained in the CMS, and the synchronization with the corporation-wide used SAD database follows afterwards. However, for the radio archive, the SAD database is the preferred application for metadata storage. In this case, the CMS system also stores part of the metadata, but more as a backup to the SAD database than for actual productive use.

Systems interaction

Both process support maps clearly show that the CMS is a vital system in both of the business units' daily activities. However, it can be observed that there are a lot of other systems that are required to work together with the CMS. These systems range from the most common office applications like a mail client (e.g. Microsoft Outlook), via text-editing software (e.g. Microsoft Word and Windows Notepad), and internet browsers (e.g. Microsoft Internet Explorer) to file system browser (e.g. Windows Explorer). These programs are compiled under the category *desktop tools* in our process support maps, but they can be visualized individually in the two spreadsheets of the process modeling tool (see Appendix C and D). Although such systems are a clear necessity in every corporate working environment, it was often observed during the interview process that the information transfer between these individual systems occurs with a risk of error and failure. This can be dedicated to the fact that the employees often have to bridge the technical gap between these systems with manual work. An example is the printing of documents, which are later read and information from them is transfered manually to another application. This is also valid vice versa, meaning that a case was observed where a database ID number generated by the CMS needed to be added to a formal document. For this, the document was printed and the ID was written on the paper printout by hand. This media discontinuity and implicit transitions from digital to paper and vice versa are a known point of failure, where human errors can disrupt or alter the information flow from one system to another. This practice was observed in both archive departments.

Information retrieval

Another notable difference is the way information is retrieved from the CMS in the two business units. The radio archive rarely uses the interface of the CMS to directly access the archived content. Instead, research and preview of essences is done via the *Medienbroker*, a web application, which is directly connected to the SAD database and the CMS. The Medienbroker is the central entry point to the archive: it allows the user to research an essence in the SAD database by entering relevant metadata information (author, title, etc) and then identifies the corresponding essence in the CMS.

The essence is then available for export through the Medienbroker to the desired destination and in the desired format. Opposed to this, the television archive uses the CMS system directly. The archive is browsed directly through the CMS and the metadata research is also done in the CMS, since for the television archive it is the main system for metadata storage and management.

Usage, access rights, and terminology

The managed metadata also differs in each department. The difference in this is case is the management of metadata about usage rights. As the broadcasting company has a vast number of audio and video files at its disposal, an important aspect is to assure that these essences are only used by those who also have the right to use or publish them. This can be a very complicated task if the enormous amount of files and the possibly very large number of users are considered. The usage rights, thus need to be properly managed. The radio archive outsources this task to a specialized department of the company, which is charged with managing these rights and making sure that the users requesting files have the right to use them, and have eventually paid for it. In the television archive, the data on rights is managed by the archive department, along with the usual formal and content related metadata. This information is also stored in the CMS. In this sense, the broadcasting company speaks of the radio archive as managing *essences* while the television archive stores *assets*, because their essences are completed with metadata about rights.

Interaction

Finally, the way users interact with the CMS in each department is different as well. As the radio archive uses the Medienbroker application, users never interact directly with the CMS. The only employees, who directly use the CMS are those of the archive department, when they are performing imports and processing of essences. Internal company access, as well as access for external users is done exclusively through the Medienbroker's web interface, using a standard browser. The situation is different in the television archive. As the range of functions of the CMS's web interfaces exceed those of the Medienbroker with respect to preview and selection of the archive's content, not only employees within the archive department itself use the CMS directly for searching and ordering but also editors and other staff members, of the public broadcasting company.

7.5 Outlook

7.5.1 Future work

In this report, we described the methodology used to tackle the company's issue of two diverging installations of a CMS used in the archive departments of radio and television. Due to the high level of independence of the individual departments within this company and the rather different content that needs to be managed, the two installations of this CMS developed very differently. Now, a recently introduced – IT-strategy & -planning department – is looking for systems that could possibly allow for the realization of synergistic effects. As the introduction of the same CMS in the two different archive departments was originally intended to realize synergistic effects, it was an adequate candidate for examination. Unfortunately, as shown in Section 7.4.3, we could hardly identify any synergistic effects in the usage of the CMS. Even if originally the same type of the CMS was introduced in both archive departments, the perception and usage of this application by the staff as well as its embedment in its surrounding system landscape are highly different. Each system developed in accordance with the needs of the respective department.

As we discovered that the systems themselves provide only very limited possibilities for the achievement of synergies, we considered suggesting a different starting-point. Instead of looking for synergies on the system level, we recommend the further analysis of the business processes in both archive departments in order to identify best practices. Since we found some disadvantageous steps within the business processes we have analyzed, we think that improving these steps would bring a greater advantage. Furthermore, we suggest that our work should be continued and even expanded to other departments (e.g. production) of the public broadcasting company, which may disclose further potential for improvement. Especially for a further development of the business processes the input of other departments is a highly valuable source. It is even thinkable that not only the business processes in the single departments have to be changed, but the structure of organization as well. Further investigations would be useful for clarifying in this point.

Moreover, we suggest introducing and using a sophisticated EAM tool instead of the presently used process modeling tool, because of its shortcomings detected during our current project, (cf. Section 7.3.4.2). For decision support in selecting a proper tool one can use the EAM tool survey [Ma08] carried out by sebis. This survey evaluates the products of major players in the market of EAM tools. For the introduction of an EAM tool it is recommended that an elaborate testing phase is performed so that its usefulness and constraints could be experienced. For the testing phase a project like this one would be a good starting point.

7.5.2 Further relevant patterns

Due to the scope and time constraints of the student project, only the most suitable patterns could be applied. Nevertheless, the Pattern Catalog [Bu09a] offers further concerns and patterns, which could be of some interest. Although the Pattern Catalog contains several suitable concerns, most of them do not completely fit with the requirements, which the broadcast company stated. But some indeed refer to questions, which came up when analyzing the company.

Obviously, a process, which includes both employees and computer systems has transition points, the so-called human-computer-interface. As an additional result of the conducted analysis, some weaknesses in this area have been recognized. However, the judgment of the human-computer-interface was mainly influenced by the first impression, meaning that a structured examination would still be necessary to allow for more meaningful statements. The following two concerns are related to the human-computer-interface and could serve as the origin for such an analysis: "How can a more continuous IT support concerning business processes be realized?" (C-95) "To which extent are the business processes supported by business applications? Which business processes are supported manually? Can the automated support be extended?" (C-78) The associated methodologies are the "High Level Process Support" (M-29) and the "Business Process Data Flow Analysis" (M-30). The results could be illustrated by the V-Patterns "Process Support Map visualizing horizontal Integration" (V-28) and "Cluster Map visualizing Business Object Flows between Business Applications" (V-48). Both visualizations are based on the I-Pattern "Process Support" (see section 7.3.4.2) and the latter one in addition on "Interfaces and Information Flows" (I-63).

It was also previously mentioned, that an additional visualization was created on demand of the broadcasting company. However, the need for a better overview of the processes in general could not be matched to a specific concern from the Pattern Catalog. The methodology, which delivered the data for the process modeling was very similar to the M-Pattern "Process Analysis" (M-6). For example, a relatively high level of granularity was expected, details of the activities were disregarded and the methodology was "targeted at employees responsible for specific business processes" as the Pattern Catalog states for M-6. Following the hierarchy downwards from the process analysis, the connected V-Pattern is the "Business Process and Business Function Relationship" (V-12). Its similarity to the used visualization of the process modeling tool can be recognized as there are similar elements with related meanings. However, the process modeling tool includes much more complex control flows as it allows e.g. branching and loops. The I-Pattern "Process Landscape" (I-12), which belongs to V-12, can be used to structure and explain the information of the tool's models without substantial change.

7.6 Conclusion

7.6.1 Lessons learned

During each project one acquires not only insights on the analyzed issue, but also learns how to do things better; these are the lessons learned. First of all, we recognized that the expectations and goals of the principal should be much more clearly formulated and communicated. Furthermore, the main goal and its sub goals should be recorded in written form and their feasibility should be discussed with all main stakeholders before the project starts. In addition to that, it is necessary that the principals and the advisor attend eventual status and synchronization meetings. This additional information and coordination effort would enable the goals to be adjusted and formulated more precisely.

Another lesson learned is that it may be more efficient to use only one team with four to six members and proceed in a sequential manner. We were a big team with eight members, which split into two sub teams with four members each. Each of the two sub teams investigated one archive department.

Therefore, the sub teams got the same view as the staff of the analyzed department. Thus, both teams had the same barrier in communicating and exchanging the gathered information during the synchronization meetings as the staff had encountered before during former attempts in finding synergetic effects. By using a sequential approach with a smaller team, the synchronization effort can not only be reduced, but it will also facilitate team members and the team as a whole to gather a greater breadth of understanding through obtaining the viewpoints of both departments.

With respect to the interviews we carried out, we learned that it is not enough to take notes during the interviews. Some of the information deemed unnecessary at first was needed later on but not conveniently available. Therefore, we recommend using an audio or even video recording during the interviews. In addition, the unstructured interviews could be enhanced by a small standardized questionnaire asking questions about the interviewee's opinion about insufficient process steps, possible synergies, or other generally relevant questions.

One last thing we did not quite learn but rather re-experienced is that a tool, which is not intended to fulfill some task is not the proper tool to use for the respective task. From the beginning, the process modeling tool disclosed its weakness with respect to the aimed-at goal of visualizing which business application is used during the fulfillment of the observed business processes. The rigid modeling conventions we had to conform to amplified this weakness even more. We therefore suggest either to use a different tool, which allows the realization of the aimed-at models or to opt for a more loosened modeling convention in order to get the desired results.

7.6.2 Acknowledgement

We want to thank all participants of this project for the help, especially our advisors in the two departments for their endurance and their helpfulness. Also, we thank all our interviewees, who explained their workstations in detail with live executions of their main tasks and the answering of all our questions, in fact we asked a lot of questions. The feedback in the synchronization meetings from different company workers, e.g. project manager of other or older projects in the IT area at the company, was very informative and valuable for us as well. Last but not least we want to thank the company project coordinators for their always-open ear for our special needs and the good cooperation during the project.

Appendix

Appendix A: Complex process designed with the process modeling tool

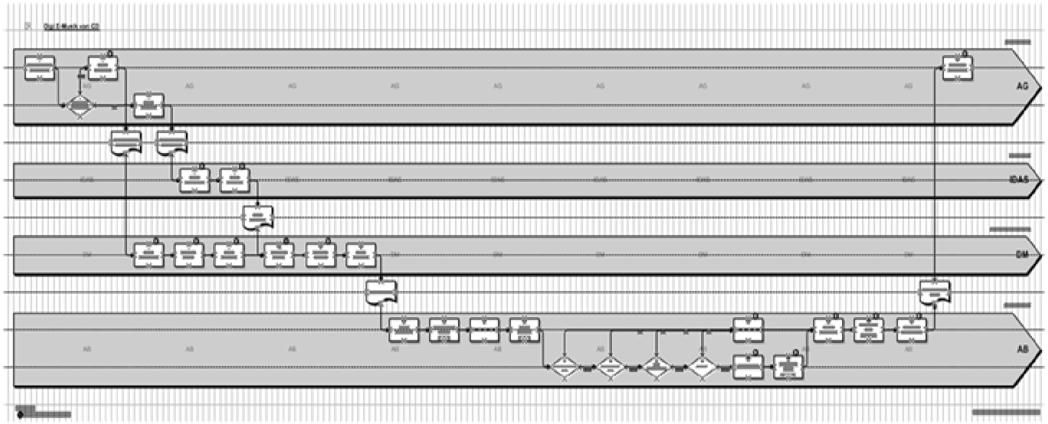

Figure 7.9: Screenshot of a process overview in the process modeling tool

Appendix B: Process model spreadsheet for Radio

Processes	System											
	EAC	Internet Explorer	Media Archive	Medienbroker	Microsoft Outlook	Microsoft Word	MUSAD	Notepad	Production system	Sequoia	Windows Explorer	WOSAD
Overview radio archive												
E-Music Digitalization	X		X	X	X					X		
L-Music Digitalization	X		X	X	X							
Digital sampling		X		X								
Standard import			X		X						X	
Documentation Music			X		X	X	X	X	X	X	X	
Documentation Word			X		X	X			X	X		X
Archiving from Documentation Word		X		X					X			
Research in Medienbroker		X		X								

Appendix C: Process model spreadsheet for Television

Processes	System														
	CMS Browsing Client	CMS Catalogger	CMS FESAD Exportmonitor	CMS Retrieval Client	CMS Transfer Monitor	Desktoptools	FESAD	Harris Media Client	Harris XMediaBrowser	Internet	LIZI	Paper-based	PASAD	Telephone	WebTV
Overview television archive															
Digitilaze Euronews tape								x							
Import sending material		x			x	x			x			x			
Import thematic archive material		x			x	x			x			x			
Add formal Metadata	x	x	x				x			x		x		x	x
Add rights Metadata		x				x					x			x	
Research for editors	x			x											
Research / order / transfer DVD or tape	x			x					x						
Maintain database	x			x											
Internal research	x			x				x							
Research / order essence							x						x		
Process order									x				x		

Bibliography

[AIS77] Alexander, C.; Ishikawa, S.; Silverstein, M.:. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press. 1977.

[Bu96] Buschmann, F. et al.:. *Pattern-oriented software architecture: a system of patterns*. John Wiley & Sons, Inc. New York, USA. 1996.

[Bu08] Buckl, S. et al.:. *Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008)*. Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2008.

[Bu09a] Buckl, S. et al.:. *EAM Pattern for Consolidations after Mergers*. In *SE 2009 – Workshopband*. Kaiserslautern. 2009.

[Bu09b] Buckl, S. et al.:. *State of the Art in Enterprise Architecture Management 2009*. Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2009.

[Er06] Ernst, A. M. et al.:. *Using Model Transformation for Generating Visualizations from Repository Contents – An Application to Software Cartography*. Technical report. Technische Universität München, Chair for Informatics 19 (sebis). Munich, Germany. 2006.

[Fo02] Fowler, M.:. *Patterns of Enterprise Application Architecture*. Addison-Wesley Longman. 2002.

[Ga94] Gamma, E. et al.:. *Design Patterns: Elements of Reusable Object-Oriented Software (Addison-Wesley Professional Computing Series)*. Addison-Wesley Professional. 1994.

[IE00] IEEE:. *IEEE Std 1471-2000 for Recommended Practice for Architectural Description of Software-Intensive Systems*. 2000.

[Ma08] Matthes, F. et al.:. *Enterprise Architecture Management Tool Survey 2008*. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2008.

[St73] Stachowiak, H.:. *Allgemeine Modelltheorie*. Springer-Verlag. Wien, Austria. 1973.

[Wi07] Wittenburg, A.:. *Softwarekartographie: Modelle und Methoden zur systematischen Visualisierung von Anwendungslandschaften*. PhD thesis. Fakultät für Informatik, Technische Universität München, Germany. 2007.

CHAPTER 8

EA Management Patterns for Analyzing Business Applications

Patrick Blitz (blitz@mytum.de),
Johannes Ehm (johannes.ehm@mytum.de),
Philipp Andre Heinemann (philipp.heinemann@mytum.de),
Dinh-Nam Nguyen (dnn@mytum.de)

Abstract:
Business applications play an important role in a majority of current enterprise architecture (EA) management approaches. In this vein, most of the approaches agree on the importance of documenting the applications and their relationships, but only a few approaches go much further. In this paper, we propose EA management patterns to analyze the "health" of a business application in the context of an EA.

8.1 Motivation

Due to limited software developing capabilities, a company must focus its development efforts on specific IT projects. The decision should be comprehensible and therefore driven by *quantitative* information about the application landscape. For this purpose, *enterprise architecture (EA) management* encompasses the *Analyze EA* activity. The activity analyzes the current, planned, and target EA, as represented in architectural descriptions, and provides with the EA analysis results an artifact, which contains information that could be used to decide whether a specific project proposal should be executed or not. Figure 8.1 shows the activity and the thereby created or processed artifacts.

Different ways to analyze EAs or application landscapes, as a part thereof, exist. They range from informal analyzes based on the "gut feel" of the enterprise architects to formal analysis techniques employing metrics. Metrics can be used in the context of application landscape management to measure specific characteristics of the *Application & Information Layer* [Bu09]. Concerning the possibly interesting characteristic, a broad variety of options exists, ranging from business- over function- to operation-related ones. Figure 8.2 shows a holistic view on an enterprise and indicates how the metrics cross function relates to the *Application & Information Layer*. With the obtained quantitative information metrics provide a more objective basis for reviews of the applications landscape, decision making,

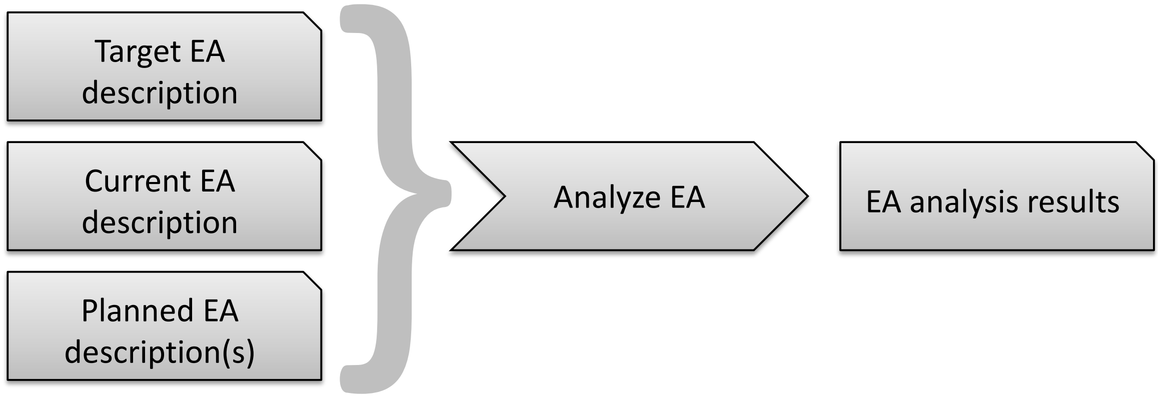

Figure 8.1: Analyze EA activity of the EA Management [Ch09]

and controlling of goal achievement [LS07]. Metrics may further help to bridge the communication gap between the stakeholders from business and IT [Bu08b].

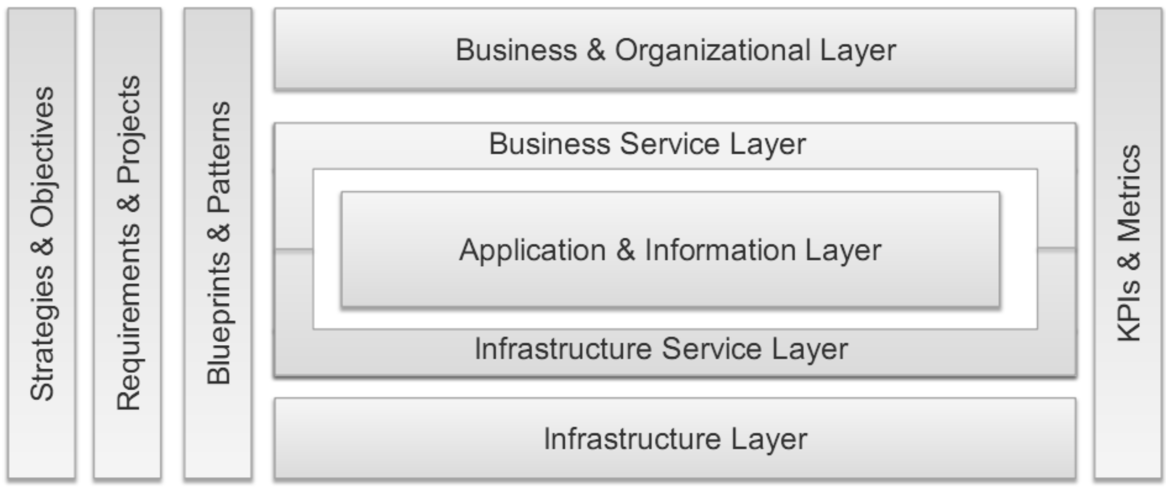

Figure 8.2: Holistic view of an enterprise with the Application & Information Layer and the Key Performance Indicators & Metrics as a cross function [Ch09]

This paper provides *EA Management Patterns* that in accordance to [Bu07] describe proven-practices for performing EA management or subactivities thereof. In particular, the patterns contains in this paper describe the use of metrics to analyze and communicate specific characteristics of an application landscape. The patterns were developed, put into action, and validated at the IT provider of a financial service provider. In the notion of [Bu07], the paper presents three types of patterns: a Methodology Pattern (*M-Pattern*) Business Application Rating explains necessary steps to analyze a characteristic of an application landscape and to retrieve accurate quantitative information. The Information Model Pattern (*I-Pattern*) Business Application Rating gives a model of the gathered information. The two Viewpoint Patterns (*V-Pattern*) visualization of component strength and weakness (cluster map) and visualization of the landscapes degree of performance (radar chart) provide two possibilities to visualize the gathered quantitative information.

The aforementioned V-Patterns rely on the principles of Software Cartography [Ma08], a technique for representing application landscapes in an intuitive and graphical way. The used visualizations are called software maps and resemble the characteristics of maps from "conventional" cartography, e.g. are structured into a base map and additional layers. In the field of software cartography, a software map may contain application systems on its base layer, while additional layers can be used to visualize additional specific characteristics of an application. Via this *layering principle*, it is especially possible to add quantitative information about business applications. This possibility is described in the last section (see Section 8.7). Furthermore, the section discusses *EA Tools*, which are used at the financial service provider to realize the corresponding V-Patterns.

The remainder of the article is structured as follows. Section 8.2 explains the application landscape and the issue of the financial IT service provider. The following Section 8.3 describes the proposed M-Pattern. Section 8.4 as well as Section 8.5 propose a visualization of the results of the M-Pattern. Section 8.6 provides the used information model. Finally, Section 8.7 gives an outlook, discusses *EA Tools*, and sketches further ideas.

8.2 Problem statement

The financial service provider runs an application landscape with more than 100 applications. The applications, which are developed by the company itself or by third-party companies, are mostly written in Cobol or Java, and are for internal use only. The company has an existing software map of the application landscape containing the applications itself and the connections between the applications. With this map, an overview on the landscape is provided, while detail information, e.g. on the "health" status of a business application is not given. To augment the map with detailed information about the applications, the company has done several surveys to collect data about the applications, which could be used within the map. The data was collected through interviews and describes age, lines of code, function points, development efforts, or maintain efforts for the business applications.

The company has the goal to find a way to use the existing data along with additional gathered information to support the software map with details about the applications. The details should provide the possibility to compare the applications regarding specific characteristics. The company is especially interested in the complexity and architectural conformity of the application landscape. Architectural conformity measures whether the application uses an architecture which conforms to the guidelines of the company. The complexity metrics accounts for the internal complexity of an application based on its components. Quantitative information about the characteristics should help in the decision making process of projects, which target overhauling and modernization of specific applications. In the past, such decisions were based on the gut feeling of the responsible staff.

In general, the application landscape has been developed for many years and necessarily contains outdated technology and so called legacy applications. Also, enterprises application landscapes often contain a high number of applications and a dense web of interconnections between them. With the growing age, the complexity increases and the architectural conformity decreases because software ages [En09]. Especially older applications that have a monolithic rather than a loosely coupled structure, do not align with the architectural guidelines and are difficult to maintain as well as to enhance [MWF08]. Further, these applications do not align well with the business demands [HW08] and are often erroneous, responsible for high costs and low customer satisfaction. Additionally, it is a problem that applications are changed constantly [MWF08] and that new applications are added as frequently as new connections between applications are established [LTC02].

To prevent increasing complexity and decreasing architectural conformity with growing age, it is necessary to overhaul and modernize the applications in small steps to realize a managed evolution. The described EA Management Pattern can be used to detect the applications with the highest complexity and the lowest architectural conformity, to trigger modernization efforts and measure the improvements made. Although the proposed EA Management Pattern is used to analyze the application landscape of the financial IT service provider regarding the complexity and architectural conformity, the pattern could also be used to analyze other characteristics. Due to the generic structure of the pattern, the pattern is not restricted to analyze only the characteristics of complexity and architectural conformity.

8.3 M-Pattern business application rating

The M-Pattern *Business Application Rating* describes a methodology for analyzing the current application landscape of an enterprise. The methodology should help to identify critical applications and to compare them regarding a specific characteristic.

Context

The enterprise has a large and heterogeneous application landscape. Although an overview of the application landscape exists, the company lacks information in detail. Especially, information for rating and comparing the applications based on their health status does not exist. Thus, the enterprise has limited knowledge about critical applications and no quantitative information for decision making.

Problem

You do not have quantitative information for decision making to establish a managed evolution. Furthermore, you cannot measure the improvement through the managed evolution. You feel the risk of an unmanaged evolution through decision making based on the gut feel of the responsible staff. The following *forces* influence the solution:

Knowledge versus Costs: Does knowledge about the required data exist or are high costs for more accurate information through the efforts of obtaining the necessary data expected?

Quantitative versus gut feel based decision making: Does decision making base on accurate quantitative information or base on estimated information?

Solution

The methodology provides a solution for the problem described above. Figure 8.3 shows the four-step approach of the pattern.

Step One: Collection of Background Information

In the first step, the existing overview about the current application landscape is analyzed. Also, the key interests of the stakeholders and the aims of the analyze activity should be discussed, because this information helps to define the application landscape characteristics. These can be used to derive the necessary data about the application landscape that needs to be collected. The stakeholders should provide possibilities to collect the required data. If the required data does not exist, it is necessary to find a way to retrieve it. Besides interviews, it is possible to retrieve data by using tools. The retrieval of data should be undertaken in close cooperation with the stakeholders as they have the necessary contacts for interviews and the knowledge on the application of tools within the application landscape.

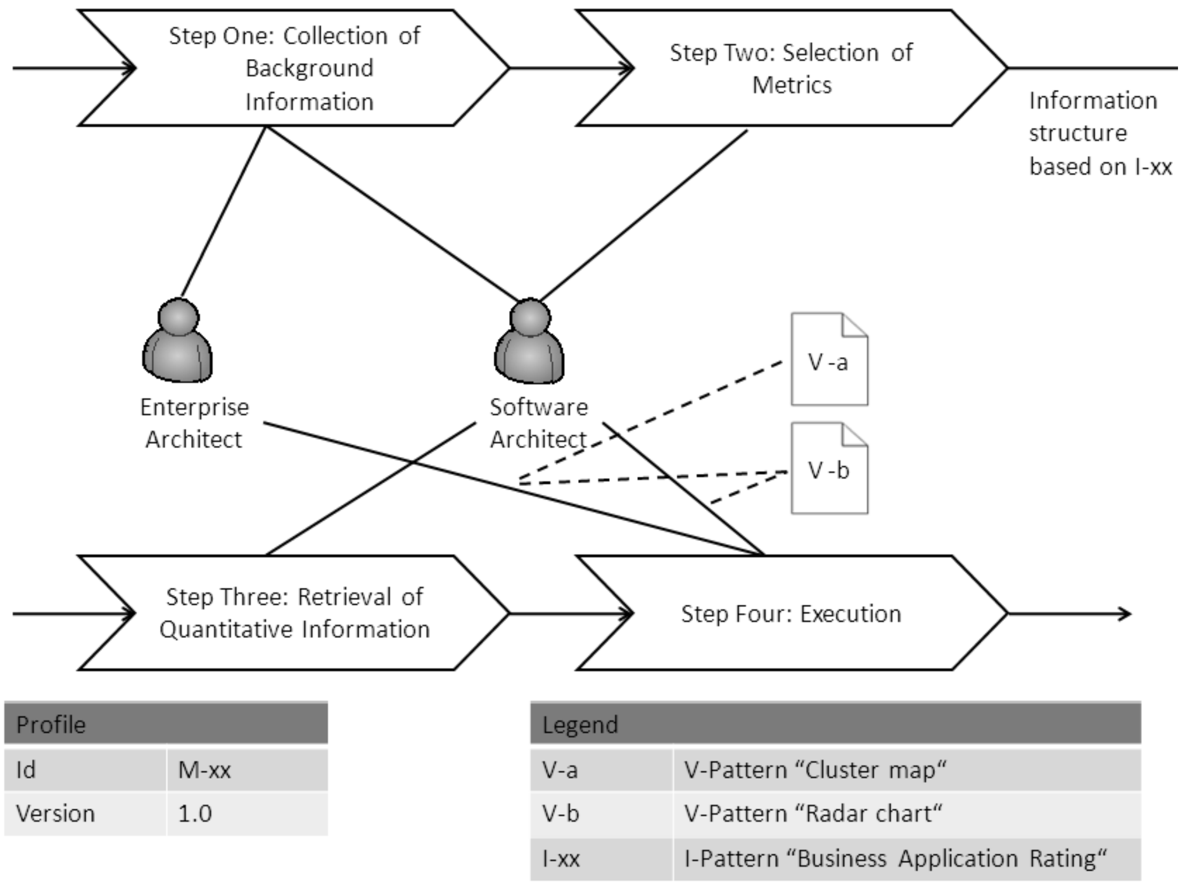

Figure 8.3: M-Pattern Business Application Rating process

Results of this step:

- key interests of the stakeholders and the aims of the analyze activity,
- characteristics of the application landscape, which should be analyzed,
- possibilities to collect the required data

Step Two: Selection of Metrics

The characteristics are usually composed of several metrics (e.g. the characteristic complexity is determined by the application of the metric *lines of code* or *number of control flows within a module*). Therefore, it is necessary to specify the underlying metrics of a characteristic. The metrics could be retrieved and grouped via brainstorming, through literature analysis, interviews with professionals or advice of experts. A so-called `CriteriaGroup` aggregates multiple metrics with similar or related meaning.

Results of this step:

- `CriteriaGroups` belonging to characteristics
- metrics belonging to criteria groups

Step Three: Retrieval of Quantitative Information

In order to obtain quantitative information, it is necessary to define a method, which extracts the quantitative information from the gathered data. The metrics calculate values, which must be aggregated to a value for the whole criteria group. The aggregation should consider a prioritization of the metrics, as not every metric is of the same relevance. Depending on the selected metrics, different approaches exist to retrieve quantitative information. The results of the metrics could be a simple yes or no answer or a score on a ordinal scale. The approach of aggregation has to use the upper and the lower limit and maps the results of the metrics on a numeric value. The range between the upper and the lower limit is divided into several intervals to group the numeric values. Quantitative information contains a score for each criteria group.

Results of this step:

- weights, which indicate the relevance of the metrics
- method to aggregate the results of the metrics and provide quantitative information

Step Four: Execution

In the last step the required data should be collected, the metrics should be applied, and the results should be aggregated to obtain quantitative information. Afterwards it is necessary to visualize the results.

Results of this step:

- collected data
- results of the metrics calculation
- quantitative information about the applications
- visualization of the quantitative information

8.3.1 Example

As described in Section 8.3, the methodology was applied at a financial IT service provider in order to determine the health of applications and to trigger the efforts of overhauling and modernization of specific applications to achieve a managed evolution. The stakeholders provided an overview of the application landscape. They also provided the two characteristics *complexity* and *architectural conformity*, which should be used to detect unhealthful applications within the application landscape.

The metrics were found through literature research. Afterwards, the metrics were divided into several groups. The Table 8.1 and 8.2 in the Appendix 8.7.2 show the various metrics and criteria groups identified.

In order to obtain quantitative information about the complexity, an aggregation method has to be established. The quantitative information should be a score from 0 to 5 points where 0 points indicates a high complexity and 5 points a low complexity. As we could not expect every indicator to score in the corresponding interval, we established transformations to adapt the indicators to the scale. While for ordinally scaled indicators a simple monotonic mapping would be sufficient, more complex considerations apply in the case of interval or rational scaled values. A corresponding mapping function is based on the mean and standard deviation of all retrieved values of a metric..

The mean corresponds to 2.5 points, a positive variation with a length of one standard deviation corresponds 5 points, a negative variation with a length of one standard deviation corresponds 0 Points. The area between 0 and 5 Points is equally divided in 6 subareas, so the (positive, negative) variation between two points is the difference of 0.4 of the length of standard deviation. Using this calculation method, it is possible to obtain for each metric quantitative information.

Due to the fact that the metrics of the architectural conformity only provides yes or no answers, the upper limit is set to 1 and the lower limit to 0. Besides, it is necessary to provide also an aggregation method for this characteristic. Similar to the complexity, five points indicate a high architectural conformity whereas zero points indicate a low and five points a high architectural conformity.

The necessary data was collected through the knowledge of the stakeholders. An MS excel sheet was developed to derive quantitative information from the collected data. Figure 8.4 shows the sheet. The sheet contains for each metric a value and a status field. The value field contains the collected data required for the metric. The status field indicates whether data is available or not. The sheet also visualizes the information through the usage of the V-Patterns.

Complexity

System		Sales		Financial Management		Human Resource		Loan security		Customer Management		Public Relations		Service Improvement	
Overall estimation		3,2		3,2		3,6		3,1		4,3		3,9		3,9	
Criterion	Description	Value	Answered?	Value	Answered?	Value	Answered	Value	Answered	Value	Answered?	Value	Answered?	Value	Answered?
Size		4		3,5		4		1,5		4		4		2,5	
Line Of Code	Number of lines in the text of the program's source code	0	no	0	no	0	no	45000	yes	0	no	0	no	0	no
Function Points	Number of function points	800	yes	6600	yes	2000	yes	4000	yes	1000	yes	2098	yes	32214	yes
User		1,5		1		1,5		4		2,5		3		3,5	
Service requests	Number of service accesses or service requests	9822	yes	90004	yes	11699	yes	0	no	34677	yes	9807	yes	0	no
Stakeholder	Number of roles in an application e.g. developer, supporter, project leader...	21	yes	9	yes	11	yes	5	yes	0	no	5	yes	8	yes
Structure		3		4		2,5		2,5		5		2,5		3,5	
Interfaces	Number of offered and used interfaces	14	yes	9	yes	16	yes	20	yes	0	no	15	yes	13	yes
Sebis Cyclomatic Complexity	Number of interactions in an application	0	no	0	no	0	no	0	no	0	no	0	no	0	no
Update/Improvement		5		5		5		5		5		5		5	
Business requirements	Number of recent projects to include new business requirements into application	0	no	0	no	0	no	0	no	0	no	0	no	0	no
Pflege		2,5		2,5		5		2,5		5		5		5	
Change Requests	Number of change requests	0	no	0	no	0	no	0	no	0	no	0	no	0	no
Support requests	Number of support tickets	100	yes	50	yes	0	no	80	yes	0	no	0	no	0	no

Figure 8.4: Example of a questionnaire with visualization in form of radar charts (Source: Own construction)

8.3.2 See also

The I-Pattern *Business Application Rating* describes the attributes and the relationships of an application, the characteristics of applications and the CriteriaGroups.

This M-Pattern uses the two different viewpoints to visualize the rating of applications. V-Pattern *visualization of the landscapes degree of performance* described in Section 8.5 displays the rating of business applications as a cluster map regarding to a specific characteristic. V-Pattern *visualization of the components strength and weakness* shown in Section 8.4 illustrates the rating of a single business application as a radar chart regarding to a specific characteristic.

8.4 V-Pattern: visualization of component strength and weakness

Section 8.3 shows how to determine useful criteria and rate the business application in of an application landscape. These results can be used for several analyses, which need specific visualizations. One concern is to assess the strength and weakness of a particular application regarding a specific characteristic – for example: standard conformity.

According to the information model pattern, described in Section 8.6, a characteristic is a set of CriteriaGroups. Each of these CriteriaGroups corresponds to a single rating and shows a particular strength or weakness of the component. Insights into these rating results can be given by a radar chart.

Each CriteriaGroup and its rating result can be visualized as an axis in the radar chart. An example of this is shown in Figure 8.5. The scale of an axis represents the rating-result which ought to have a range from 0 to 5, whereas 0 builds the center and 5 the outer border.

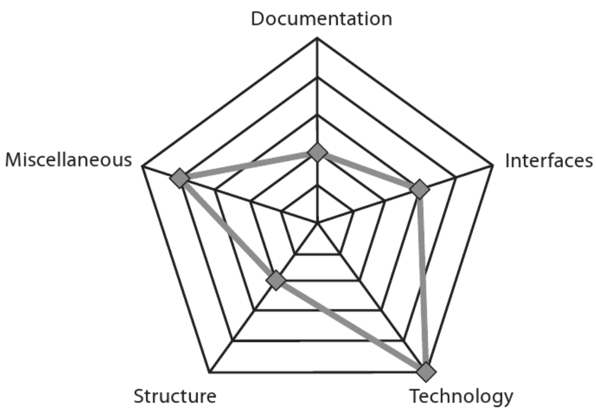

Figure 8.5: Radar chart (Source: Own construction)

8.5 V-Pattern: visualization of the landscapes degree of performance

Another concern is to compare the business applications in the application landscape with each other. The aim is to display the rating-results, regarding a particular characteristic like standard conformity, for every single business application and also to make the applications easily comparable.

The solution is an extension of the visualization pattern “Standard Conformity Layer” (V-5) [Bu08a] which is based on a cluster map and colors a business application red or green, depending on whether the particular application is standard conform or not.

The rating calculated in step four (“execution”) of the methodology shown in Section 8.3 is based on the information model pattern described in Section 8.6. This features a rating within the borders 0 and 5. To visualize this rating, different colors are assigned to different values. In order to improve easy conception of the continued information, a suggestive colorcoding ranging from red to green can be used (cf. Figure 8.6).

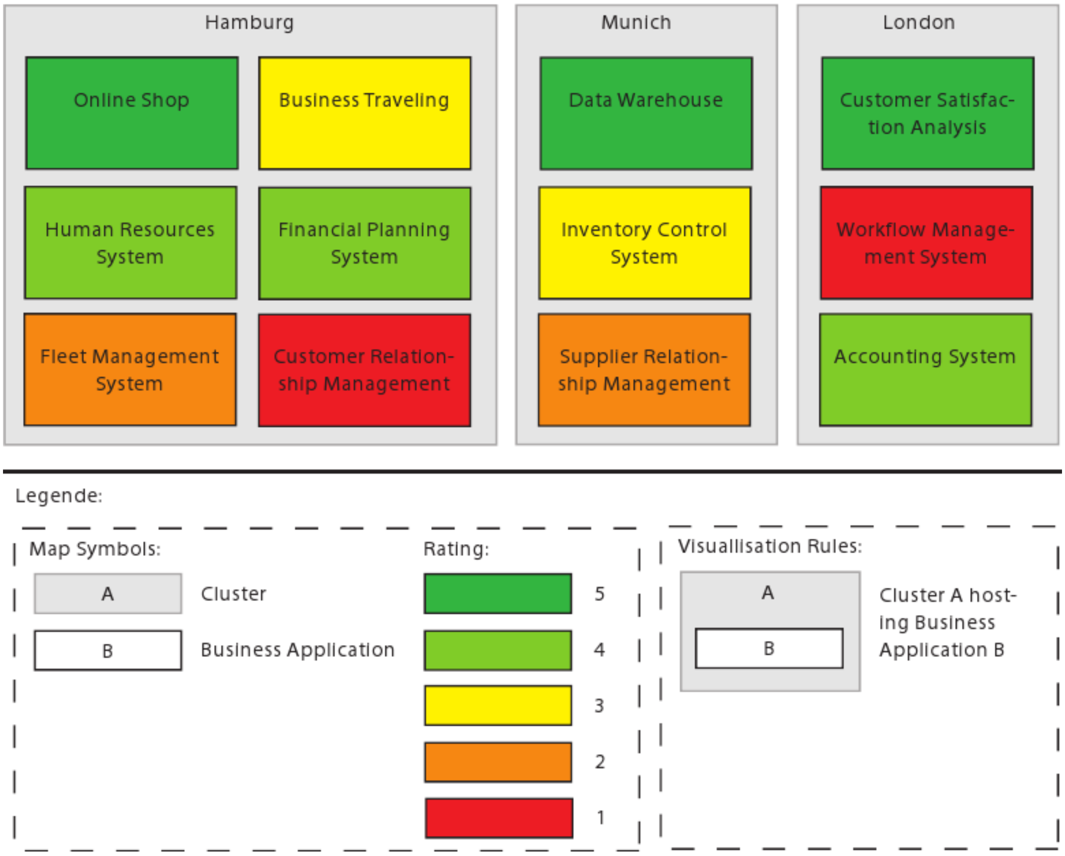

Figure 8.6: Cluster map (Source: Own construction)

This visualization makes it easy to identify the degree of performance regarding a specific characteristic for each component in a single map. Consequently, these maps help to compare the components and to identify lacks or strengths in the application landscape.

8.6 Information model for business application analyses

Section 8.3 shows a methodology to collect and use criteria to rate business application in an application landscape. The following solution shows how this information can be structured and stored using the following classes:

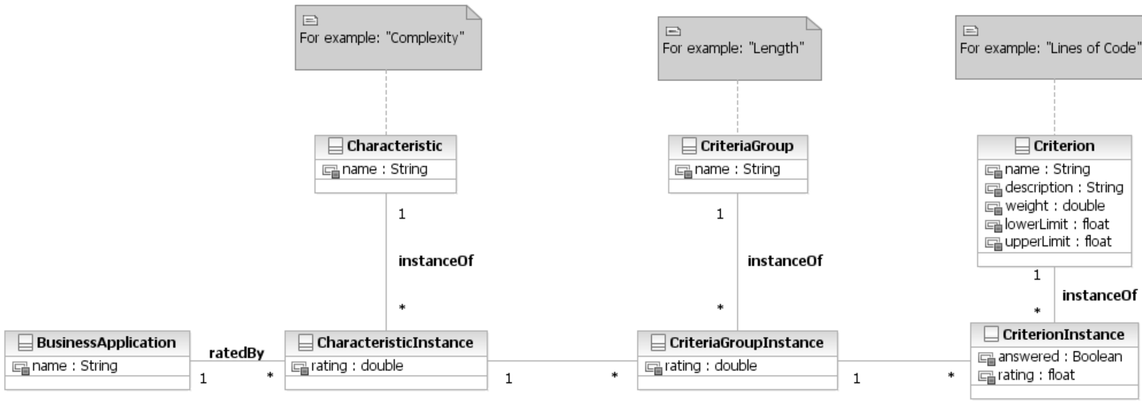

Figure 8.7: I-Pattern (Source: Own construction)

BusinessApplication: A software system, which is part of an information system within an organization is called "BusinessApplication". An information system is therein according to [Kr05] understood as a socio-technological system composed of a software system (i.e. the business application), an infrastructure, and a social component, namely the employees working with the system. An information system is further described as contributing to the business process support demanded by the organization. [Bu08a]

Characteristic: A specific perspective on a business application in order to assess it, in respect to a particular property, e.g. complexity, is called "Characteristic".

CharacteristicInstance: A concrete rating of a Characteristic for a specific BusinessApplication.

CriteriaGroup: A set of criteria, which topically belong together is called "CriteriaGroup". CriteriaGroups are used to rarefy coarse-grained "Characteristics" into finer grained groups, e.g. requirements of maintenance, which contain in return the fine-granular criteria.

CriteriaGroupInstance: A concrete value for the CriteriaGroup rating of a specific BusinessApplication.

Criterion: A measureable atomic information, e.g. the amount of change request, about a business application is called "Criterion".

CriterionInstance: This class holds the value for a particular rating of the associated Criterion for a specific BusinessApplication.

Each of these entities (except BusinessApplication) contains the value `rating`, which has to be calculated as visualized in Figure 8.8. According to the illustration, the three steps of calculation are described below.

8.6.1 Step One: Calculate the Criterion rating

Below, we again resort to interval scaled criteria, as for ordinally scaled criteria, a monotonic mapping would form a sufficient transformation. If an interval scaled criteria does not fit into the used zero to five interval, a transformation is established to adapt the indicators to the scale.

A criterion in its simple meaning is a name-value-pair. To make it rateable, the value has to be measured and a `lowerLimit` and an `upperLimit` have to be assigned in order to define the valid range

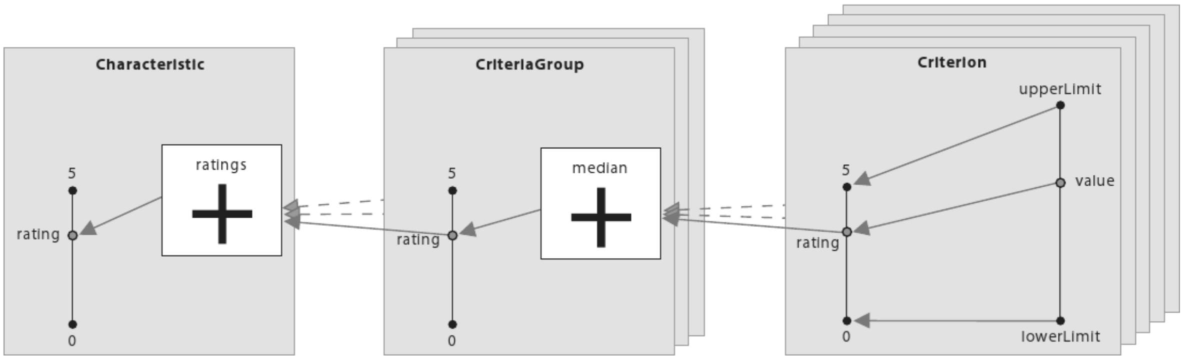

Figure 8.8: Calculation of the ratings (Source: Own construction)

for the `value`. After this the measured `value` has to be transformed into a homogeneous `rating`, which ought to be in the range between 0 and 5.

$$Criterion.rating = \frac{value - lowerlimit}{upperLimit - lowerLimit} \cdot 5 \tag{8.1}$$

8.6.2 Step Two: Calculate the CriteriaGroup rating

As shown in Figure 8.7, a CriteriaGroup contains a set of criteria. This set can be described as:

$$CriteriaGroup = \{C_1, \ldots, C_n\} \tag{8.2}$$

with $n \in \mathbb{N}_0^+$ as the amount of Criteria in this CriteriaGroup. Therefore, the rating of a CriteriaGroup is an aggregation from the individual ratings of each criteria to a common rating for the CriteriaGroup. This rating has to be calculated using the median-operation which requires a sequence of the ratings in the CriteriaGroup. This sequence has to be monotonically increasing like the following

$$r = (rating_{i_1}, \ldots, rating_i)_{i \in \mathbb{N}_0^+} \tag{8.3}$$

with $rating_j \leq rating_{j+1}$ and $i, j \in \mathbb{N}_0^+$ as the index in the Criteria. The median $\tilde{x}$ can be calculated on top of this sequence applying the following formula

$$\tilde{x} = \begin{cases} r_{\frac{n+1}{2}} & \text{if } n \text{ is odd} \\ \frac{1}{2}(r_{\frac{n}{2}} + r_{\frac{n}{2}+1}) & \text{if } n \text{ is even} \end{cases} \tag{8.4}$$

with $n \in \mathbb{N}_0^+$ as the amount of elements in sequence r.

Because the sequence r was bounded between 0 and 5 the rating $\tilde{x}$ of the CriteriaGroup is bounded as well.

8.6.3 Step Three: Calculate the Characteristics rating

Similar to the CriteriaGroup which was a set of Criteria, a Characteristic is a set of CriteriaGroups:

$$Characteristic = \{CG_1, \ldots, CG_n\} \tag{8.5}$$

with $n \in \mathbb{N}_0^+$ as the amount of CriteriaGroups in this Characteristic. Because the structure is the same, the calculation is similar as well.

As already shown in the previous step, the set of CriteriaGroups has to be transformed into a monotonically increasing sequence of ratings, which can be done using the following formula

$$r = (rating_i, \ldots, rating_i)_{i \in \mathbb{N}_0^+} \tag{8.6}$$

with $rating_j \leq rating_{j+1}$ and $i, j \in \mathbb{N}_0^+$ as the index in the CriteriaGroups. Afterwards the median can be calculated applying the following formula

$$\tilde{x} = \begin{cases} r_{\frac{n+1}{2}} & \text{if } n \text{ is odd} \\ \frac{1}{2}(r_{\frac{n}{2}} + r_{\frac{n}{2}+1}) & \text{if } n \text{ is even} \end{cases} \tag{8.7}$$

with $n \in \mathbb{N}_0^+$ as the amount of elements in sequence r.

Because the sequence r was bounded between 0 and 5 the rating $\tilde{x}$ of the Characteristic is bounded as well.

This information model pattern accommodates a possibility to structure criteria and calculate the ratings of business applications. Furthermore, it provides the possibility to analyze rating-results regarding specific concerns, as shown in Section 8.4 and 8.5.

8.7 Outlook

Summing up we have shown a method to determine the complexity and archticture conformity for a set of Business Applications. In this section, we outline further steps we deem necessary to practically implement the method.

8.7.1 Tool Integration

Successful EA management relies on (Software-)tools to create software maps and different other visualizations. Hence, the metrics developed in our approach should be integrated into existing EA management tools to visualize the findings.
We examine the integration closer for two tools. Focusing on the SoCaTool [Er06] and troux [In], we highlight possible integration paths.

8.7.1.1 SoCaTool

The SoCaTool, developed at the chair of Software Engineering for Business Information Systems of TU München, is a powerful software cartography and mapping tool. By modifying the metadata using the provided development environment, it is possible to show charts similar to Figure 8.5 directly in SoCaTool. In fact, Figure 8.6 was generated using SoCaTool.

8.7.1.2 troux

Troux 7.0, developed by troux technologies, is a set of tools for the complete EA management process. To implement our metrics in troux, we have to modify the underlying metadata-schema. This can be done either over the webinterface (*troux eplorer*) or by using the desktop application *troux archtiect*. The metadata-repository (*troux metavers*)has to be expended by the relevant parts of Section 8.6. Entering the result data of the characteristics can than be carried out using the troux explorer webinterface From the stored data a software map following Figure 8.5 can be constructed in the *troux architect*.

8.7.2 Further Ideas

To further extend our approach, a method for distributed data gathering should be implemented. This method should allow user to directly see the state of "their" business application. For example, the *troux explorer* could be used to provide data entry forms for the given characteristics.
Additionally, more and extended characteristics should be developed extending the architecture conformity and complexity criteria given in this paper.

Appendix

Criteria Examples

Some exemplary criteria for the characteristics "complexity" and "conformity" are give in the following Tables 8.1 and 8.2.

Criteria group	Criterion	Description	Literature
Size	Lines of Code	Number of lines in the text of the program´s source code	[LB06]
	Object-oriented analysis and design (OOD) metrics	Inter alia number of children of a class	[LB06]
	Development effort	Calculated in person-months	
	Function Points	Number of function points	[LB06]
User	User	Number of different users working with the application	
	Service requests	Number of service accesses or service requests	[FP00]
	Stakeholder	Number of roles in an application e.g. developer, supporter, project leader...	ITIL, ISO 20000
Structure	Interfaces	Number of offered and used interfaces	[FP00]
	Types of interfaces	Strong or weak, come-in or come-out	
	Cyclomatic complexity	Number of control flows in a module	[Mc76]
	Halstead´s metrics	Number of operations and key words in a module	[LB06]
	Sebis Cyclomatic Complexity	Number of interactions in an application	
	Data structures	e.g. number of tables in a database, number of relationship between tables	[FP00]
Update/ Improvement	Projects	Number of projects related to an application to adapt new requirements	
	Business requirements	Number of recent projects to include new business requirements into application	[MWF08]
	Project duration	Duration to adapt new requirements into application	
Maintenance	Change requests	Number of change requests	ITIL, ISO 20000
	Support requests	Number of support tickets	
	Maintenance effort	Calculated in person-months	
State of knowledge	Documentation	Size of the documentation	
	Application manager	Number of application managers	
	Guidelines for documentation and for programming	Follow the guidelines?	
Requirements	Non-functional requirements	Availability, Security, Serviceability, Reliability	DIN 66272, Volere

Table 8.1: Metrics to analyze the complexity of an application (red-labeled are the relevant metrics selected by the principal). Source: Own construction

Criteria group	Criterion	Description	Literature
	Programming guidelines		
Development	Documentation	The form of application´s documentation	
	Modeling Case Tools	Tools used for modeling.	
	API Documentation	Is there documentation for available outgoing API?	
	Modeling	Modeling complete available?	
	Supporting system	Is there a maintenance process, which meets the importance of the system?	
Components of the technology	Programming Language	Programming language used for the implementation	
	Technological platform	The underlying technique with the application	
	Exchange data formats	Data formats exchanged over interfaces.	
	Protocols of interfaces	Used protocols for communication	
Server and storage environment	Server hardware	Usable hardware	
	Server operating system	Requested operating system	
	Storage technology	Data inventory´s keeping	
Utilization	User management	Mechanism for access controls	
	Usage	Usable form of the components	
	Data access	Data access only encapsulated	
	Loose coupling	Utilization of a middleware system	
Client	Operating system at client	Technological environment by client	
	Standard conformity	Fulfillment of present web standards	
Interfaces	Protocols in use	Utilization of default protocols	
	Interfaces in use	Is it known, who uses which interfaces?	
Human Resources	Responsible person for IT	Is there a responsible person for this application	
	Responsible person for business	Is there a responsible business unit for this application	
	Responsible person for interfaces	Is there a responsible person for each interface	
	Clear definition of layers	Are functional components defined?	
		Are interfaces defined?	
		Are data only by the application recallable?	

Table 8.2: Metrics to analyze the architectural conformity of an application (red-labeled are the relevant metrics selected by the principal). Source: Own construction

Bibliography

[Bu07] Buckl, S. et al.:. *Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008)*. Technical report. Technische Universität München, Chair for Informatics 19 (sebis). 2007.

[Bu08a] Buckl, S. et al.:. *Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008)*. Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2008.

[Bu08b] Buckl, S. et al.:. *Towards simulation-supported enterprise architecture management*. In *Fachtagung Modellierung betrieblicher Informationssysteme (MobIS2008)*. 2008.

[Bu09] Buckl, S. et al.:. *State of the Art in Enterprise Architecture Management*. Technical report. Technische Universität München, Chair for Informatics 19 (sebis). 2009.

[Ch09] Chair of Informatics 19 Technische Universität München:. *Lecture: Software Engineering for Business Applications - Master Course*. online. 2009.

[En09] Engels, G. et al.:. *Design for Future – Legacy-Probleme von morgen vermeidbar? Informatik-Spektrum*. 32(4). 2009.

[Er06] Ernst, A.; Lankes, J. S. C. W. A.:. *Using Model Transformation for Generating Visualizations from Repository Contents - An Application to Software Cartography*. Technical report. Technische Universität München, Institut für Informatik, Lehrstuhl für Informatik 19. 2006.

[FP00] Fenton, N. E.; Pfleeger, S. L.:. *Software Metrics. A Rigorous and Practical Approach.: A Rigorous Approach*. Thomson Learning Verlag. 2000.

[HW08] Hafner, M.; Winter, R.:. *Processes for Enterprise Application Architecture Management*. In *Proceedings of the Proceedings of the 41st Annual Hawaii International Conference on System Sciences*. 2008.

[In] Inc, T. T.:. `http://troux.com/company/news/pressrelease.asp?pr=080319_v71_pr.xml&parchive=2008`.

[Kr05] Krcmar, H.:. *Informationsmanagement*. Springer Verlag. Berlin. 2005.

[LB06] Laird, L. M.; Brennan, M. C.:. *Software Measurement and Estimation: A Practical Approach (Quantitative Software Engineering Series)*. Wiley-IEEE Computer Society Pr. 2006.

[LS07] Lankes, J.; Schweda, C. M.:. *Constructing Application Landscape Metrics: Why & How*. Technical Report TB0701. Technische Universität München, Chair for Informatics 19 (sebis). 2007.

[LTC02] Lindvall, M.; Tesoriero, R.; Costa, P.:. *Avoiding architectural degeneration: An evaluation process for software architecture*. In *International Symposium on Software Metrics (Metrics'02)*. pages 77–86. 2002.

[Ma08] Matthes, F.:. *Softwarekartographie. Informatik-Spektrum*. 31(6):527–536. 2008.

[Mc76] McCabe:. *A Complexity Measure. IEEE Transactions on Software Engineering*. 2:308–320. 1976.

[MWF08] Murer, S.; Worms, C.; Furrer, F. J.:. *Managed Evolution. Informatik-Spektrum*. 31(6):537–547. 2008.

CHAPTER 9

Analysis of a risk management system with its components and interfaces at a financial service provider

Maximilian Graß (max.grass@gmx.de),
Michael Weißberger (weissber@in.tum.de),
Ralf Wernicke (mail@rwerni.de),
and Gregor Wylezich (wylezichgregor@web.de)

Abstract:
Focus of this report is the risk management system (RMS) of a global acting financial service provider (FSP). There are different RMS installations for each country and region. Additionally, the RMS communicates with other external systems like credit bureaus. Therefore, a correct overview depicting the interactions with other systems of each installation is needed to provide a basic understanding of the RMS in each country. In order to develop a standardized message format which can be used by the different RMS installations, the messages sent between the RMS and its components are analyzed in respect to conformity.

9.1 Motivation

As the report has its main focus on the RMS it is crucial to understand its basic tasks and setup. Based on the current problems with the RMS, two principal project goals are identified. Each concern derived from the project goal is presented in detail in the following sections.

Key motivation is that gaining fundamental knowledge about the different RMS installations and updating their documentation would not only reduce costs and complexity, but in addition this knowledge could be also precious when developing a new RMS.

9.1.1 RMS overview

The RMS checks proposals from customers for its risk, before a contract starts. In order to do this, it communicates with various systems[1]. Figure 9.1 sketches the overall context of the RMS.

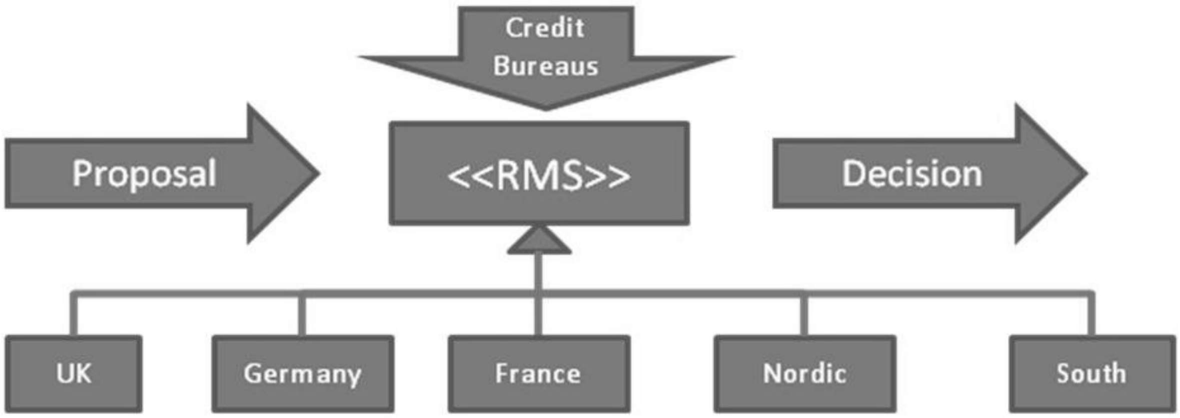

Figure 9.1: Overview of the existing risk management systems

Due to different laws in each country and regional bound systems[2], the RMS varies for each region. Consequently, this results in a high complexity and a low transparency of what systems participate in each RMS installation and how all the systems communicate with each other. Furthermore, it is unclear what information is exchanged by each system.

9.1.2 Project goals

As a first result of this problem, it was necessary to provide an overview of the RMS in order to familiarize quickly with the similarities and differences of each RMS installation. Thus, an appropriate way of presenting this information in a compact but informative way was needed, tailored to the customer's needs. Additionally, the working process for each RMS installation had to be analyzed. The goal here was not only to have a correct documentation of the processes but also to have a customized presentation in order to support an easy comparison of processes. As an additional result, an "ideal"'[3] process was designed.

The second project goal was to analyze the messages sent between the systems in respect to conformity. The messages are XML-files with different fields for each RMS installation. It become obvious, that a standardized message type for all RMS would reduce complexity and documentation overhead. Thus, instead of having many different messages there would be only one message. Moreover, this knowledge could be relevant when implementing a new RMS. As a side effect, the present documentation was checked for correctness and unnecessary[4] fields were discovered for each RMS installation.

[1] E.g. the front end where proposals are made and credit bureaus for customer liquidity information
[2] Regional front-ends and credit bureaus
[3] Ideal in this case represents the way the customer thinks the process best fits into the incumbent business process.
[4] Fields that are always left blank as they are remaining fragments of older RMS versions

9.1.3 Structure of this document

In the following, one goal is directly linked to a specific project problem. Subsequent description starts with an example and the context where the problem occurred. A specific solution to solve the problem is presented in the problem section. Graphs and other visualizations applied in the project context are explained in the section viewpoint. Each description ends with a general outcome section.

9.2 RMS components and interaction process

9.2.1 Example

The analyzed RMS runs on four instances - one for each country namely Germany, United Kingdom, France, and the Nordic region. They differ but possess a functional core which is the same in every system.

9.2.2 Context

Each RMS instance for each country runs on the same location and is administered by one single authority. This facilitates the data collection and analysis.

9.2.3 Problem

You are experiencing an independent versioning of systems that provide your business with the same core-functionality. Therefore, you need a correct overview which shows the interactions with other systems of each instance that is also capable of providing a basic understanding of how the RMS works for each country. You want to familiarize quickly with the different systems but you also need to understand the processes of each instance in detail. For that reason, on the one hand, you require an abstract high level landscape overview, representing the interactions and interfaces among the systems, and on the other hand, you demand a detailed process flow for each RMS. Moreover, the process flows should be modeled in a way to make them comparable. The presentation of this information has to be tailored tightly to your needs.

9.2.4 Solution

This project contains two different, but coherent tasks. For that reason, it is split into two sub-tasks.

The first one has the focus on the work flow and the interactions within the context of the RMS of a single country. It has to consider the RMS for all countries or regions. Therefore, this solution provides a consistent depiction of the work flow of each of RMS.

The second has its focus on the design of an overview of the RMS for all countries including the systems involved and the interfaces among them. Hence, both subtasks need to be kept consistent.

If an additional interface is detected during design of the work flow, it has to be added to the landscape overview, in the case systems involved are not connected yet. Thus, the landscape overview is a more abstract representation than the work flow. Consequently, the overview of the landscape only contains one single interface between systems, whereas the work flow may contain multiple data exchanges among these systems. Figure 9.2 depicts the iterative approach for this solution. The result thereby is always a process chart.

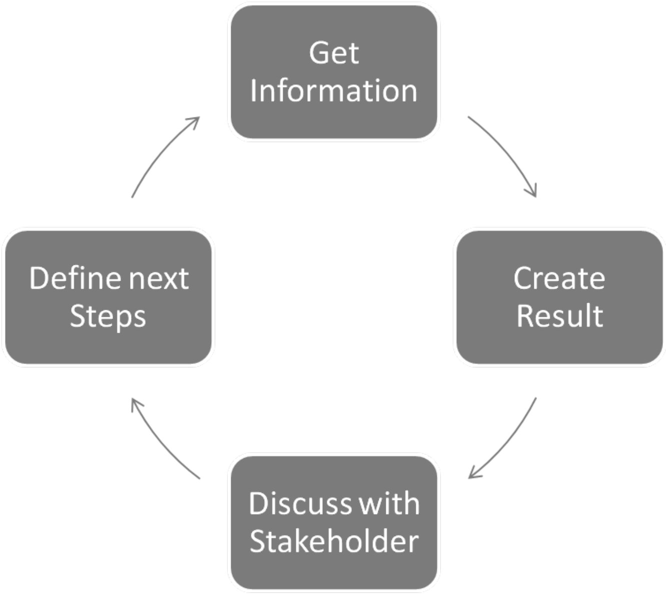

Figure 9.2: Iterative steps of the solution process

9.2.4.1 Create an Overview of the Interaction Process

As a first step, the RMS of Germany and UK are analyzed, since these systems are the most relevant ones for FSP. Additionally, for these systems more detailed information is available.

In doing so, the available documentations of the systems are studied. The documentation of the RMS UK contains a flow chart of the process. The structure of this chart is used as a template for the design of all other RMS in order to find similarities and differences among the systems.

In the next step, the draft of the existing flow chart is adapted to the RMS of UK. As an important feature, the communication between the systems involved is added. Therefore, FSP uses standardized messages, as described in section 9.4.5. The result is shown in figure 9.4. In the next step, we assign the template to the RMS of Germany. In addition, the communication among the different systems involved is taken into consideration. After ensuring the visualization is meeting the stakeholders' demands, the templates are applied to the remaining RMS.

During study of the documentations it turns out that for the RMS of France, not enough information is provided to depict the work flow. Consequently, the RMS of France is excluded from the scope of the project upon approval by the stakeholders. In a next step, we harmonize the structure of the different flow charts in order to make them comparable. The idea is to find a visualization that allows the process maps to be put on top of each other.

At first, involved subsystems and the particular process steps supported by theses subsystems are identified. In the context of each RMS, these are the following systems with their main functions:

- **Proposal System** (to initiate a new contract and customer interface)
- **Contract Management System (CMS)** (to administrate contracts)
- **Credit Bureau** (to obtain a credit information about a customer)
- **Risk Management System** (to calculate whether a contract should be closed)

Each kind of system is color-coded and arranged identically on each process map. Additionally, the same method is applied to the process steps by aggregating detailed process steps to bundles and then identify bundles that perform the same task in each RMS. Afterwards, equal process steps among the different RMS are identified and color-coded as well.

Therefore, it is necessary to differentiate between process steps, which are

- unique,
- equal within all three RMS and allocated within the same subsystem,
- equal within two of three subsystems, and
- equal within two of three systems and allocated within different subsystems.

In addition, we also distinguish between bidirectional and unidirectional communication flow between systems by using different types of arrows. The visualization as seen in figure 9.5 meets the final demands of our stakeholders.

9.2.5 Modeling the high level overview

Similar to the approach for modeling the process flows, the documentation of FSP are studied, in order to develop the high level overview of the RMSs. Furthermore, there exists a current version of the FSP system overview (cf. figure 9.3).

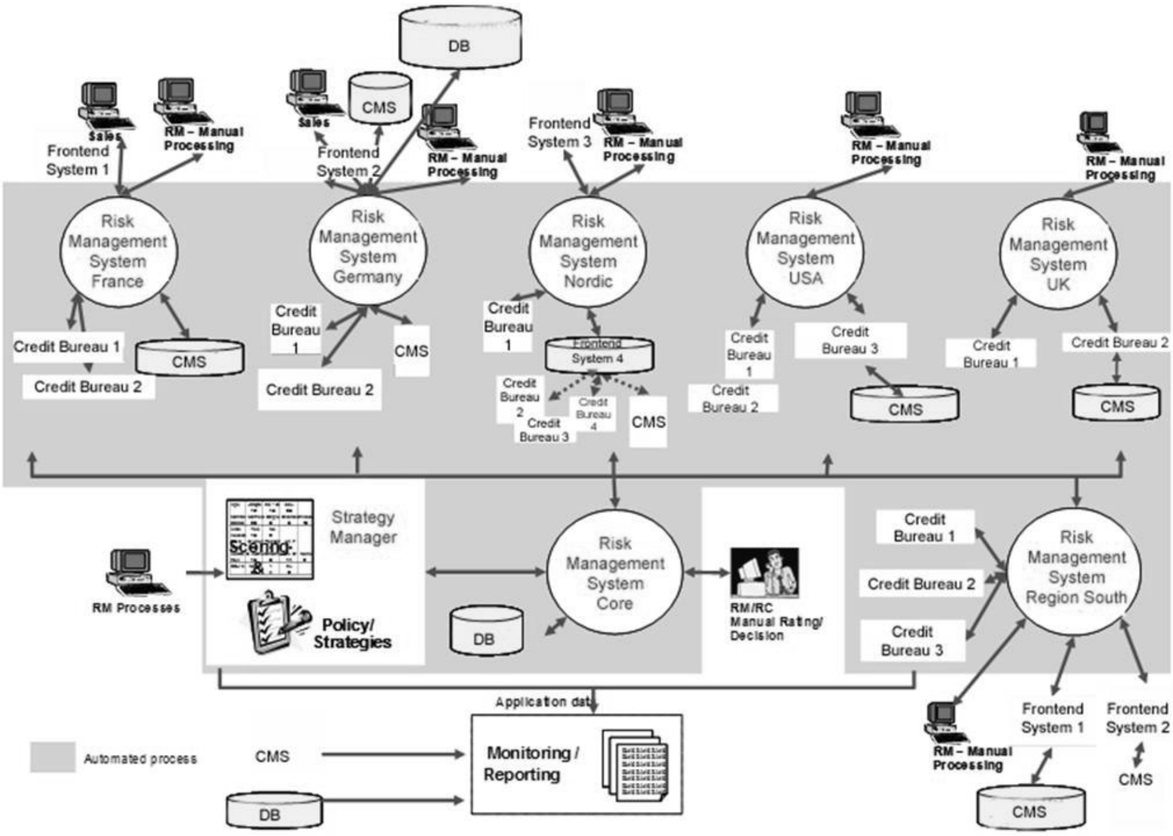

Figure 9.3: Existing overview on the RMS landscape

Using this information, a new system overview is developed, following the iterative process as described in figure 9.2. Therefore, we firstly determine the types of systems involved in the risk management process. As a result, the four sub systems as described above are detected. As mentioned before, this project focuses on four different regions (Germany, United Kingdom, France, Nordic region), but in our overview a fifth region, namely South, is added.

Given these initial parameters, we model the overview as a cartesian map, arranging the countries on the x-axis and placing the subsystems on the y-axis. With this approach it is very easy to compare the different implementations. Figure 9.6 shows the first approach - the systems involved are depicted, including their nature and the interfaces between them.

The developed overview is subject to discussion and validation by means of several meetings with the stakeholders. As a result the subsystems are re-arranged, figure 9.7 shows the final result.

9.2.6 Outcome

The results of our analysis of the as-is state are the uniform representation of the process flows of the different systems and a new system overview making the risk management process among the countries comparable. The analysis results in a recommendation, how to implement one central system, reducing complexity, and interface usage. In summary, the new system overview is much more intelligible than the old one, presenting the as-is state in a better way. Furthermore, the risk management process can be compared in an easier way among the analyzed regions.

Additionally, a possible improved process flow is modeled. It integrates the process steps that are supported by the analyzed systems. As mentioned before, we only model the process flow of three RMS, due to lack of information.

	Germany	**United Kingdom**	**Region Nordic**
Germany	-	17/24 (71%)	14/24 (58%)
United Kingdom	17/18 (94%)	-	12/18 (67%)
Region Nordic	...	...	-

Table 9.1: Comparison of the process steps

Within table 9.1, we show the number of process steps which are equal to the RMSs. This number is written before the slash, the total number of process steps is written after the slash. The RMS on the y-axis is compared to the RMS on the x-axis. So, for example, the RMS of Germany has 17 equal process steps with the system of the United Kingdom and a total number of 24 process steps, which yields a percentage value of 71 percent. For all three RMSs we had 12 process steps which were equal among all systems. The system of United Kingdom has nearly the same process steps as Germany (94%).

The analysis reveals, that the system implementation used in Germany is the most useful concept except for some redundant interfaces. On the one hand, the implementation of the system in UK can be seen as a subset of the implementation in Germany, because nearly all fields (17 of 18) of the UK system are included in the system of Germany. Moreover, the system of UK and region Nordic share only 12 processes. Furthermore, the current implementation of the system in Germany follows the concept of one central interface best. For that reason, we used this process (Germany) to model a possible "ideal" risk management implementation. Figure 9.8 depicts the resulting process flow.

9.2.7 Consequences

Our approach as described above requires several reviews with stakeholders and domain experts. Without regular meetings and feedback it is not possible to achieve this result.

9.3 Viewpoints

This chapter shows and describes the process work flow, the process overview and the "ideal" process that derived from the analysis.

Process maps

The graphs 9.4 and 9.5 show the first approach and the final version of the process work flow.

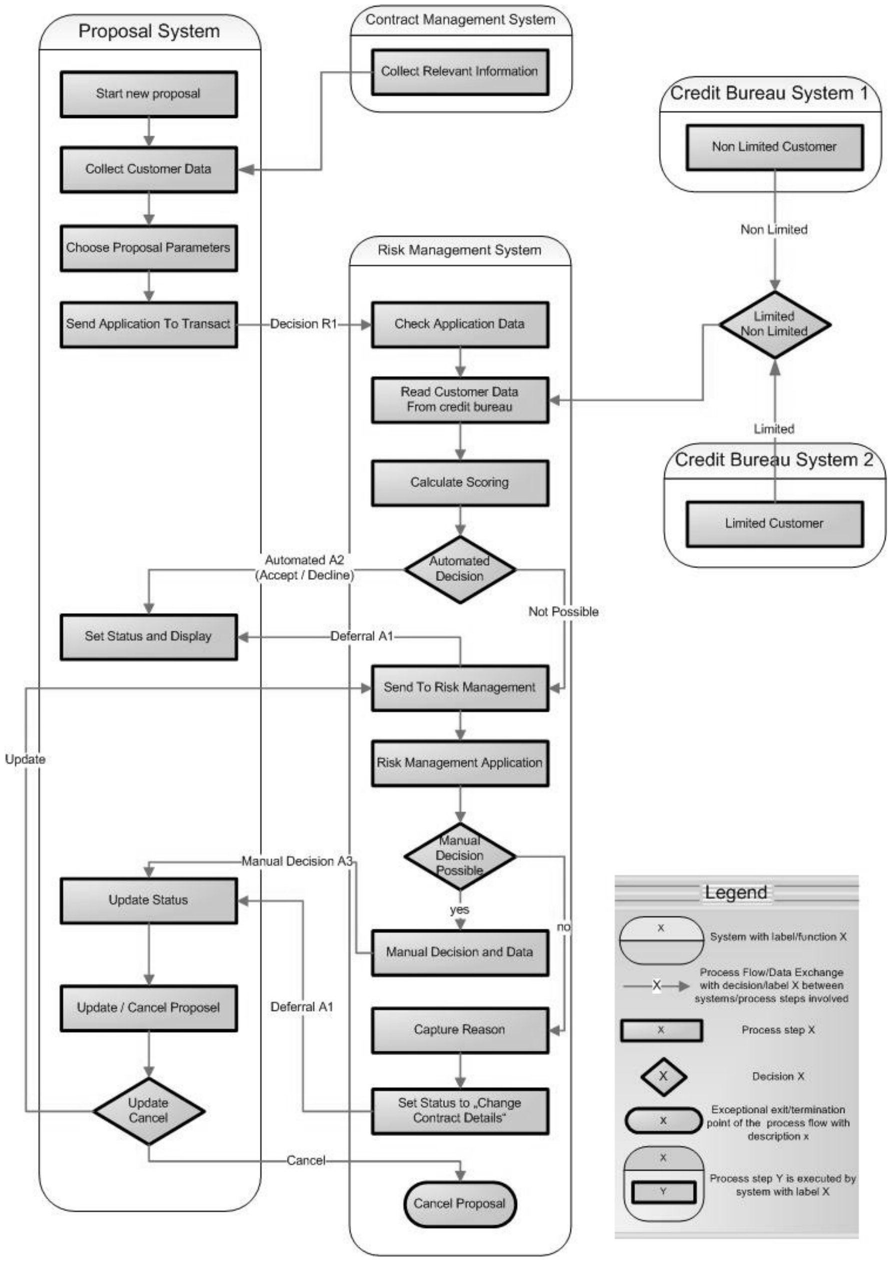

Figure 9.4: First draft of the depiction of the workflow

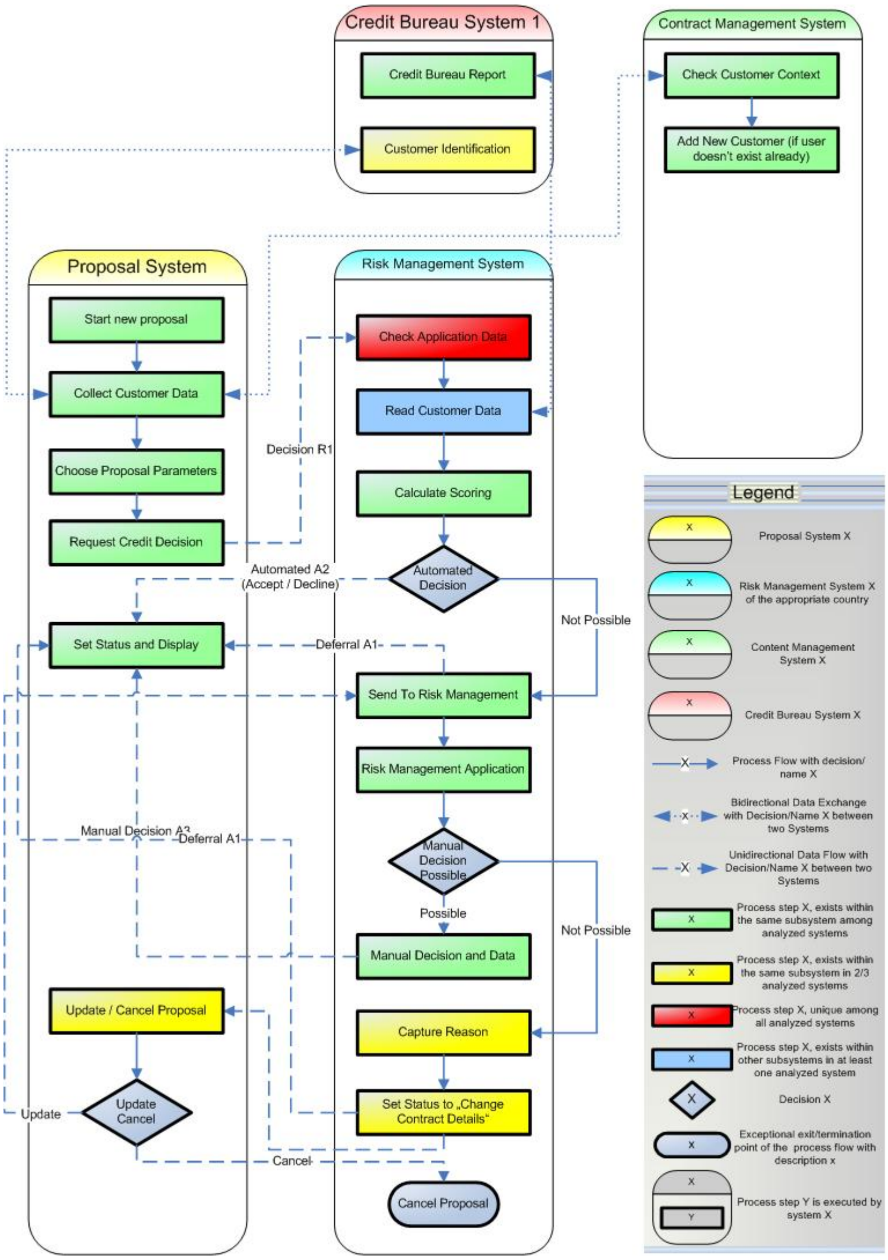

Figure 9.5: Final Overview of a Workflow

Process overview

Interfaces among two subsystems are displayed with arrows. Thereby, data flows are always presented in a vertical direction. The new design makes it easy to determine all credit bureaus at one stroke, because they are all arranged in one row called "Credit Bureau". In addition, credit bureaus of a specific country can be determined by considering in which column the credit bureaus are arranged. For example, the intersection of second row and third column shows the credit bureaus of the Nordic region.

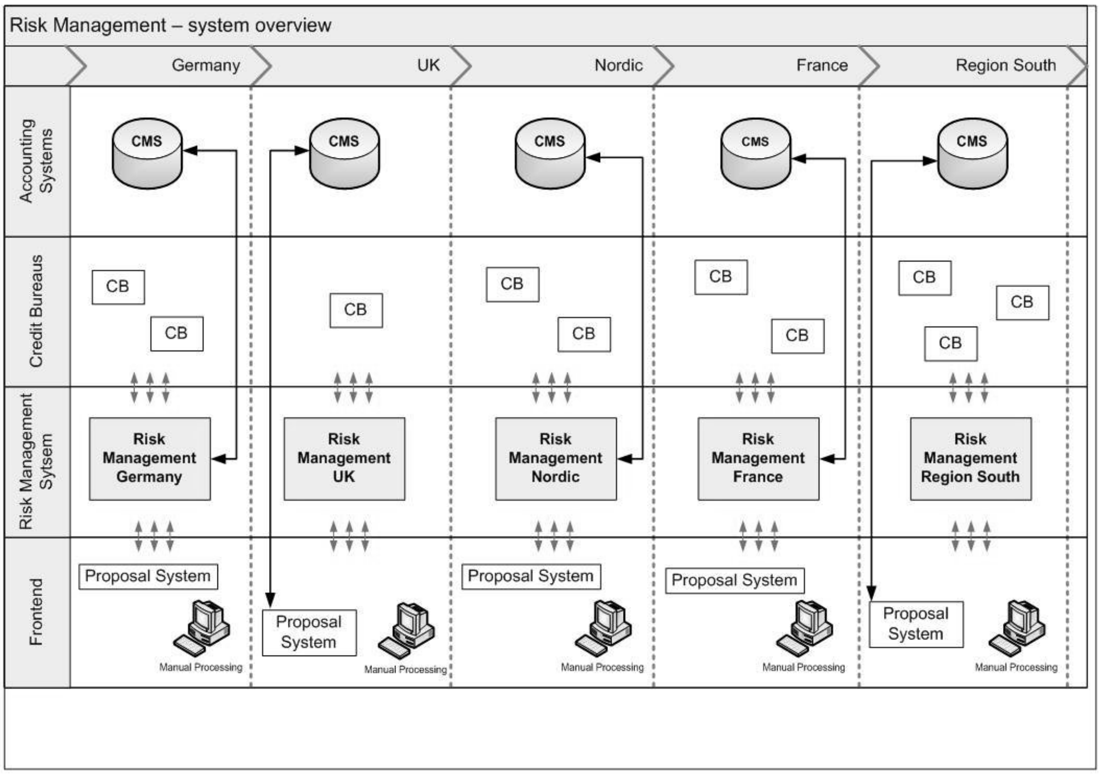

Figure 9.6: First Draft of the Overview of the Landscape

In the final version, the internal systems are colored in green, external systems are colored in red. We use the same coloring for the internal and external interfaces. Moreover, the same symbols for the same sub systems are chosen, e.g. credit bureaus are always displayed as a hexagon. As a consequence, the y-axis could be used for additional information, as the symbols already depict the type of a subsystem. Nevertheless, to label the y-axis with the subsystems is beyond the scope of this project.

Ideal process

The number of interfaces is reduced in order to get a minimal number of interfaces among all subsystems. It is obvious that the "ideal" process flow contains all the process steps that are similar among all implementations. Furthermore, the RMS acts as a central interface to all other subsystems. Thus, only one interface has to be modeled, to make communication possible among all other subsystems.

9.4 RMS message analysis

9.4.1 Example

As described so far, the RMS has several interfaces to other systems. For the data exchange between those applications XML messages are used. These messages are specified for each country and therefore may contain different information due to the different requirements of each country.

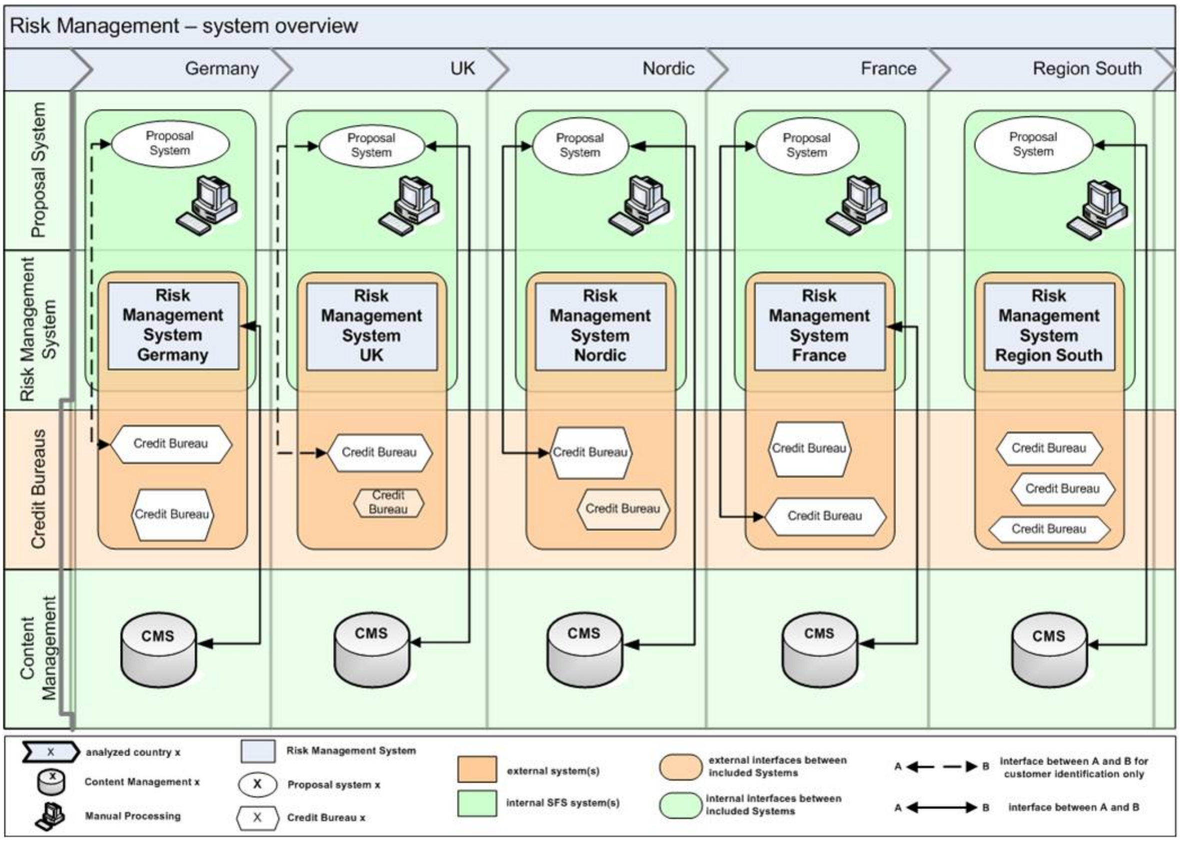

Figure 9.7: Final Overview on the landscape

9.4.2 Context

In contrary to the structure, the content of the XML messages is confidential as it contains sensitive customer data. Consequently, no content analysis is possible for these messages.

9.4.3 Problem

You feel that standardized messages would help to make your RMS simpler in respect to complexity and would provide fundamental knowledge for implementing a new RMS as well. Additionally, your present documentation might be outdated and should be checked for correctness. You want to identify message fields that are no longer used by any of the subsystems and should be removed therefore.

9.4.4 Solution

Figure 9.9 describes the three steps of the message analysis. Additionally, the iterative process as described in section 9.2.4.1 was respected: In each cycle as much information as possible is extracted and the result presented to the stakeholders. Afterward, the procedure for the next iteration is set up in accordance with the project partner.

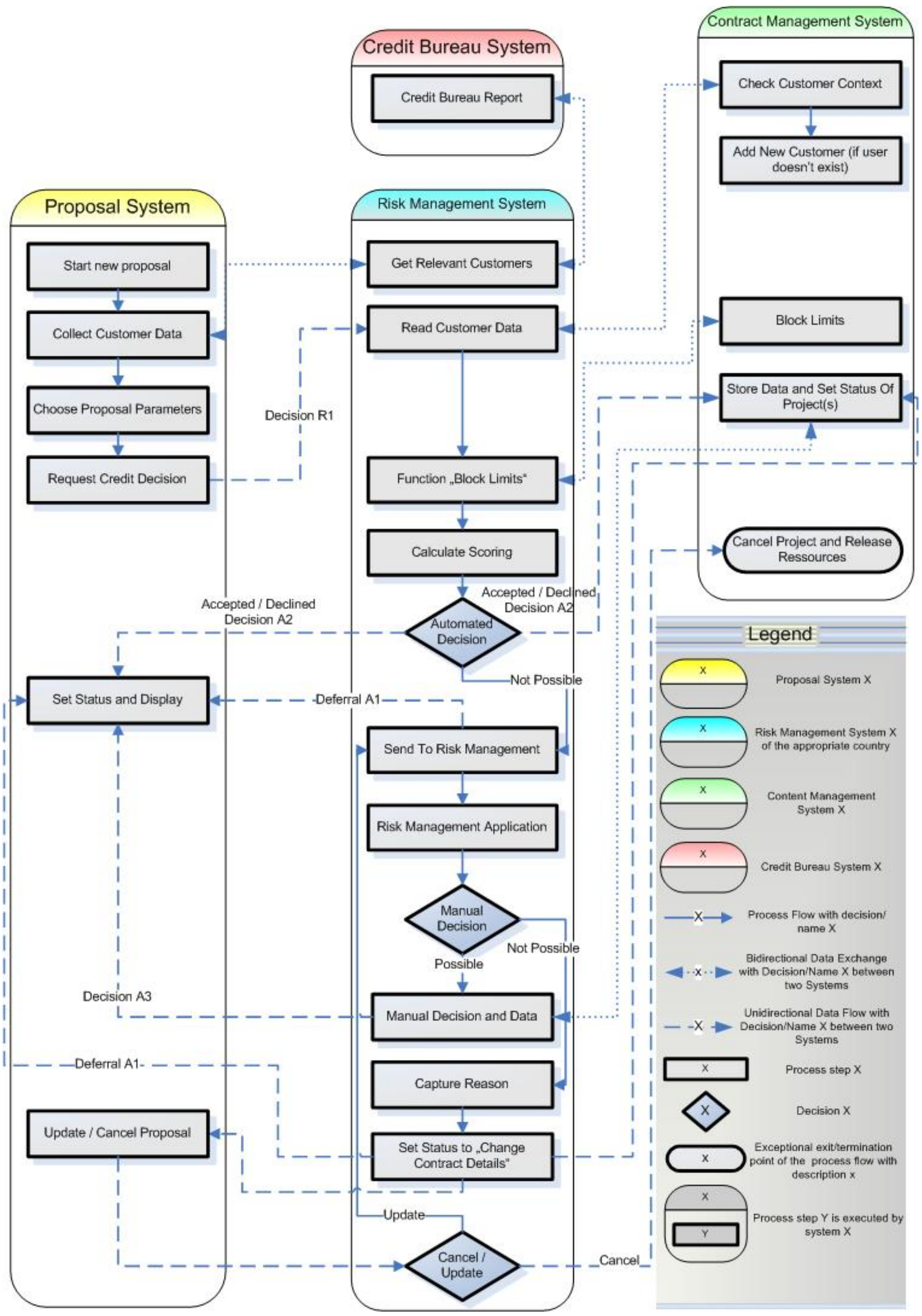

Figure 9.8: Possible 'ideal' process flow

9.4.5 Study of existing documentation

As a start, textual documentation files and spreadsheets containing interface descriptions are provided for all RMS installations in the different countries. There is a high number of different messages that are sent between the systems. Each information exchange between two systems starts with a request message named 'R1'. There are three possible messages that can be sent as a response. They differ

Figure 9.9: Steps of the solution

in their content: A1, A2 and A3. We concentrate on the request message, justified by the following reasons:

- The R1 is the initial message sent by all RMSs.
- The R1 request contains most fields, so it is the largest XML message; a profound analysis here has the most benefit for the customer.

The attempt is to create an approach for the R1 message and to transfer the procedure to all remaining messages. To do this, the first step is to analyze the R1 request. As a first approach, an excel sheet with a cross classified table as shown in table 9.2 is compiled.

The table contains an artificial field identifier (FieldID). Every field name with equal meaning gets the same ID. The column "Message" describes the message in which the field occurs, e.g. R1. Because of the varying systems used in the countries, fields are named differently. Therefore, a mapping has to be done from the local field name used in the respective RMS. For instance, 'Kundennummer' and 'Customer_ID' both identify a customer. This information can be found in the columns labeled with Field names. The last columns describe the different systems. An 'X' marks the system in which the message field occurs. With such a table it is possible to obtain an overview of all used XML message fields in respect to the different countries. However, due to the high number of different fields in the XML messages, this is too complex and time consuming as a first approach unless the amount of message fields to be analyzed is small enough[5].

		Field Names		**Systems**		
Field ID	**Message**	**Field Name: RMS**	**Field Name: Regional System**	**Germany**	**UK**	**France**
1	R1	Customer_ID	Kundennummer	x	x	
1	R1	Customer_ID	Customer_No			x
2	R1	Customer_direction	Kundenadresse	x	x	x
⋮	⋮	⋮	⋮	⋮	⋮	⋮

x:= Message field occurs in that system

Table 9.2: Cross classified table of XML message fields

9.4.6 Transfer into database

In order to have a quick overview on how each message differ in size and field names, the next step consists of loading the interface description spreadsheets into a database system. Thus, it is possible to use the provided database functionality, like joins, to examine the files. The spreadsheets are provided in the Microsoft Excel format. Due to the comfortable import functionality of Microsoft Access for Excel files, this database system is used.

[5]In this analysis, the amount of message fields lies between 37 and 418 for each RMS

The result of the import is a database table for each R1 request in every country. As each system is to be compared with every other system, 10 natural joins on the field names were performed in total, for five different countries. This way, we are able to find out the amount of identically named field which also occurs in the joined table. For instance, if we join Germany with UK, this results 37 hits. This means that 37 fields of the message used in the German system also occur in the UK system. Comparing messages with themselves result in the total number of fields in every country. An overview of the result is given in table 9.3.

	Germany	UK	South	France	Nordic
Germany	94	37	37	14	41
UK	37	331	312	9	159
South	37	312	418	9	159
France	14	9	9	57	14
Nordic	41	159	159	14	191

Blue numbers indicate the total number of message fields in the specific country

Table 9.3: Result of the join operations

It is obvious that table 9.3 does not provide a final result but it is a good way to get a first understanding of the differences in the messages. At a first look, the largest R1 request is made by the system South with a total of 418 fields, followed by the UK system with a field count of 331. Nearly all fields from UK are also included in the South System

This becomes even clearer, if we use a table where we express the intersection with a percentage value like depicted in table 9.4. It shows that, e.g., for the Nordic System, 87% of the fields are already included in the UK-System.

	Germany	UK	South	France	Nordic
Germany	100%	39%	39%	15%	44%
UK	11%	100%	94%	3%	48%
South	9%	75%	100%	2%	38%
France	25%	16%	16%	100%	25%
Nordic	21%	83%	83%	7%	100%

Table 9.4: Result of the join operations - intersection percentage

The next step is to identify the fields that are used by all systems. Therefore a natural join on all tables is performed returning the intersection of all messages. The result is listed in table 9.5. In this case, the result leads to the conclusion that the employed interfaces in each country differ significantly as it is impossible for the RMS to make a request with only six fields. The fact that in each country the R1 request works and that theoretically always the same information is needed for the risk management process draws the conclusion that the naming of the fields must be different. Therefore, we need to create a translation table that provides a matching between field names that are named differently but convey the same information.

To get a general impression, if this intention can be satisfied, the first look is on the two most homogeneous systems UK and South. However many field names like CU_Partner_Add_Line1_1 and CU_Partner_Add_Line3_2 occur. The question is, whether these fields are really still used, or just out of date and not used anymore. In this vein, a look at XML messages which are sent in produc-

Field Name
CU_Customer_Name
CU_Country_Code
TR_Sales_Person
TR_Requested_Amount
TR_Term_of_Contract
TR_Product_Type

Table 9.5: Field names which occur in each RMS instance

tive environment between the RMS and the respective front end system is needed. They are called productive XML messages in the following.

9.4.7 Analyzing of productive XML messages

The ideal situation for this process is to be provided with XML messages from different countries that have the same content. This means that the risk management process in each country is running with the same customer, assets and conditions. However, due to the specific characteristics and employment of this RMS, this approach is not applicable. Consequently only ordinary productive XML messages can be used.

Five different messages are provided: Two for the RMS in Germany, one for the systems in UK, France and Nordic. Unfortunately, no "real" messages can be provided as they contain sensitive customer data, but messages from a test system is used instead, containing fictive data. Also no messages for system South is available. Consequently this system is not analyzed.

A Java program is developed which is capable of analyzing XML messages in terms of that it can distinguish between used and unused fields. Furthermore it counts the number of message fields. The key advantage of this method is that the program can be used internally by FSP; the output of the program does not contain sensitive data anymore and can be used to draw conclusions. The output of the program is shown in table 9.6.

XML Message	Total Number of Fields	Number of Used Fields	Specified Number of Fields in the Documentation(cf. Table 9.3)
R1-Germany (1)	53	24	94
R1-Germany (2)	53	24	94
R1-UK	476	68	331
R1-France	103	41	57
R1-Nordic	448	3	191

Table 9.6: Output of the Java program for productive XML messages

As a first result we conclude: none of the productive XML message has the field count specified in its documentation. For the systems of Germany, UK, and France the related documentation is out of date. As for the Nordic system, it is supposed that the delivered XML message is not productive because only three fields are filled out.

To draw a conclusion about the used fields in each message a sample size of one instance is not very sufficient. For this reason the program is modified to be capable of handling multiple XML messages.

The output is very similar. In addition to the number of how often the field name appears in the messages, it is shown how often this field is filled out.

Thus, it is possible to decide with a higher reliability if a message field is still needed or not. For instance, in table 9.7 we can see the beginning of the result of the program for ten messages of the same type. The field 'Customer_name' appears in every message, and is always carrying data. However, the field 'Legal_form' is only filled out eight times. Consequently, this field is not mandatory. The field named 'CU_Cust_add_2' exists in every XML message but is never used. In that case, these attributes can be considered to be out of date and are not longer necessary.

XML Field Name	**Number of XML Messages with Spec Field name**	**Times Filled**	**% Used**
Customer_name	10	10	100
Legal_form	10	8	80
CU_Cust_add_2	10	0	0
⋮	⋮	⋮	⋮

Table 9.7: Output of the Java program for multiple XML messages

However, the evaluation of ten messages from the Nordic systems shows the same result as the first message analysis: Only three values are set. In agreement with the stakeholder it is decided that these messages are regarded not to be productive and that a further evaluation of the Nordic system does not make any sense.

For each of the remaining systems, we are able to describe a new message with minimal size. The new message has to fulfill the same tasks as the old one and therefore all fields which are used at least once are included.

XML Message	**Total Number of Fields**	**Number of Fields Always Used**	**Range of Use]0;always[**	**Always Empty**
R1-Germany	53	24	6	23
R1-UK	483	67	23	393
R1-France	103	103	103	0

Table 9.8: Output of the Java program for all messages in each country

This result leads to the following conclusions:

- The previous given XML file from UK is already out of date; the productive message contains seven more fields.
- All fields specified in the system of France are needed.
- UK holds far more fields than necessary.

In order to reach the final goal of having one standardized message for all systems, all given XML R1 messages for UK, France and Germany are analyzed by the program. The result can be found in table 9.8.

9.4.8 Outcome

One standardized XML message is provided for the R1 request, aggregating information from RMS in Germany, UK, and France. However, due to a lack of information, this is not possible for system

South and Nordic. The documentation and even the test systems are out-dated. Additionally, FSP has a tool embodying the method of creating standardized messages for any amount of messages given. That way, FSP can choose which systems shall be analyzed and is even able to repeat this when the messages would become out-dated in the future.

9.4.9 Lessons Learned

The following lessons learned are derived from the message analysis:

Even with an out-of date documentation, a quick look on these documents as a first impression is good. However, the analysis and conclusions drawn should always be based on productive - and therefore up-to-date - data. If this is not possible due to company regulations e.g. sensitive customer data, the analysis can only be made by the customer. The task then is to provide the customer with a detailed instruction plan of what steps have to be undertaken. This may result in a step-by-step process document and/or an application which is executed by the customer directly.

Also when the approach is unclear due to old or missing information, a close collaboration with the customer in the form of an iterative process is needed to ensure that the outcome and the steps to be taken always beneficial for the customer. This is because each iteration may bring new information which could push the project goal into another direction. If a project goal cannot be directly met due to time constraints, at least a course of action and a detailed documentation should be provided to the customer, enabling him to continue with the analysis.

9.5 Conclusion and Prospect

Parallelizing the work on the project goals turned out to be crucial to get the challenging work done in a short period of time. In this vein, we were able to work simultaneously on the two project goals as described in the section 9.1. The realization of the two main goals was supported by an intensive and regular communication with all stakeholders. Nevertheless, there are further steps which should be realized in the future.

As far as the project goal "process modeling" is concerned, the next step should be a detailed validation of the designed process maps through the stakeholders. A well-advised set of process steps is an important factor for a useful new process that reduces complexity and the number of interfaces. The "ideal" process flow, we modeled might be a good starting point for this task. Furthermore, the new system overview is a good basis to outline the communication among the systems that are involved in the current risk management process. Due to the new design, it might be meaningful to add additional information in the overview map.

The next step related to the goal "message analysis" would be to complete the global message, which we created for the R1 request by including all available systems. As seen above, this task can be supported by the Java program we developed. In addition to the R1 message, a global message should be created for all other message types which are used for communication between the different systems. The procedure will result in a message, which may contain duplicated fields, which have to be eliminated. Duplicate fields are caused by the problem that only field names can be compared in a safe and automatic way, but not their content. Thus, it is possible that fields appear several times with different names while containing the same information. The outcome of this analysis could be a mapping table as shown in Table 9.9.

Name Germany	**Name UK**	**Name France**	**New Name**
CU_BP_No	CU_BP	-	CU_BP_NO
-	-	CU_Acteur_Code	CU_Acteur_Code
BR_Ovd_Debt120	-	-	BR_Ovd_Debt120
⋮	⋮	⋮	⋮

Table 9.9: Mapping table for the field names

Appendix

	Exists in RMS			
Field Name	**DE**	**UK**	**FR**	**% filled out**
AP_APPLICATION_NUM		X		100,00%
AP_APPLICATION_SOURCE	X			100,00%
BR_Ovd_Debt120d_Cur_Dr	X			100,00%
BR_Ovd_Debts90d_Cur_Dr	X			100,00%
CU_Acteur_Code			X	100,00%
CU_Country_Code	X	X	X	100,00%
⋮	⋮	⋮	⋮	⋮

Table 9.10: Global XML message for RMS in France, Germany, and UK

CHAPTER 10

Prototypic Requirements Elicitation and Evaluation of Application Information Needs

Christian Tonhäuser (christian.tonhaeuser@gmx.net),
Thomas Dierl (dierl@in.tum.de),
Florian Gall (gall.florian@gmx.net),
Franz-Ferdinand Müller (ffmueller@googlemail.com),
and Gerrit Remané (gerrit.remane@in.tum.de)

10.1 Overview

This method describes an approach to elicit the information needs of stakeholders who are possibly affected by the establishment of enterprise architecture management (EAM) process and a supporting tool in an organization. Furthermore, it depicts how these information needs and functional requirements can be used to derive problem specific visualizations, and how appropriate V-patterns can be described. The method should be used at the beginning of an EAM establishment project as its results prove/disprove the project's necessity, define its scope and affect implementation details such as the information model.

10.2 Example

The insurance company GloboInsure has been growing fast in recent years. The reasons for this were the organic growth as new customers were acquired and new markets were entered all around the world. In addition, GloboInsure pursued an active mergers and acquisitions (M&A) strategy; it bought and integrated former competitors in order to extend the product portfolio offered to its clients.

These activities have a big impact on GloboInsure's current information technology (IT), resulting in an increasingly heterogeneous application landscape, multiple interfaces between these applications, and opaque responsibilities. This complexity itself poses a problem, but more critical is the lack of documentation since there is no central information source or tool that contains the relevant information about the application systems, their interconnections, the hardware they are running and the business processes they support.

GloboInsure decided to initiate an EAM-project which should set up an EAM tool and establish the corresponding EAM processes. To do so, GloboInsure first of all analyzes the current as-is application landscape in order to enable the company to define a target landscape and finally prioritize and implement projects. As a long-term vision, these steps will lead to a homogenized application landscape.

In the first phase of the establishment for an EAM process GloboInsure has to find out how a prospective EAM tool would be used in the day-to-day-life of its employees and which information about the application landscape has to be represented to fulfill their requirements. In addition, it also has to analyze, how the needed information has to be visualized to guarantee ease of use.

10.3 Context

This method is applicable in the early phase of basically every EAM project. In particular, if the application landscape of the organization is relatively big (typically more than 50 different applications), strongly connected and growing, the need for a central "single-point-of-truth" EAM tool can be regarded as crucial. Another aggravating factor is the company size; the bigger the company, the more complex it is to identify the employees, who might use the EAM tool and to analyze the requirements they have. These stakeholders are dispersed over multiple business functions and departments, and might even not be aware of the fact that they are potential users of the EAM tool. Company size also affects the accessibility of application information; financial information is often used and stored only in the accounting department, the underlying hardware infrastructure might be outsourced and the processes supported by the application might not even be explicitly stated.

Another important contextual factor which affects the necessity to apply this method is the experience of an organization with the topic of EAM. If the project is "first-of-its-kind", the employees usually do not know the possibilities an EAM tool offers to improve transparency and to support the development of the application landscape. In this case, the actual business-side information needs are still undocumented, implicit and therefore difficult to discover. A structured approach such as the one described in the solution section is required to identify and document them. Experience with the tool that should support EAM is also an important factor since the information needs have to be mapped into a so-called information model. However, the mapping into a complete and correct information model is a prerequisite for a tool-supported EAM, which is mandatory in today's application landscapes with hundreds of systems and even more relationships.

10.4 Problem

The basic problem you may face in an organization is a more or less uncontrolled growth and quick changes of application landscapes over time. This leads to problems regarding the management of the applications because either no one or only a very limited number of people can comprehend and monitor the increasingly complex system of systems. Besides the difficulty to identify these people, they might also be located in different departments or business units of the organization. Additionally, you might have to search for information, when there is a need for it for a specific task, since most of this information is divided geographically and you cannot access this easily. This lack of (standardized) documentation implies that there is no persistent and accessible central information pool which helps you to analyze the application landscape from different viewpoints. Consequently, there is a massive administrative overhead and coordination effort, when you are trying to access application information such as costs, interfaces, responsibilities or underlying hardware.

To address this problem, you can use EAM tools and processes, which has been a field of research in recent years. The establishment of such tools, however, is a difficult task and needs structured approaches. The remainder of this paper describes a generalized procedure which supports a first step of this establishment process: the identification of information needs and the elaboration of corresponding visualizations.

10.5 Solution

For identifying the requirements and information needs that should be satisfied by a central EAM tool, a very systematic approach is mandatory due to plurality of stakeholders and needs. For this purpose it is recommended to proceed in three main steps:

1. Gathering information
2. Creating information model
3. Creating visualizations

A more detailed plan can be seen in Figure 10.1 and will be described below.

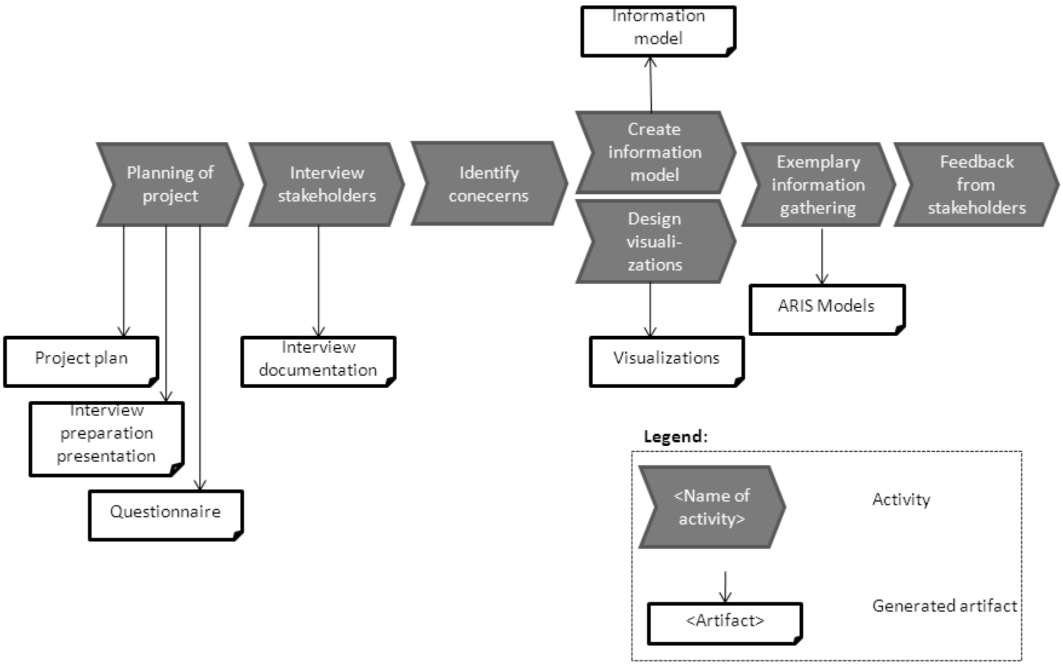

Figure 10.1: Project plan

The precondition for this plan is that the stakeholders for the EAM tool who should be interviewed were already selected. If this was not done yet, it is important to identify stakeholders from both business and IT departments and especially from the EAM group. The number of interview partners should not exceed ten to facilitate the smooth consolidation of the results.

In the first phase a project plan has to be created containing buffer periods and deadlines. Afterwards the interviews with the stakeholders have to be prepared. The interviews are crucial to gather information needs and possible visualizations as well as exemplary data for the prototypic implementation. For more information on this topic and an exemplary questionnaire see Sections 10.5.1 and 10.6.

After conducting the interviews, the results have to be evaluated. This should be done by developing the visualization and information models in parallel due to their relationship with each other. Thereby leaving out necessary classes in the information model is avoided which might cause problems otherwise. During the evaluation the focus should primarily be on the main needs (which are mentioned by more than one stakeholder) and on the importance of every information need to avoid a too complex model.

In the next step the developed information model has to be filled with the content of the applications in scope, and suitable visualizations have to be created. These preliminary results need to be discussed with the interview partners in order to receive feedback to check, if their expectations were met. If this is not the case, necessary adjustments have to be made to the information model and/or the visualization until the results are satisfactory.

10.5.1 Interviews

The first phase is already a highly critical one, at least from a time management perspective. The major challenge is to agree on appointments for an interview with all relevant stakeholders which can be a difficult task, especially if stakeholders from higher management levels are involved. There are several options to design the interviews, each with different advantages and disadvantages. However, the structured interview is a good choice since it helps to focus on key issues. A questionnaire is the first necessary artifact for conducting structured interviews. It is recommended to use the same questionnaire for all stakeholders in order to get comparable results. However, if the stakeholders have significantly different backgrounds or functions, some questions may be customized. Good questions in the questionnaire are those that help the interview partner to identify his most important needs. Therefore, it is not effective to ask questions like: "What information do you need?" or "What would be a good solution?", but to ask questions from the three categories shown in Figure 10.2 in a cyclic process. An example questionnaire can be found in the appendix.

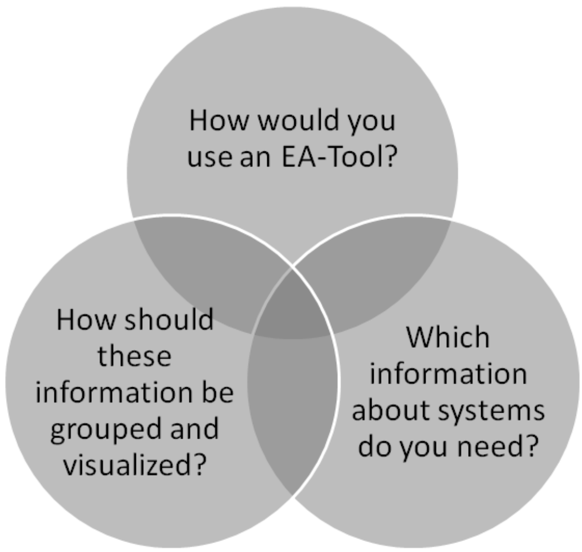

Figure 10.2: Project plan

To prepare the interview partners for the interview sessions, it is recommended to send them a short introduction or presentation upfront. It should at least contain the following topics:

- Goals of the project, context of the interview
- Overview of the interview content and process
- Examples, e.g. possible visualizations, example attributes, use cases, etc.

In the actual interviews, a short introduction to EAM or the purpose of an EAM tool may be used as opener, especially if the interviewer assumes that the interview partner would possibly profit by giving him some examples of usage. After this optional introduction, the questionnaire can be used to elicit the information needs. For documenting the answers of the interview partner, it might be helpful to have two interviewers. If this is not feasible, one could record the interview or even take a film thereof. The latter option allows analyzing the person's emotions expressed by intonation, mimic and gesture later on. However it should be considered that this requires the consent of the interview partner. After the interviews, one should exchange contact information (if not already done) to be able to get back to the interview partner for further questions. This is necessary because after having analyzed the interview, it might be necessary to get in contact with the partner to clarify open points.

10.5.2 V-Pattern

After preparing and conducting the interviews one has to interpret the results. On the one hand, you have to design possible viewpoints of the required information which have been mentioned by the interviewed stakeholders. On the other hand, this means to classify the resulting needs and derive an information model, which reflects all these needs.

The first step is to create viewpoints that satisfy the stakeholders' needs. Sometimes stakeholders mention these needs or even specific viewpoints explicitly during the interviews, therefore, simplifying the process. But in most cases the stakeholders do not directly formulate their information needs; they may only tell about their needs and wishes given a certain context. In these situations helpful viewpoints using existing V-Patterns from the EAM Pattern Catalog (EAMPC) [Bu08] have to be derived. The results of this task can then be presented to the stakeholders to get feedback and to improve the quality and usability of the viewpoints. This process can be iterated several times until the viewpoints fulfill the stakeholders' information needs. Typical types of viewpoints are e.g. Process Support Maps which can be found in the EAMPC as V-17. A similar map is the Responsibility Map which is shown in Figure 10.3, which shows the role of an actor according to an application system.

Another issue that might come up is the plurality of information needs of different stakeholders. Some stakeholders may need similar pieces of information but might have a different focus or viewpoint and therefore want different types of viewpoints for it. In these cases, the individual viewpoint needs and the complexity and diversity of the viewpoints, which increase with the number of needs, have to be balanced. Since the benefit of EAM depends very much on the acceptance of its deliverables, one ideally wanted to satisfy every need mentioned by the stakeholders, but since it is infeasible to implement every type of viewpoint, one had to prioritize and implement only the most relevant ones. The prioritization can be based on a variety of factors, but if several people ask for the same kind of viewpoint, this is usually a strong indicator for higher relevance. When creating and implementing the viewpoints one should use symbols and rules in a way that enables the user to understand the viewpoints' semantics intuitively, saying one should use similar symbols for similar elements across several viewpoints. Some general rules of thumb may be used here, e.g. based on the semantics of the Process Support Map the x-axis usually describes processes whereas boxes mostly represent applications or application systems. In some cases, it can also be helpful to add graphical illustrations, e.g. a computer symbol when describing servers or other technical resources. Colors may be used to describe attributes, e.g. a color coding for low, medium or high levels in green, yellow and red. In order to describe the semantics of a type of viewpoint, a legend is a vital component. In this legend all the symbols and visualization rules are defined in order to make the viewpoint distinct and unambiguous. The main goal in this respect is to create simple and easily understandable viewpoints as the stakeholders will use these for a longer time and should hence be able to understand them quickly and easily.

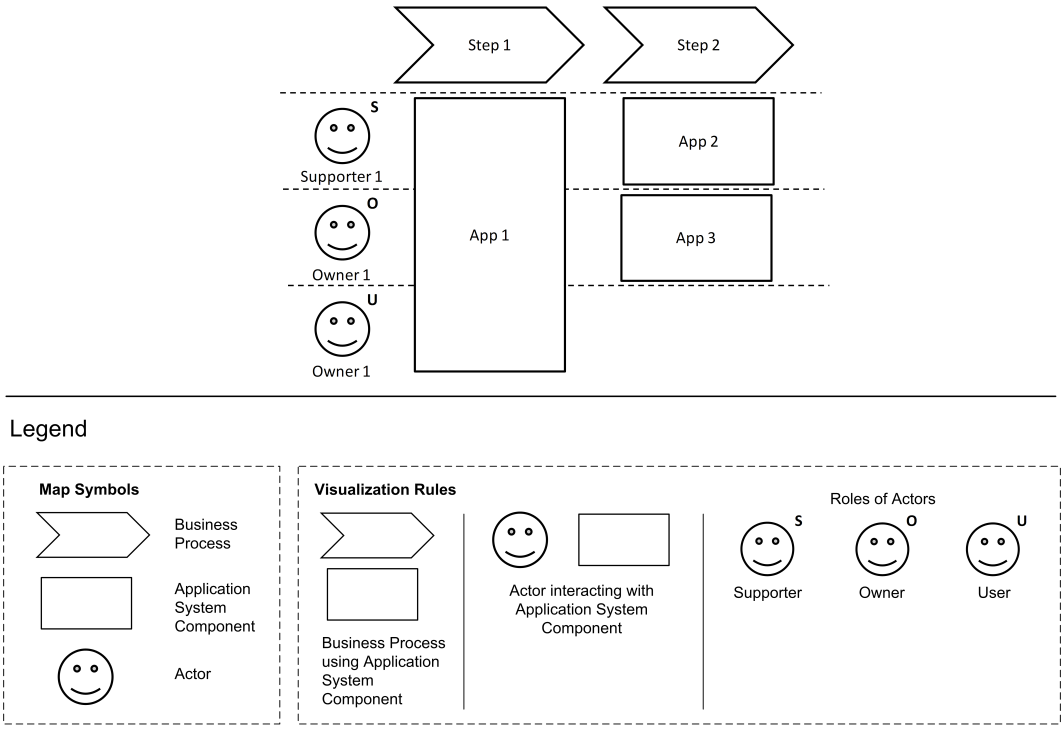

Figure 10.3: Responsibility Map

10.5.3 I-Pattern

Parallel to the creation of visualizations described in section 10.5.2, the information that is required by the stakeholders and is needed to create the visualizations has to be compiled into an information model. In this model the required information attributes and the relationships between them can be displayed. Usually this model is created using a well defined modeling language such as the Unified Modeling Language (UML), i.e. the UML class diagram.

On the one hand you have to make sure that the visualizations, which were derived in the first step can be created using this information model. On the other hand you have to try to keep the model as simple as possible to be able to modify or extend it easily in the future. This will result in a limited amount of data which has to be maintained and therefore in a higher degree of acceptance by the users. For this reason some requirements should be omitted in the model in case they are not of particular importance and they make the model significantly more complicated. This of course generates a feedback loop to the creation of visualizations since it restricts the concepts they can show. The information model and the visualizations depend on each other and their development requires permanent harmonization.

Common elements within the information model can be found as I-Patterns in the EAMPC and should be used in the right context for the information model. As these models are quite abstract and therefore difficult to understand for other people, it is hard to discuss these types of results with the stakeholders. Nevertheless, a step of quality assurance is necessary to avoid false representations of reality, i.e., the multiplicities of relations should be discussed thoroughly since this is an important element of the model and has many implications on the visualizations that can be derived.

During this process several issues may arise. Some I-Patterns might become necessary to enable you to implement certain types of visualizations. In some cases this can lead to difficulties as different visualizations can require different I-patterns in the information model. Then you have to choose one type of visualization over another depending on the stakeholders' needs. Further, one I-Pattern may even become vital to your model as this might be the basis for several visualizations, e.g. the `SupportRelationship` (I-30, see Figure 10.4) is the basis for several types of maps such as the Process Support Map or the Responsibility Map. In this case, this pattern should be the starting point. Figure 10.5 shows an information model which is built around the pattern (I-30) mentioned above.

Generally, there are several patterns involved in a model. Your main task is to combine them in an effective way and to add specific elements and attributes where necessary.

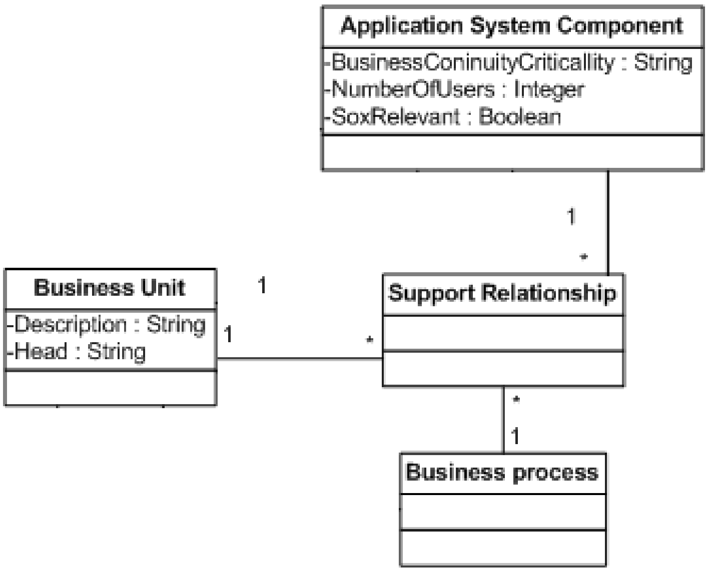

Figure 10.4: Process Support Relationship I-30 (modified)

When creating the information model, you have to keep a constant level of abstraction when describing the objects within the model. This may be difficult as some of the stakeholders' requirements are formulated quite concrete whereas other elements might not be described in a detailed way. So you have to find abstractions and generalizations for elements within the model where necessary. But there is no exact pattern how to do this, the elements and their level of abstraction heavily depend on the context and the way they were formulated by the stakeholders.

When discussing the information model with the stakeholders it might also become necessary to use a different representation of the model since many people will not understand UML class diagrams. So you have to create other graphical or textual descriptions of the model which can then be the basis for discussion. As usual, concrete examples using existing data can help to quickly comprehend the concepts and to achieve common understanding.

As the information model is a central component of EAM in this context, this step is very important in the process. A high quality model is the basis for high efficiency in operation and can easily be implemented in an EAM tool.

10.5.4 M-Pattern

In this section typical usage scenarios of an EAM tool are described as M-Patterns which were identified from the interviews. The information model and the visualizations rely heavily on these methods since they have to fulfill the elicited requirements.

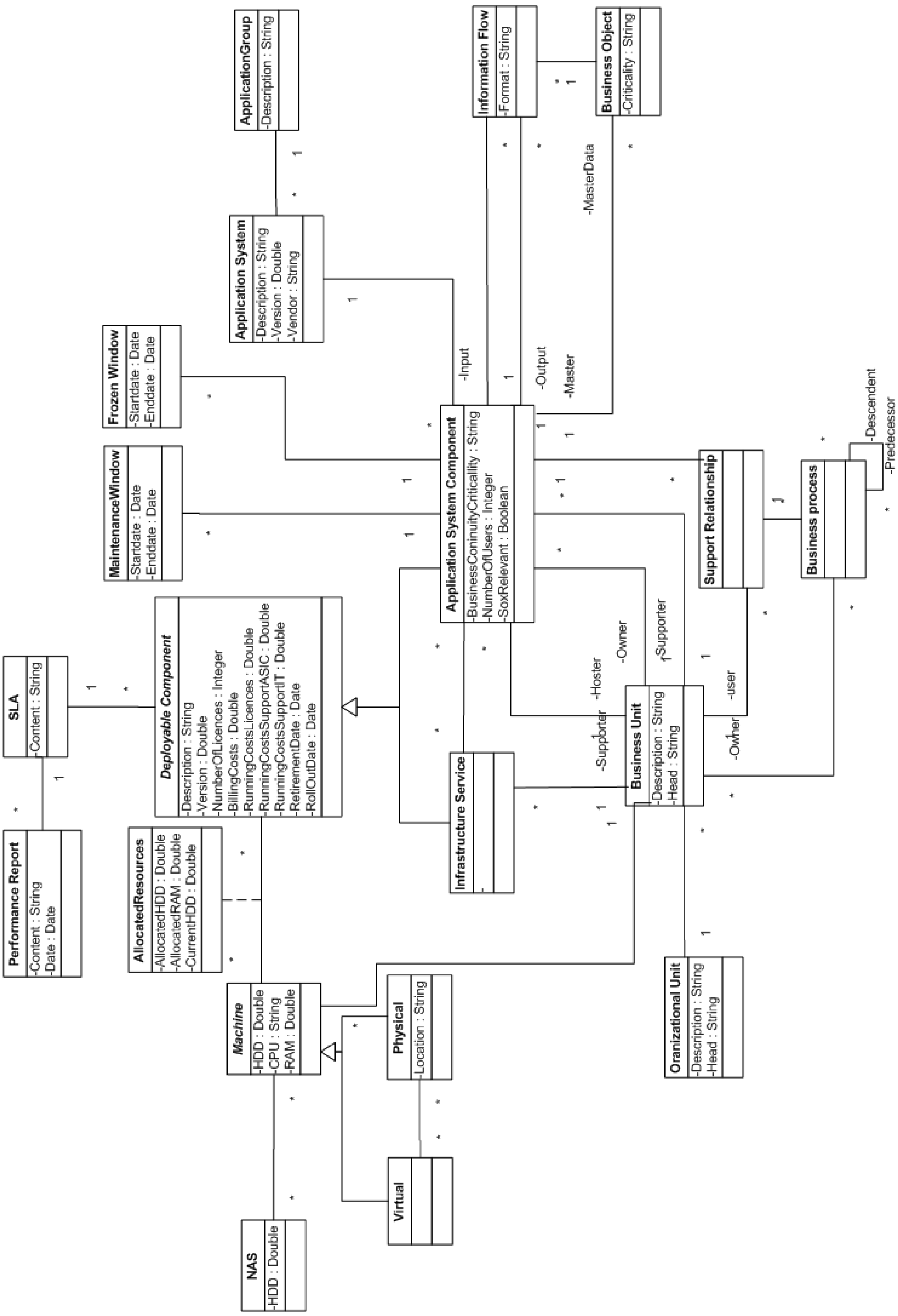

Figure 10.5: Exemplary Information Model

10.5.4.1 Planning the roll out of system-changes

M-Pattern Overview

This method describes the planning process of roll-outs for software updates and bug fixes.

Problem

The IT and business departments usually agree on the availability of certain services as a share of the total time for fixed periods (i.e. 99% availability per year). In many cases, however, this agreement is not detailed enough to meet the business' expectations, since there are periods in which the availability of systems is very important whereas sometimes availability is an issue of low importance. An example is financial accounting; during the monthly payroll accounting availability is crucial whereas the rest of the time the system is on idle. Therefore, the business and IT departments have to communicate business critical periods in detail, containing the critical period and the business impact of non-availability during these critical periods. In these periods there should be as few changes to the system as possible, because changes often cause unexpected behavior.

Solution

Most maintenance operation on systems like updates, bug fixes and configuration changes can be planned ahead, so at the beginning of a new planning period the IT departments should define fixed periods for maintenance activities and document them in the EAM tool. However, this may result in conflicts with business units' operations, if they execute critical business function during the prospective maintenance times. To avoid such conflicts, the business units need to define so called "frozen windows" in the EAM tool, during which changes to the system are not allowed. As selection criteria, the business needs to check, in which periods of time their business processes are running and/or are in a highly critical phase. So whenever the IT is planning to perform maintenance operations, the system they need to ensure not to deploy during a "frozen window".

Implementation

The company needs to have a central documentation collaboratively used by business and IT departments to avoid interruptions during critical business process phases. Best practice for this would be a central EAM tool, from which all departments can draw the information.

Consequences

Conflicts between business and IT are reduced by using a common documentation and checking for frozen and maintenance windows before system changes.

10.5.4.2 Planning of hardware resources

M-Pattern Overview

This method describes the planning process for hardware resources.

Problem

Reducing IT operating costs and ensuring a certain level of IT performance are top-level goals of nearly every IT department. To achieve these goals the load of hardware resources is an important factor. The balance between efficient utilization, performance, and costs needs to be defined.

Solution

There are two main occasions for planning the size of a systesizing of hardware resources.

1. During the annual budgeting process
2. If big changes to the number of users are to be expected, for example by acquisition of another company

During the planning process the utilization of the system, including the utilization history, needs to be known. If virtualization is used the work load of the physical system needs to be included as well. Key indicators are the maximum and the average utilization, but also machine age, failures and criticality of hosted applications have to be considered.

To plan the future needs of hardware resources the actual and future number of users needs to be estimated and compared to the current utilization of the system and the target performance.

Implementation

The company needs to have a central documentation for hardware utilization, application deployment and the number of users collaboratively used by business and IT. Best practice for this would be a central EAM tool.

Consequences

Recurring projects with the target to determine the number of users and the utilization of hardware resources become obsolete. Planning the utilization considering the suggested indicators leads to an optimized balance between performance and costs.

10.5.4.3 Managing consequences of failures of IT components

M-Pattern Overview

This method describes the process of identifying the consequences for the business in case of failures of an IT component (be it hardware or software) and techniques for managing them.

Problem

The highest priority of virtually every company is to ensure its future existence. To achieve this and to gain the trust of the investors, every company should be aware of risks threatening this top goal and be ready to handle them if they become reality. Best practice for this is a defined risk management process.

Due to the high importance of IT, the failure of IT components poses a risk to the company. To quantify this risk and to determine how to handle it, the company needs to know the consequences of errors of IT components.

Solution

Identifying the consequences of failures of IT components for the business requires a detailed assessment of the hardware and software resources of the organization. This can be done in two steps:

1. Check, which business process is supported by which application. Based on the importance of the business process assign a level of criticality to the application.
2. Check, which applications run on which hardware to identify the criticality of hardware components.

The consequences of a hardware or software failure can be assessed by asking every business unit what would happen, if certain applications did not work anymore. This requires a rather time consuming analysis, depending on the level of detail you want to achieve. In addition, one can also analyze factors like machine load and age to estimate the probability of failure of hardware components.

Knowing the criticality of all hardware and software components, the reliability of hardware components and the consequences of a failure, it is possible to estimate the expected amount of damage and to define proper actions for all identified risks. To achieve holistic and consistent risk management ensure that this process is running permanently.

Implementation

The company needs to have a central documentation for dependencies between hardware, software and business processes collaboratively maintained by business and IT. Best practice for this would be a central EAM tool.

To ensure the future management of the risk a periodical process should be defined.

Consequences

Risks regarding failure of IT-components and their consequences are consciously and permanently managed.

10.5.5 Implementation

The implementation of EAM processes does not necessarily require a specific tool, but the usage of a tool can increase the quality of visualizations and of the collected information. It also facilitates the management of large amounts of data, especially, if changes occur frequently or information is provided by more than one department. However, the decision in favor or against an EAM tool depends on the organizational context.

If an organization decides to use an EAM tool, the implementation of the specific EAM heavily depends on the results of the steps before and on the restrictions of the respective tool. A crucial activity is the implementation of the created information model as a meta model into the tool. To do so, the tool has to support the created meta model or an appropriate mapping has to be found. Another important criterion for the implementation is the definition of the above created visualizations within the tool.

Creating a real scenario with exemplary data is a proper method to evaluate, if a tool offers the required functionalities and to examine its ease of use. To develop a useful scenario in the tool, it is necessary to gather some information for a real world use case. For this a fraction of the whole application landscape has to be defined and the according information has to be collected from different stakeholders. The problem about this is that it can only be done after the information model is completely defined, because only then the information needs are known. Subsequently, the stakeholders have to collect the necessary information, which implies some effort for the stakeholders; hence they should have some incentives to do so.

The prototypic implementation within the tool allows the stakeholders to get hands-on experience with the tool in order to find out, if the implementation fulfills the given requirements in a real world setting. If they demand specific changes, they can easily be considered before the rollout of the complete project or, if the requirements cannot be met, the tool has to be discarded. In this case, another tool has to be chosen or, if no suitable alternative can be found, the project has to be stopped and completely rethought.

The test of the prototypic implementation solely by the interview partners, is usually not sufficient, because they all have been involved in the analysis and creation process, so it is advisable to look for some other stakeholders who were not directly involved before. This way the elicited requirements can be checked against the requirements of these people. In order to make the solution accepted by the stakeholders, it is necessary to have detailed descriptions of the visualizations and information ready.

10.5.6 Evaluation

After having implemented the visualizations, it is important to evaluate the results against the stakeholders' requirements. This is a very essential step since it measures the satisfaction of user needs and the process outcomes, and therefore determines the utility of the results for specific users and user groups.

It is recommended to split this procedure into three main steps:

1. Show and discuss results with stakeholder
2. Incorporate user feedback
3. Approve and promote results in a public meeting with all stakeholders

The first step is to show the results to the interview partners and to discuss the outcomes with them. Since they were the primary source for gathering user needs during the requirements elicitation, they may already have a certain idea about the desired results and can quickly determine missing features and deviations from the previously discussed concepts. They are also able to give feedback about the usability of the visualizations.

The second step is to refine the information model and visualizations according to the users' feedback. Usability issues can usually be resolved by changing the visualization whereas missing features or deviations may require altering the underlying models. After applying changes another iteration of step one is due to check again the fit of user needs and produced results. This step is crucial since it determines the acceptance and therefore the success of the outcomes.

Finally, step three is to introduce the results to a large group of stakeholders. Since the interview partners already approved the models and visualizations they are likely to find acceptance with other stakeholders as well. The interview partners may also act as promoters for the outcomes and therefore further increase user satisfaction. If no major concerns are voiced during this step, the results can be considered as fitting the user needs as well as other stakeholders' expectations.

To show results to stakeholders, a live demonstration of a prototype is recommended. This way the possibilities, the handling and the restrictions are easy to comprehend and to comment on. For this one can collect data for a set of attributes which correspond to specific user needs or use cases. Based on this, several basic visualizations are created to demonstrate how the subsequent EAM tool would present relevant information. If crucial attributes cannot be obtained or data is incomplete, reasonable assumptions can be made, but it is important to keep in mind that "guessed" information might have less persuasive power than real data.

10.6 Consequences

The main outcome of this method is a set of detailed requirements, models and visualizations which satisfy the identified user needs and are the basis for the selection of an EAM tool. It is also possible to implement a prototype of the developed EAM to demonstrate the functionality and the handling of an EAM tool based on the requirements. If such a prototype exists, it is also possible to conduct a real-life test with a limited number of users to refine the requirements and to discover problems and issues which did not occur during the evaluation.

Besides the main outcome there are also several other consequences of the requirements analysis. First of all the resulting prototype facilitates quick and comprehensive demonstrations of the developed concept and the requirements since only a low level of technical knowledge is required. This allows a variety of stakeholders and interested people to understand the main issues and to give feedback on the result. Second the prototype might serve as a reference model during the evaluation of potentially suitable tools. It can be used to compare different visualizations and to identify missing features. Finally, the process greatly increases the awareness for EAM throughout the organization. Stakeholders get to know the topic and other people with similar concerns and needs, and they are more likely to collaborate on EAM issues in the future.

A potentially negative consequence of the approach is the focusing on a specific target group. Since it is usually not practicable to elicit the requirements of the whole organization, it is only possible to approach a specific group of stakeholders. Their concerns and needs, however, do not necessarily reflect the concerns and needs of the whole organization. This problem can be countered by choosing an interdisciplinary group consisting of members of different departments, but it cannot be avoided at all.

Appendix

1. Interview partner:
2. Contact with application system
   Requirements for EAM tool

Use case (including frequency) "How would you use the EAM tool for your job?"	**Required information** "What information do you need to do this?"	**Possible visualization** "How could this information be visualized?"

3. Contact data:

Bibliography

[Bu08] Buckl, S. et al.:. *Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008).* Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2008.

CHAPTER 11

Analysis and development of an Architecture Check Process embedded into Enterprise Architecture Management

Maximilian Lacher (maximilian.lacher@mytum.de),
Adrian Staudt (adrian.staudt@mytum.de),
Felix Willnecker (felix.willnecker@mytum.de),
and Amao Xu (amao.xu@mytum.de)

Abstract:
This paper deals with the architecture check process (ACP) of and international insurance company (International Lean Six Sigma - ILSS). It explains how mechanisms of the Enterprise Architecture Management and patterns from the EA management pattern catalog of Technische Universität München can be used to optimize and standardize this process. This is necessary, as the ACP is an ad-hoc process without clear limitations or constraints. Every single step is reinvented over and over with each process execution. The first goal of this research is hence to create a documentation of the process. Understanding the ACP as an EA management process, EA management patterns are applied to provide a more standardized workflow.

11.1 Motivation

The company, named ILSS, under consideration is a multinational insurance company. In 2008 the ILSS charged about 25.5 billion Euro insurance contributions and engaged over 30000 employees in Germany [AG09, Ul09]. Its head department of IT is seated in Unterföhring near Munich, Germany. Most of IT-related decisions are undertaken in the IT department based on specifications designed by the ILSS itself. Especially IT architecture relevant decision strongly depend on constraints, which guarantee the stability and reliability of the whole IT infrastructure. To decrease heterogeneity and to promote standards for the architecture the ILSS created a catalog called CREDO. This catalog describes which standard software must be used for different architectural layers and also gives best practice advices for layers that are not strictly defined.

In terms of CREDO, standard software components are defined by their layer and the operating system, on which they are installed. Furthermore, the standards defined by CREDO are categorized in five different classes, ranging from mandatory standards and best practice advices to architectural anti-patterns. These guidelines need to be followed, if the project should be approved by the ILSS. Complementing the architectural constraints imposed by CREDO, additional legal requirements concerning the IT infrastructure apply. An exemplary requirement demands that the development infrastructure, the testing infrastructure and the deployment infrastructure are strictly kept separate.

11.1.1 Problem Context

In order to ensure the enactment of the CREDO standards and the fulfillment of the legal requirements, the ILSS established an architecture check process (ACP). This process helps to ensure a stable operation of IT systems, to guarantee the homogeneous development and to increase the degree of standardization and homogenization in IT. The ACP consists of the following phases: Plan, Prepare, Analyze, Decide and Execute. Each of these phases provides output documents and information, which is consumed by the subsequent step. Here, a major problem of the ACP becomes obvious: the output types of the different process steps are only partially standardized, such that multiple document formats are employed in interchanging information between the different steps. Further, no standardized mechanisms to link the ACP to the project inception activities in the company is given. In the light of these two problems, ILSS experiences low transparency in respect to project execution as well as low traceability of project decisions to underlying architectural standardization constraints.

Every year over 200 projects are executed in the ILSS, of which a majority of approximately 75% does not need architectural conformance checking and approval, as they have no high architectural relevance. The other projects, which heavily change the IT infrastructure or introduce a new system, need to be identified and to be analyzed in order to ensure architectural conformance. How difficult this analysis phase is depends on the documentation provided by the individual project teams, which prepare description documents that are transmitted to the ACP Team. These documents differ in notation and level of detail. To improve the documentation and rationalize the analysis process, the ILSS needs patterns and recommendations to standardize their project documentation. The aim of the mini project is the development of an M-pattern, which documents the different phases of the ACP and supports the communication between the involved stakeholders (architects, project managers, etc.) with appropriate V-patterns. In addition, I-patterns should be created, which explicate the information used for the ACP.

11.1.2 The ACP as an EA management process

In the following we shed light on the ACP by approaching the topic from an EA management perspective. This can be justified by the results gained from an interview with a responsible manager in the insurance company, who pointed out the key characteristics of ACP. These characteristics matched the ones, we could expect from an EA management process. Figure 11.1 gives an overview on EA management and related processes, of which the ACP-relevant processes are highlighted by a border.

The interview revealed, that the ACP is executed as part of the EA management and the IT architecture management process, which are in the ILSS executed in one division. This is justified from the company's experience that shows that an enterprise architect not only has to define principles and guidelines, but also understand the IT systems. From this, the following proverb was derived "an architect always implements (up to 20% of the working time)". Thereby, the architect can build up a deep understanding of the systems and increase his knowledge with on-the-job training.

Enterprise-wide requirements management is in the company supported by a tool named PLANVIEW, which is used to collect proposals for IT projects. In addition to this, the tool provides functionalities used for reporting on the projects and proposals in a management dashboard. After the proposals have been gathered, the most important and innovative proposals are selected for execution in a project

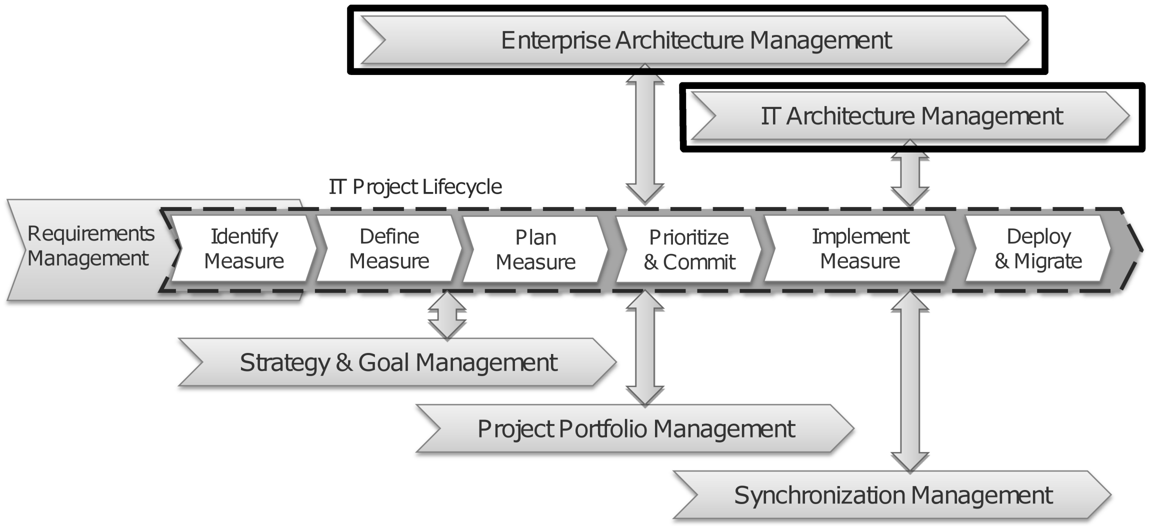

Figure 11.1: EAM and related processes

by a board called ”Betriebswirtschaftlicher Lenkungsausschuss”, who further compiles a prioritization of the proposals. In a final step, the decision on the project portfolio for the next period is decided. During the decision also information on the architectural conformance of the projects is taken into account. This information is provided by the ACP, which can in this vein be identified with the ”Assess EA” activity from the process framework described by TU München (see Figure 11.2).

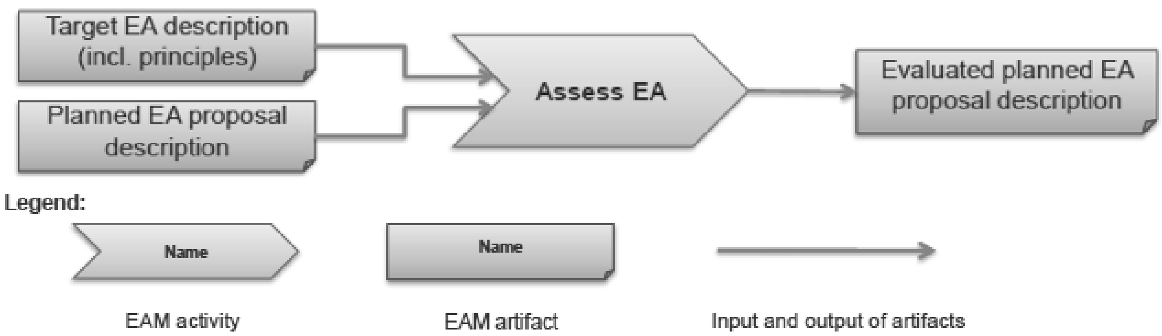

Figure 11.2: Dynamic EAM model

In a first step during the ACP, the teams checks, whether the new project is relevant for the ACP. If this is the case, the project proposal and the architectural requirements (”target architecture” in terms of Figure 11.2) are compared. In most cases, the ACP results in an approval for the project, while some conditions are imposed. Examples for such conditions are technologies, platforms or programming languages that have to be used instead of what originally was planned by the project team. The output of the ACP is called ”Fit document”, which describes the imposed conditions.

11.1.3 Paper Structure

At first we have characterized the environment in which we were working and have given a short overview over the tasks that we have focused on in this project

After this short introduction we will present the main problems that we found and had to face in the first section in detail. We will explain which problems existed and how we discovered them. In the following chapter our approach to solve these problems will be described and our results will be depicted.

In the final part of this paper we will demonstrate the consequences and trade-offs that our solution has to overcome.

Finally we will give a summary about our work and an outlook for the future steps that the company should do to improve their enterprise architecture management.

11.2 M-Pattern: Architecture Check Process

Below, we establish a method that can be used to check the architectural conformance of a project. The method employs different visualizations that comply to the viewpoints as presented in Sections 11.3 and 11.4. Further, project specific visualizations are used during the method. These visualizations do not directly link to the ACP and are hence not discussed in detail during the work. Figure 11.3 gives two examples of such visualizations, to which we come back in the outlook of the paper.

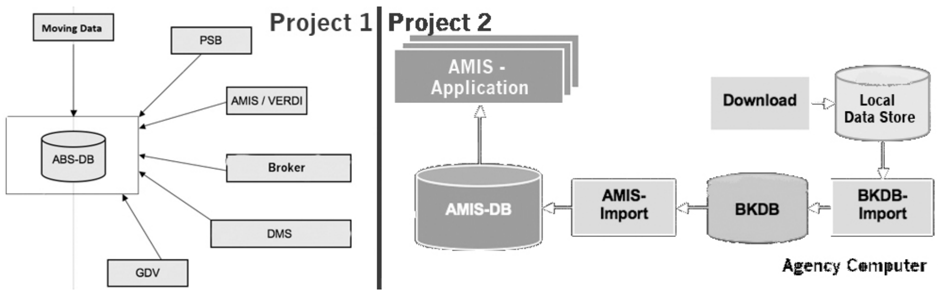

Figure 11.3: Two exemplary visualizations for architecture projects

11.2.1 Solution

The ACP (cf. Figure 11.4) is the main institution for checking projects, after they have been approved by the Project Portfolio Management. It is mainly a process but defines furthermore a team, which owns and runs this process, the ACP-Team. The ACP-Team analyzes and discusses the planned infrastructure changes to fit to the ILSS standards and legal constraints. The decisions made by the ACP must be followed. It is therefore important to prepare the ACP process carefully by the project teams.

The ACP decomposes into five relevant phases, which are detailed subsequently.

Plan phase
The plan phase is completely done by the project team that needs approval for their project. Main objective of this phase is to plan the project considering time and involved components. This is necessary to prerequisite for the remainder of the ACP.

Prepare phase
The prepare phase is mostly done by the project team. To gather and to arrange the information for the check is the main task of this phase. The ACP team decides, if the provided documentation is sufficient to execute the check process. Furthermore, the ACP team can decide that the check is not necessary, because the project is not of high architectural relevance. This decision can be made based

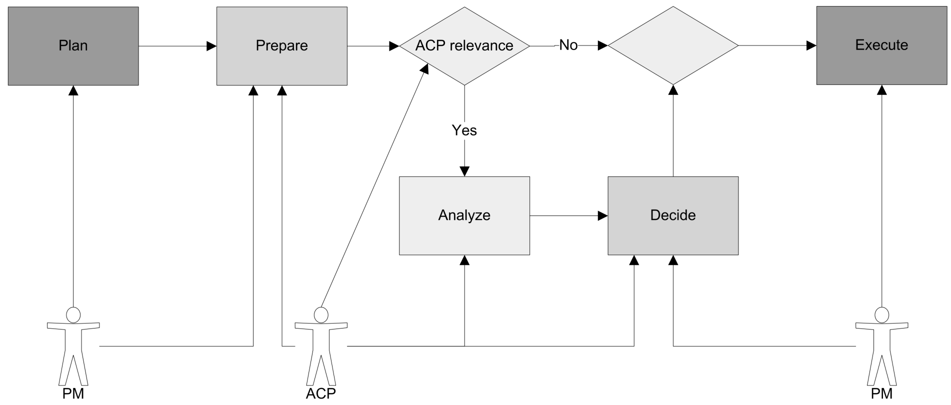

Figure 11.4: Overview of the ACP process

on the ACP preparation checklist (cf. Section 11.3). As a side effect, the project team is forced to think intensively about system capacities and architectural resources. Therefore, the quality of the project plan may increase.

Analyze phase
This phase is the main phase of the whole ACP with a rich internal structure, cf. Figure 11.5. After deciding that the project influences the architecture of the company, the ACP checks the documentation to minimize effects on the business architecture. Besides the CREDO standards and legal issues, the capacity of the systems and the communication channels need to be analyzed. Decisions like increasing the architecture resources arise from this phase. Most important for the project team are decisions that effect the project plan and chosen technologies.

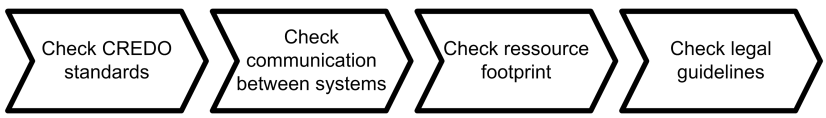

Figure 11.5: ACP Analyze phase

The main benefit for the ACP team in this phase, is the standardized documentation created by our improvements for the process. The artifacts submitted are now strictly defined and therefore reduce the overhead for the ACP Team.

Decide phase
The last main ACP phase is the decide phase. The project team, usually represented by the project manager and a ACP team member bargain about conditions for the project. These conditions must be followed.

Execute phase
The execute phase is the development and deployment phase of the project. The main difference between an usual execution phase is that the conditions out of the ACP must be considered.

Force: General Information vs detailed information
The conflict here is, how detailed information is necessary for project visualization diagram. On the one hand, detailed information is preferred for the ACP to help making evaluation and decision easier. On the other hand, preparing detailed information is costly and properly unrealistic for the project team. For example, certain information for a project is hard to forecast at the plan phase. In this case, there could be a big deviation between the estimated the value and actual value. Therefore the

balance of generalization and specialization must be controlled and reconciled between ACP and the project team to make the visualization diagram for a project meaningful and convinced.

11.3 V-Pattern: ACP Preparation Checklist

Below we detail a checklist that can be used to gather information on the architectural relevance of a project.

11.3.1 Context and Problem

In a company, the projects of high architectural impact, e.g. as they introduce new business applications or change the technology stack, should be assessed and provided with additional guidance for execution. It is nevertheless not feasible to assess every project due to the sheer number of projects to be executed. Therefore, you want to establish a checklist to prepare the selection of the projects of highest architectural impact.

11.3.2 Solution

Each project manager has to fill out the ACP preparation checklist during the project inception phase. This checklist consists of three sections shown in Tables 11.1, 11.2, and 11.3, respectively. In the case

Question	Yes	No
Is a new system developed?		
Are expenses expected for the project from more than 50 person months?		

Table 11.1: ACP preparation checklist – part I

that one of the two questions listed in Table 11.1 is answered with "Yes", the project has to undergo the ACP. While the two questions especially allude to newly developed systems, the second part of the checklist refers to projects that change already existing systems. With the help of the questions in Table 11.2, the architect can assess, if the overall architecture is changed in a way that need guidance and architectural checking.

Question	Yes	No
Has the communication flow changed?		
Is a new technology used which was not yet in the existing project? (hardware or software)		

Table 11.2: ACP preparation checklist – part II

If the project changes an existing application, which was relevant for the last ACP, and one of the questions listed in Table 11.2, the project must undergo the ACP. From an architectural point of view, this especially applies, if the communication between application landscape constituents is changed. A change could be a modification from asynchronous to synchronous communication. Another possible change could be that a new component can provide the same information but is e.g. more reliably than the current used component, leading to the replacement of the current component. The involvement of new subsystems poses another critical point. All related projects must be checked again, due to the fact that the communication flow may be changed. Therefore, these projects must be checked, even if all corresponding modifications conform to the intended architecture.

In addition to the above mentioned questions, the project manager must also describe the main use cases of the project. With this information the EA management team can prioritize the applications that should be assessed. The questions shown in Table 11.3 may be helpful for this assessment.

What is(are) the critical use case(s)?
Critical Use Case 1
Critical Use Case 2

Table 11.3: ACP preparation checklist – part III

If the project manager creates a documentation following the aforementioned viewpoint, the EA management team should have sufficient information to choose and prioritize the projects for the ACP without having the need to enter face-to-face communication with all project managers.

11.4 V-Pattern: ACP Project Assessment Document

Below, we sketch a questionnaire that can be used to gather information necessary for project assessment.

11.4.1 Context and Problem

In a company, a well defined set of architectural standards exists to which all business applications should comply. After having identified projects that introduce new business applications or rework the architecture of existing ones, the planned architectural changes should be assessed. Getting the necessary information about the planned architectures from the project teams in a concise and easy to use manner, is the problem, which is addressed by the subsequent solution.

11.4.2 Solution

A *Project Assessment Document* (see Table 11.4) may provide the information necessary for assessing the architectural changes that a project is planned to realize. Therefore, the document identifies the affected applications and relates them to the layers introduce in a standardization document (as CREDO in our example). The standardization document further specifies the admissible applications for the different layers, which can be used to identify standard deviations by the project.

Layer	Used Applications
Web Server	
Application Server	
Messaging	
Database	
Operating System	

Table 11.4: ACP project assessment document

Actually the project development team or at least the project manager should know what applications are standard compliant, but experience has shown that in the past many project manager did not know about the standards. This short table will not create a huge effort to fill out and the EAM-team can easily figure out, if all used applications comply with the company's standards.

The second part of project assessment document takes a deeper look into the components developed in the project and their interaction with existing systems. The components' properties that are described in this part of the project assessment document are expressed via *key performance indicators* (KPIs). These KPIs, on which we elaborate below, target both the developed system and the development project.

KPIs are important for the measurement of resources, that need to be taken in consideration. If you know when your application is under high utilization, it is possible to add additional resources for that period of time, but if you do not, it is possible that the application cannot respond in the specified period of time. Use cases like that are important for critical applications. Finally, critical applications need to be checked by the EA management-team so that KPIs need to be checked within ACP.

One of the most important KPIs is the size of a message. Messages are transmitted from a component to another. If you know the size of each message, it is possible to calculate the necessary bandwidth for the underlying network. Another important and related KPI is the frequency of the sent messages. If you have two components, you will have to identify what kind of message types these two components will send to each other, how big the size of each message is and finally you need to identify the frequency of each message type. If you know that values, it is possible to determine the necessary bandwidth.

Coming from an EA managementperspective, it is also very important to know if sent messages through two components are symmetric or asymmetric. It is better to have asymmetric connections because this design is more loosely coupled and it is much easier to replace one of these components with another one. Symmetric sent messages can block the entire communication, if one of the needed components are temporary not available. This information is for the EA management-team of prime importance.

Other KPI deals more with hardware based characteristics. In the message part, the network bandwidth was a hardware based characteristic. Now we have to handle directly with computers and therefore it is inevitable to know how much memory or CPU time a component needs. Usually back end services runs on hosts (this means especially mainframes) the used CPU time of a component can be measured in MIPS (millions instructions per second). Sure it is hard to know in advance how much MIPS and memory a component will require, but handling with those sizes will result in a better understanding.

At least the project manager has a better understanding of his own project if he can provide this data. This situation leads to a win-win situation for the EA management-team and the project manager. Information density grows with each ACP related document. Most of the KPIs we got from our contact person. We had to determine how this KIP could be used for visualization.

11.4.3 Consequences

This solution has several advantages. On the one hand it makes the check process much more efficient for the EAM-team than before. Actually the EAM-team needs to ask the project manager to call in a meeting, so that he can explain in a face-to-face communication what the architecture means. This effort can be reduced with our documents. On the other hand these documents will increase the effort for the project manager. A nice side effect is that the project manager needs to know the own developed application in detail. Anyway this leads to a better understanding for the application architecture.

The main advantages are first of all, that the EAM-team can determine immediately if a project is ACP relevant or not. Second all, relevant communication through the components can be easily figured out. The development of this document was obvious because of the tight related standardization document.

11.5 Development of further V-Pattern

Up to this point, the ACP checklist and the ACP project specification document form two textual viewpoints that are used in the communication between the project team and the ACP team. Additional visualizations that are used by the project team actual exist but do not follow a standardized viewpoint. Introducing additional viewpoints could in this respect be helpful to leverage the communication of information between the two aforementioned teams. Indeed, the motto: "One picture (a good visualization) is Worth Ten Thousand Words." [Ba21] especially applies in this context. Preparing the development of such viewpoint, we elaborate on information that should be contained in a useful visualization:

1. Which dependency exists between the functional components?
2. How communicate different components with each others?
3. Resource requirements (software, hardware, communication bandwidth) for components
4. Request and response time between the components
5. Which components (dependency, communication, resource requirement) and how they are changed in application landscape

These above points represent functional requirements for further V-Patterns to be developed, especially from the ACP perspective. Meanwhile the requirements of project leaders and IT engineers for the V-Pattern also play an important role. According to their requirements the following attributes of V-Pattern should be considered during designing. First the visualization language should be simple and understandable. It is unnecessary and costly to prepare visualization documents if the read and write rules are too complicated. Second, the V-Pattern should be relativly precise and concise to reduce the complexity and cost. Detailed description about every new change in project is sometimes unrealistic for the project leader, and also unnecessary for ACP. A condensed V-Pattern is more preferred in this case. Third, the V-Pattern should be flexible. Considering the diversity of IT projects, which are delivered to ACP, the V-Pattern should be compatible in different situations with different aspects. For example, the V-Pattern should be designed to be suitable for not only the project, which aims to update the database version, but also the project, which aims to develop new application system for the IT landscape. These points are collected by opinion from relevant V-Patterns users. We regard them as our dysfunctional requirements for V-Pattern.

Based on the list of V-Pattern as contained in the EA management pattern catalog (see [Bu08]), we were further able to select V-Patterns that are possibly suited as graphical communication means for the ACP and project team:

1. V-5 Standard Conformity Layer
2. V-24 Cluster Map for hosting relationship
3. V-25 Cluster Map for using relationship
4. V-37 Effects of a Project Proposal on the application landscape
5. V-39 Effects of a Project Proposal on the application landscape (detail)
6. V-41 Cluster Map indicating standard vs. individual software
7. V-48 Cluster Map visualizing Business Object Flows between Business Application V-55 component Cluster Map

8. V-63 Information Flows
9. V-67 Standard Conformity Exceptions
10. V-76 Technology Usage
11. V-81 Communicating Applications

Each of these V-patterns represents only certain aspect relevant for the ACP. Most of them are related with each other as some V-Pattern can be used as a layer for the others. That means, it is possible to combine them in one visualization. Based on the above selection, we further elicited attributes for a general V-Pattern usable in this context.

1. Clustering rules. For V24, V25, V55 the only difference is the rule of clustering. And these V-patterns are actually the basic layers for the all of the other patterns. Therefore, the clustering rule should be the first basic attribute for the V-pattern.
2. Attribute of component (application) in different V-patterns, the attribute is different. For example V-76 shows the technology usage of each application component in landscape; V41, V67 shows the compatibility of components; V37, V39 show the change of components. In further development of these attribute we could also use specialized symbols to represent the resource requirements of component in the V-pattern.
3. Communication translation. Good examples for this are V48, V63, V81. Important information is the type of translated new, the type of interface, information flow consequence.

According to the requirements from ACP and other users of V-Pattern, it should be designed in precise and concise way. Therefore, we collect all the necessary attributes of the V-Pattern from EAM pattern together, using existing legends to create one condensed individual V-Pattern. Based on this, we insert also some new attributes, which are not referred in EA management pattern catalog [Bu08] but very important for ACP. The goal of this V-Pattern is to represent a global visualization of all relevant key information for a project. An exemplary visualization in Figure 11.6 illustrates a possible V-Pattern.

A component is only abstract definition in this V-Pattern. It could be a huge application landscape unit, but also could be a segment of program in software. The importance of this definition is, the component unit should have resource requirement (communication bandwidth, software and hardware), interfaces and interaction with other components. In one visualization diagram, there are different clustering rules, for example hosting, using, or some self defined clustering rules. Rules are indicated with short descriptions in the clustering unit in visualization diagram. Different types of interfaces and communication types are represented with different colors and different symbols, which are explained in the legend.

Important rules for V-Patterns are, that only changed units and their direct connected units are included in the visualization diagram. We use different colors to indicate the changed components. It could be a new component, changed component, replaced component or unchanged but directly connected component. In principle, the new component and replaced component should be indicated to further detailed visualization diagram.

The special resource unit is defined in legends, which could contain different information such as CPU, memory size, MIPS, etc. The resource unit must be assigned to a certain component. The information flow should be indicated with different arrows for different types (synchronous, asynchronous) and with detailed size of information flows.

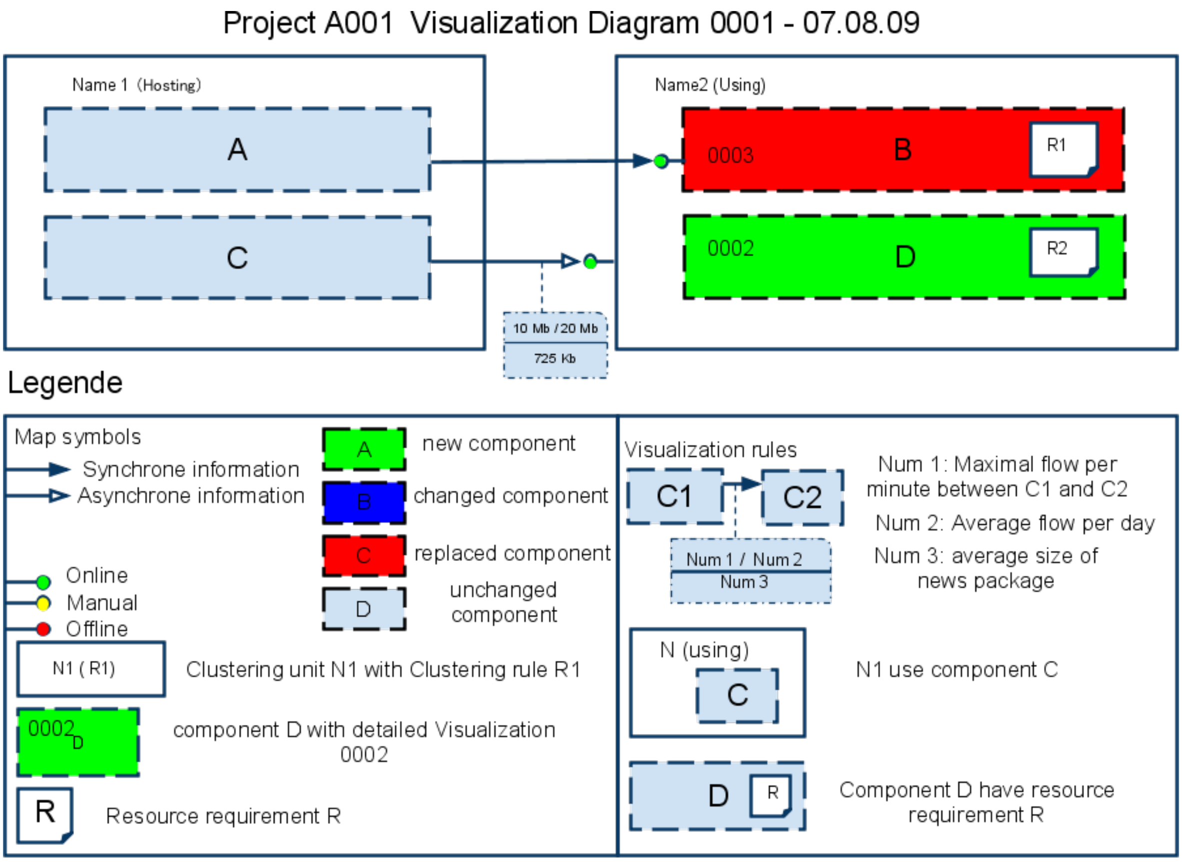

Figure 11.6: Exemplary visualization of ACP- and project-relevant information

11.6 Outlook

In this project, we developed a way to improve the efficiency and effectiveness of ACP based on EA management techniques. In detail, we developed a management method that reflects the ACP (see Section 11.2) and linked the method to two viewpoints that are used to assess architectural relevance (see Section 11.3) and architectural conformance (see Section 11.4) of a project. The feedback of the using company showed the applicability of our approach in practice and highlighted the two main benefits of our solution. First, it helps to speed up the decision process for the ACP, and makes the decision based on the quantifiable measures. Second, we suggested rules of standardization for visualizing the project's information for the ACP and sketched a suitable viewpoint in Section 11.5. Nevertheless, future improvements could be applied to the solution. These are alluded to subsequently.

Cost and Benefit
Prepared documents about a project help to fell decisions within the ACP. If preparation of such documents is also costly, then the question arises, does the usage of documents bring profit for the whole company or does it bring only additional cost from the aspect of the entire IT department. It could support analyze and decisions of the ACP, but it may meanwhile increase the cost for the project team. How to balance cost and profit for our solution, is the next important question.

Global repository for whole company
A company wide repository with architectural information would help to analyze the whole IT Infrastructure and guide decisions for the company's IT. Meanwhile it could also reduce costs and complexity of communication between different IT departments. Furthermore, this repository could be used to

store, exchange, and provide information for the whole Enterprise Architecture Management. The Project Portfolio Management as well as the IT Infrastructure Management could benefit from such an improvement.

Bibliography

[AG09] AG, A. D.:. *Zahlen, Daten Fakten zur Allianz Deutschland AG.* http://www.allianzdeutschland.de/unternehmen/zahlen_daten_fakten/index-2009.html (cited 2009-08-25). 2009.

[Ba21] Barnard, F. R.:. *in Printers' Ink.* 1921.

[Bu08] Buckl, S. et al.:. *Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008).* Technical report. Chair for Informatics 19 (sebis), Technische Universität München. Munich, Germany. 2008.

[Ul09] Ulm, U.:. *Wirtschaftswissenschaften und Mathematik Kongress 2009 der Universität Ulm.* http://www.wima-kongress.de/WiMa/Firmenstaende/Allianz.html (cited 2009-08-25). 2009.

CHAPTER 12

Summary

Retrospective, different advantages and disadvantages of our approach to teach EA management in a lecture accompanied by practical mini projects at industry partners become apparent. This chapter summarizes the lessons-learned and experiences taking three different perspectives. The first perspective (*advisor*) reflects our direct experience with the students and industry partners during the projects. In contrast, the two other perspectives (*industry partners* and *students*) are not based on original experiences but have been gathered by us through questionnaires from the corresponding groups. Due to the limited number of industry partners, we abstain from empirical analyses of the questionnaire data, but provide qualitative findings in the subsequent sections.

12.1 Lessons-learned: the advisor's perspective

Conducting the mini projects for the first time, we came across some difficulties in their execution that an advisor willing to implement this kind of teaching program himself should be aware of. Firstly, the importance of clearly shaping the EA management problems that the students solve prior to the project start should not be underestimated. Here, the advisors and the corresponding industry partners should invest sufficient time to come to a concise and clear proposition that reflects the time constraints of the mini project. Some re-shaping might necessarily happen during the execution of the mini project, nevertheless, a clear starting point is of high importance for a successful and timely execution of the projects.

Secondly, at the beginning of the corresponding projects, the advisors have to provide sufficient input and guidance in coordinating meetings between partners and students. Further, quality assurance of students' presentations is of high importance, due to the intrinsically highly "political" nature of EA management. Here, strong ties between the advisors and the representatives of the industry partner can be beneficial to avoid political pitfalls and to ease the introduction of the students into the corporate environment of the partnering company.

A third important aspect centers around the notion of the deliverable, i.e. the work product that the student teams should create. Here an intricate tension has to be resolved, namely the tension between the two types of "customers" of the student projects. On the one hand, the partnering company wants to get a real-world EA management problem solved; on the other hand, the advisor wants to see the students applying well-grounded techniques and methods. Demanding a project report is in this respect one way to resolve the tension, as the report mirrors the advisor's interest and calls for a structured, well-grounded project procedure. Frequent meetings with the students, i.e. once every week, may further be used to avoid that the mini project teams drown in the clutter of project execution, forgetting on the techniques and methods that they should get acquainted to. This especially applies, as the solution of the company's EA management problems might call for additional methods that had not been taught in the preceding lecture, e.g. methods for constructing metrics or conducting structured interviews. In this respect, vivid exchange between students and advisors may be beneficial to provide further information to the project teams in time.

Finally, the utilization of patterns showed to be highly beneficial in two ways: At first, patterns provide an easy entry into the topic of EA management and describe solutions that can be helpful for solving the problems in the partnering enterprise. Secondly, the pattern structure, especially the triptych *problem*, *solution*, and *consequences* helps the students to structure their mindset, when working at the industry partner. In this respect, advising the students to comply to the pattern structure for their final report may, in advance, positively influence their way to approach their specific problem. Admittedly, some exceptions may apply in this respect. Specific problems, especially ones concerned with the establishment of an EA management function or parts thereof are not easy to document as patterns. Here, student and advisor should devise an alternate structuring principle tailored to the application topic but also enforcing rigor in method application.

12.2 Lessons-learned: the industry partners' perspective

During the students' presentations and in discussions with the advisors of the lecture, the majority of the industry partners showed high interest in the mini projects. The partners especially emphasized that the projects were useful in two ways: On the one hand, the students were able to address and partially solve labor intensive EA management-related problems in the participating companies; on the other hand, the practitioners saw the projects as vehicle to prototypically implement the pattern-based approach to EA management in their company. In this respect, the industry partners' general resonance on the mini projects was strongly positive, further reflected by the willingness of a majority of partners to host more student mini projects in the future. Also the general assessment of the projects' utility was positive.

The majority of the industry partners gave a positive image of their students, which presented themselves as motivated and well-prepared mini project teams. This especially applies to the presentations given at the beginning and the end of the corresponding projects. To our surprise, most of the industry partners strongly focused on the presentations of their specific topic, in some cases not taking the time to attend the presentations of students working in a different company. In this respect, the expected cross-fertilization of projects and the exchange of information between the participating industry partners did not reach the level, we would have expected. Nevertheless, some of the partners found the time to make contact and exchange ideas and insights on the topic of EA management. When it comes to the partners' assessment of the students EA management-related knowledge, all partners rated their students *good* or even *very good*. Discussing this result in more detail, two possible reasons for this fairly positive assessment can be though of:

- The lecture provided a comprehensive overview on the topic and the students were further motivated to dive into the lecture's curriculum, as the final exam was conducted halfway between the lecture and the mini projects. Further, the students had chosen their mini project in one of the first weeks of the lecture, rising additional interest in their specific EA management-related topic.

- The pattern-based approach towards EA management provides an easy to use framework for solving typical EA management problems in a company. Due to its intuitive structure, the students could quickly find and select the relevant concerns and start applying the provided solutions quickly.

One major drawback of the current setting for the mini project execution was risen by almost all participating partners. The intended duration of 2-3 months was considered far to short. This stands in accordance with our own experience as well as with the students' remarks (see below). While the setting can be ascribed to the general timeline of a graduate program at the university, we would advocate for a less rigid timeline, extending the projects' duration to at least 6 months.

12.3 Lessons-learned: the students' perspective

Many students appreciated the idea of the mini projects as an opportunity to get hands on a topic that presents itself as highly theoretic and abstract in the lecture. In this respect, the practical part of the lecture received a better overall rating than the theoretic part thereof. This is to some extent ascribed to the schedule of the overall program. To free the last two months of the term for the mini projects a full-term lecture was condensed to two months, leading to fairly long lecture sessions. The raised interest for the topic EA management is further reflected by ongoing student activities, i.e. by the number of students that in the subsequent term enrolled to research projects on the topic at our chair.

Contrasting the positive feedback, the students also raised some ideas for improvement in future terms. Firstly, the students noted the high workload related to the mini projects which is only partially reflected by the received credits. Secondly, the duration of the mini projects was subject to criticism from two different directions. While some students moaned that the compilation of the project's reports interferes with the period of exams, some other students noted that two months are far to short to actually solve an EA management problem. In this respect, the students raised the question of an alternate mode of work, e.g. a combination of lecture and mini project spanning two terms with some spare time to allow for the preparation on other exams. To allow for such extended mini project program, an adaptation to the curriculum would be necessary.

CHAPTER 13

Outlook

All in all, we can finally conclude that the offered mini projects were beneficial for the three parties: the industry partners, the advisors and particularly the master students. We as the advisors learned what it means to practically teach EA management in a larger scale tackling with several different problems at nine partners from industry. In addition, graduates were given the possibility to successfully use the acquired EA management knowledge in practice. Finally, the industry partners positively mentioned the exchange with academia as well as the specific work results of the student teams, which turned out to be helpful for most partnering companies. Additionally, we noticed an enduring positive motivation on both sides, students as well as companies; concludingly, the main goal of teaching EA management to graduate students by using real-world cases was accomplished.

Needless to say, that there is still potential for improvement considering similar courses in the future. As discussed in the Chapter 12, the overall time frame has to be revised in order to allow students to work longer on their assigned mini project, thereby, also taking into account that the lecture we offer is not the only one they have to cope with. For the next edition of the lecture, we plan to teach and train students more extensively in terms of conceptual modeling, which we identified as a major shortcoming during the project work. Therefore, we intend to set up small exercises during the lecture, aiming at equipping the mini project participants with the capability to create suitable and correct conceptual models of the EA. Besides, the exercises may ease the situation of a block lecture through the active involvement of students.

For the upcoming summer term in 2010 we plan to give students one more time the opportunity to gain knowledge in the context of EA management both from a theoretical and practical perspective. We are well aware of the extra workload this cooperation with industry implies to us and to the students, but we also see the advantages and benefits of the approach. Testing and applying theoretical knowledge by means of implementation in real-world examples is in our opinion the best way to teach EA management.

List of Figures

List of Tables