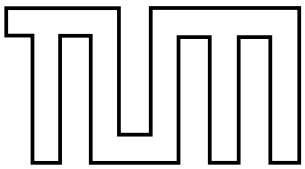
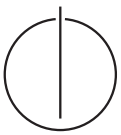


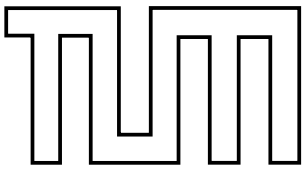
Fakultät für Informatik
Technische Universität München

Bachelor's Thesis in Wirtschaftsinformatik

Models, methodologies, and visualizations for compliance management

Thomas Dierl





Fakultät für Informatik
Technische Universität München

Bachelor's Thesis in Wirtschaftsinformatik

Models, methodologies, and visualizations for compliance management

Modelle, Methoden und Sichten für das Compliance Management

Thomas Dierl

Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Sabine Buckl
Date of submission:	15.10.2008

I assure the single handed composition of this bachelors thesis only supported by declared resources.

München, den 15. October 2008

Thomas Dierl

Abstract

Compliance is getting more and more in the focus of large and medium-sized companies. Thereby, compliance originates from different sources such as laws, corporates, or technical sources. These constraints have a huge impact on the application landscapes of modern companies, as not only the current landscape but also future landscapes are affected. Thus issues of compliance have to be monitored and managed. Therefore, reports and visualizations are used to illustrate this information in order to support efficient communication between stakeholders. Furthermore, methodologies, which apply guidance how to reach the target landscape have to be introduced.

This thesis was conducted in cooperation with a large German insurance company and addresses the topics of compliance management within the existing structures and methods established in a globally acting company. Based on the Enterprise Architecture Management Pattern Catalog (EAMPC), this work evaluated how the EAMPC could be used to support the enhancement of a compliance management. The EAMPC contains best practices for EA management gathered during an extensive survey with 30 companies. Thereby, EAMPC defines not only visualizations but also information models and methodologies.

This work showed how patterns already existing in the EAMPC can be used to establish compliance management, and developed further patterns specialized on this topic. The development of these patterns was done using an evolutionary approach in which a draft of the model was designed, discussed by the appropriate stakeholders, and redesigned until a final stage was reached. This approach was used due to the requirements of the participating company. As a result of this work, the developed patterns, were integrated in the EAMPC in order to close the experienced lack of support regarding compliance management.

Contents

CHAPTER 1

Introduction

The topic enterprise architecture(EA) management is getting more and more into the focus of practitioners and researchers. [LW04] discovered an increasing number of published papers by diverse communities in the the last years 2003 and 2004. Since then the interest in this topic has grown even further. Modern enterprises have to cope with historically grown and therefore very complex application landscapes. Because of the need for flexibility in the business and the increasing importance of information technology (IT) the companies can hardly afford such high complex application landscapes. EA management promises to master the complexity of modern application landscapes by homogenization and furthermore improves the alignment of business and IT. Although, the importance of the topic is unquestioned, there is no common understanding of either the term EA or EA management. Several different definitions and approaches have been developed by practitioners [Th08, Ke07, De06, En08], public utilities [De08, Of08], academic researchers [Za08, HW05, Wi07, Jo05], and tool vendors [ID06, Ma08]. They have in common that all of them see EA management as a complex task, but a task with several different aspects.

The work presented in this thesis is based on the foundations of the Enterprise Architecture Management Pattern Catalog (EAMPC)[1] [Bu08b] and was executed in a large German insurance company. Section 1.1 provides the motivation and goals of the thesis and Section 1.2 provides an overview about the single steps executed during the conduction of the thesis.

[1] Therefore, this work is based on the research on System Cartography at the chair for Software Engineering for Business Information Systems (sebis) of Prof. Dr. Florian Matthes at the Technische Universität München.

1.1. Motivation

As alluded to above the topic EA management is considered to be important by more companies. Because of the hard competition in the market especially big companies need to be more flexible than in history. The IT is an important asset as means to reach that flexibility.

In most companies the application landscape is historically grown and therefore very inhomogeneous. This leads to complex system dependencies and a high maintainance effort. EA management is trying to reduce this complexity by identifying redundancies, out of this potentials initiatives can be derived. This identification can e. g. be reached by using a compliance management approach. A further goal of EA management is to better align the IT with the business needs. This means, that business needs are implemented in new initiatives. Another source for new initiatives could be the need for technological changes. This could be the end of support of a used technology for example. An important step of EA management is the alignment of all these initiatives to reduce the effort for implementing them.

An important part of EA management is the integration of all the needed processes. Therefore, this work was motivated by the aim of a German insurance company to integrate the established EA management approach and the existing compliance management approach. Another goal of this work is to enhance the compliance management approach at the German insurance company by using the EAMPC. Special attention should be given to the usability, and the extension of the EAMPC.

1.2. Structure of the thesis

The EAMPC proposes to introduce an organization-specific EA Management through integrating EAM patterns [Bu08b]. Thereby in a first step the relevant concerns have to be selected out of the list available in the EAMPC. Every selected concern includes references to one or ore M–Patterns helping to address the concern. Furthermore the M–Pattern includes references to V–Patterns used by the M–Pattern and the V–Pattern again includes references to used I–Patterns. In a last step all used patterns should be integrated to an organization-specific EA management.

As visualized in Figure 1.1 this work is not structured as described in the EAMPC but similar to a software engineering process. This proceeding was chosen because of the existing information, knowledge, and temporal restrictions at the German insurance company. The following tasks were executed.

The *analysis phase* is used to gain insights into related work and already existing information. Therefore, the following three main tasks describe the analysis phase:

- Tool Proof of Concept: Because the German insurance company already did an evaluation of tools, the existing proof of concept (PoC) and the methodology behind the PoC and the tool was analyzed.
- Analysis of existing data and models: As there already exists a database with information about the applications in the application landscape, the underlying information model thereof and the existing data is reviewed.
- Analysis of EAM Pattern Catalog: The EAMPC as a foundation of this work is analyzed and interesting concerns are identified.

The *design phase* of the software engineering process is used to develop the information model as basis for the remainders of this work. This procedure was chosen because of the temporal limitation of this project to get the EAM Tool[2] usable in short time and knowledge of Unified Modeling Language (UML) [OM05] was wide spread under the involved stakeholders[3]. These steps and the according models are further explained in Chapter 4.

- Create/change draft of information model: In the first iteration a draft of the information model is created and changed within several iterations.
- Discussion of information model: After the creation of the model and in later iterations changes of the draft are discussed with several persons of the German insurance company.

As next step the information model created in the design phase was implemented in the used EAM Tool and according to this M-Patterns are developed. Further information about the *implementation phase* can be found in Chapter 5. This phase is divided into the following parts.

- Create mapping: The first step is to look for a mapping from the already existing design to the concepts available in EAM Tool.
- Import data into the Tool: After the creation of the mapping it is possible to import the already gathered data into the EAM Tool.
- Define Viewpoints: Then the visualizations of the information was created by defining Viewpoints. This step was repeated in an iterative process.
- Generate Viewpoints: To get a better discussion base the defined Viewpoints have been generated out of the existing data.
- Discuss Viewpoints: A discussion with several stakeholders of the German insurance company about the generated model starts. The findings are worked into the definitions of the Viewpoints for the next iteration.
- Create M-Patterns: Simultaneous to the import of data and the definition of the Viewpoints the appropriate M-Pattern are created.

[2] ARIS IT Architect

[3] Stakeholders are all persons or institutions which are affected by this work or the result.

After the integration of the findings into the EAM Tool a test and validation cycle starts. Within this step the usage of the already customized EAM Tool is observed. Thereby a special focus is directed on the question *Is the tool used according to the definition of the M-Pattern?*.

Out of this findings a resume is drawn and the M-Patterns will get adapted to the observed real usage. As a last step all of the findings above are distilled and integrated into the EAMPC.

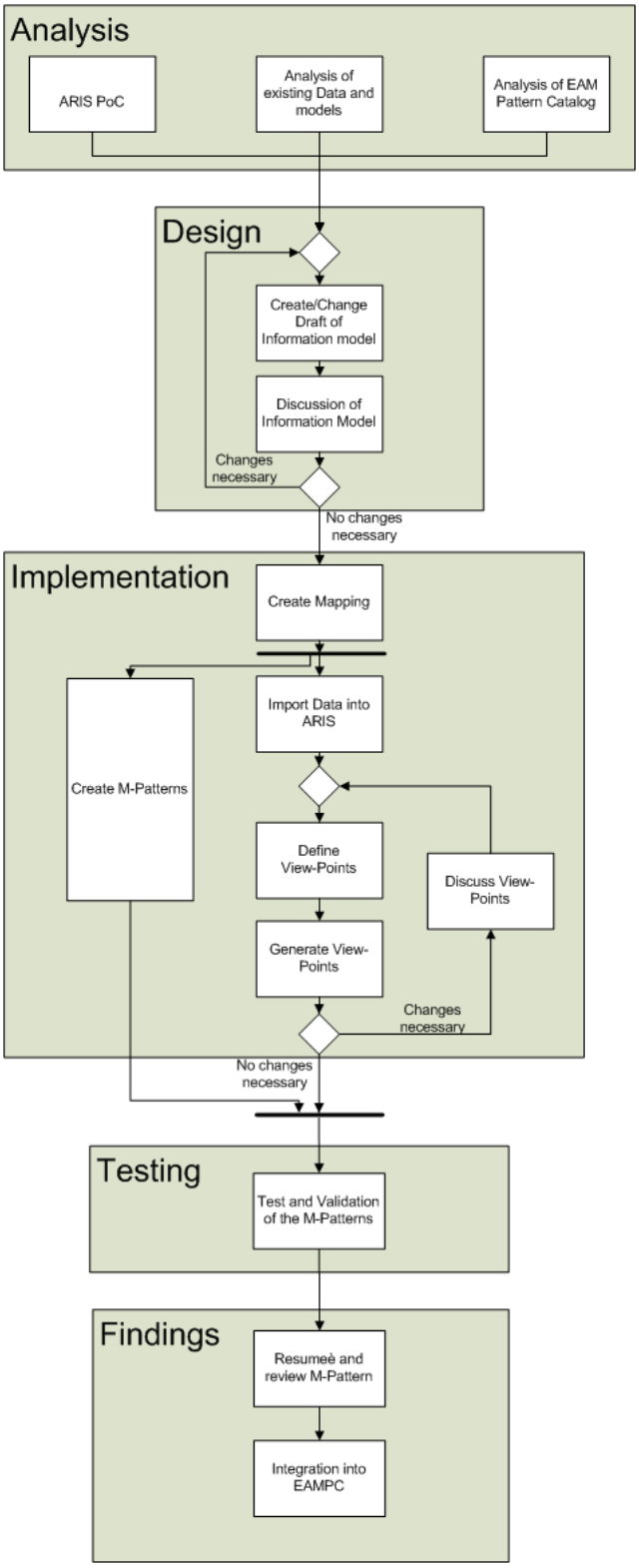


Figure 1.1.: Visualization showing the proceeding of the thesis

CHAPTER 2

Related Work

This chapter provides the foundation for the work by introducing relevant research areas and providing definitions for the mos important terms used throughout the paper. The most important research areas of this are the EA management and compliance management. Section 2.1 gives an introduction to the topic EA Management and Section 2.2 introduces the term compliance management. Supplementary, Section 2.3 sketches the information model, which served as foundation for the developed compliance management approach.

2.1. Overview on EA Management

The main goal of this work is to integrate the already existing EA management with the compliance management. For this reason it is necessary to have a closer look on the term EA management. According to [Ma08] EA management is defined as follows:

> *Enterprise architecture management* is a continuous and iterative process controlling and improving the existing and planned IT support for an organization. The process not only considers the information technology (IT) of the enterprise, also business processes, business goals, strategies etc. are considered in order to build a holistic and integrated view on the enterprise.
>
> Goal is a common vision regarding the status quo of business and IT as well as of opportunities and problems arising from theses fields, used as a basis for a continually aligned steering of IT and business.

This definition suggests that EA management is not just gathering the information about the application landscape but that the whole evolution of the application landscape is managed by EA management. Furthermore, in the above definition, EA management is defined as a process that considers more than is needed by compliance management. In Figure 2.1 this is visualized by the different layers and cross functions of EA management. Compliance management is mainly contained in the cross function *Architecture & Patterns*, but affects nearly all of the other layers and cross functions. This more holistic view enables several opportunities to the company, as for example the business and IT alignment as described by [Ro03].

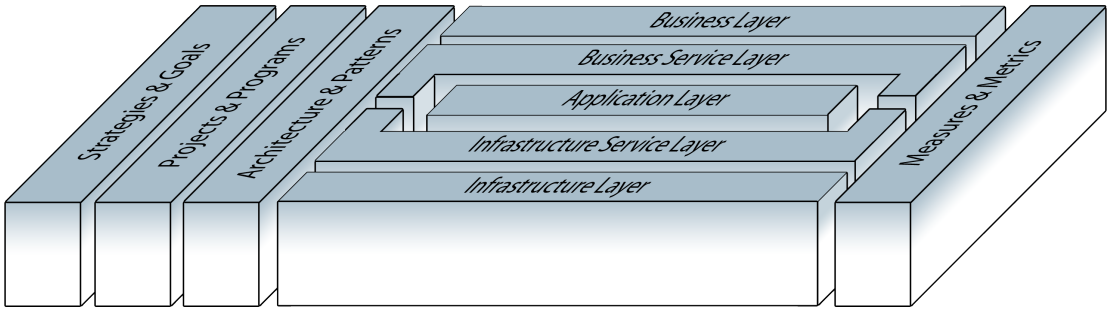


Figure 2.1.: Layers and cross functions of EA management Source:[Ma08]

Thereby, the EA management process needs to be integrated into the various processes changing the EA of a company in order to receive input from related processes and provide information.

Fig. 2.2 shows the different processes, which need to be integrated to reach the manifold goals of EA management [Wi07].

First of all the process *demand management* identifies the demands, which come from different stakeholders, e. g. application owners, business process owners, or developers. These different demands are then grouped by similarity and the affected architecture elements are identified. Afterwards the calls for action are identified and the IT initiatives are set up.

The identified demands constitute the beginning of the IT project lifecycle which the related processes influence. This impact can be summarized along the further steps each IT project runs through.

When the IT Projects are defined they run through the *IT Project Lifecycle*. As a first step the initiatives needed to reach the project goals are defined and described.

The *Identify calls for action* and the *Define & describe Initiative* are supported by the process *Strategies and Goals Management* which aligns the defined demands with already defined strategies and goals.

Subsequently the initiatives have to be fine-tuned and planned, the dependencies between the initiatives have to be observed. As a next step the initiatives have to be prioritized and approved, thereby it is possible that several initiatives are declined and the fine-tuning has to be adapted.

As visualized in Figure 2.2 the processes *Fine-tune and plan Initiative* and *Prioritize & approve Initiative* are aided by the *Projectportfolio Management* which keeps an overview about all running projects and their initiatives.

Of course after the approval the projects have to be implemented and this implementation has to be controlled and afterwards the operation of the produced systems can start and the initiative can be closed. In this part of the IT Project Lifecycle the projects are synchronized in the support process *Synchronization Management*.

While the *Prioritize & approve initiative*, *Implement & control initiative* the support process *IT Architecture Management* controls, that the architecture of the projects fits to the given rules and existing systems. As the compliance management partially aims to set rules for the applications in the landscape this process, which control these rules, is part of the compliance management.

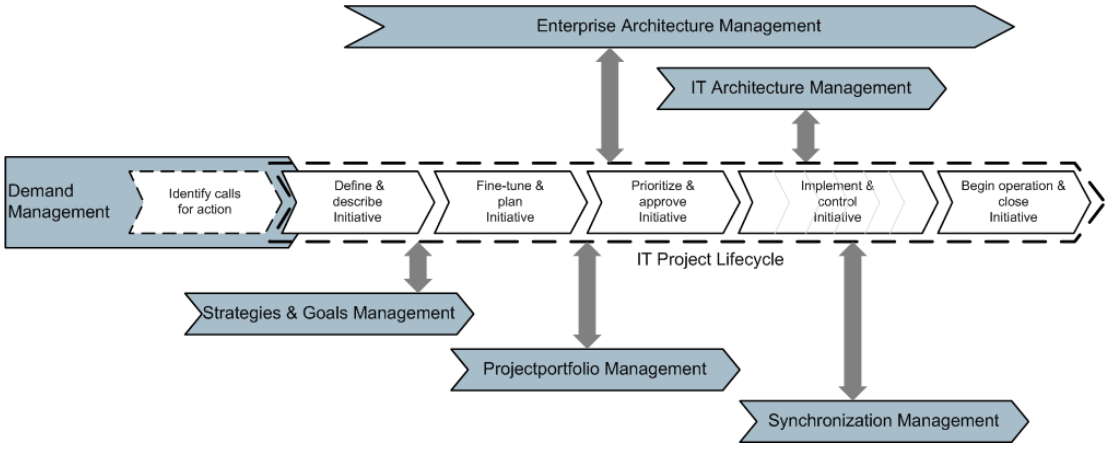


Figure 2.2.: Integrated related management areas [Wi07]

As already mentioned above EA management is a versatile approach. To introduce such an approach not only the technical aspects need to be considered but many other things, such as roles, rights, boards, and process documentations, have to be kept in mind. These elements are vital for the success of an EA management endeavor. This means that the establishment of an EA management is a complex task. To support this endeavor Software Engineering for Business Information Systems (sebis) created the EAMPC [Bu08b] which provides different kinds of patterns and "guidance needed to systematically establish EA management in a step-wise fashion within a given enterprise."[Bu08b]

2.1.1. Enterprise Architecture Management Pattern Catalog

The EAMPC identifies different types of patterns and the dependencies between them. The three patterns are *methodology*, *viewpoint*, and *information model*. All of them have been considered as useful and relevant by experienced practitioners or supported by literature.

Methodologies define activities, which are required to achieve a certain concern.

Viewpoints give hints about visualizations, which help to perform the *methodologies*.

Information model shows, which information is required to generate the associated *viewpoints*.

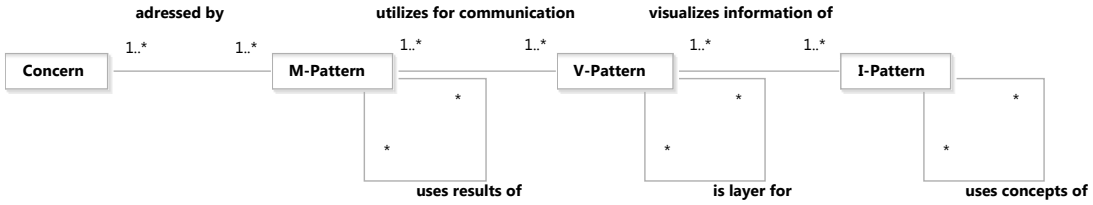


Figure 2.3.: UML class diagram showing the relationships between Concerns, M–Patterns, V–Patterns, and I-Patterns [Bu08b]

The EAMPC describes an approach to introduce an organization-specific EA management. Figure 2.3 shows the relationships between the concerns, M–Patterns, V–Patterns, and I–Patterns. Concerns in this context define *Which goal is to be achieved for which stakeholder?*. Furthermore it illustrates that the EAMPC starts by identifying the appropriate concerns. From this starting point it is possible to traverse along the dependencies between the patterns and thereby find all possibly interesting patterns. The received information can be used as inspiration basis for the development of existing EA management approach or for academic research. The interesting point for research is the easy extensibility of the EAMPC, because newly created patterns can be added with little effort.

For the creation of an EA management it is necessary to integrate all found patterns of the same kind. Because of this the EAMPC defines how to integrate the different kinds of patterns with each other

. According to [Bu08a] the integration of the M–Patterns can be achieved by the use of a process model, which defines the steps needed for the EA management. Of course the assumptions used for the different patterns have to be considered to avoid inconsistencies after the integration.

The EAMPC V–Patterns rely on the so called layering principle [Er06]. This idea allows to see the visualizations as patterns which can be integrated with each other by just think of them as different layers as in cartography. This allows to integrate a pattern which visualizes the applications with a pattern that shows performance indicators of application systems into a single visualizations showing both. [Bu08a]

I-Patterns are represented as UML class diagrams and so they can easily be integrated by just integrating the UML class diagrams. This can be done by identifying equals concepts and integrate the attributes and relationships into one class representing both concepts.

2.1.2. Gathering information

The information gathering is vital for an EA management approach. The gathering of information brings up different aspects, which have to be regarded. On the one hand the information defined in the information model must be available for gathering with an adequate effort. If the information retrieval is to expensive it is nearly impossible to keep the information up to date and therefore, makes it impossible to decide on the basis of this information.

Figure 2.4 illustrates that the information for EA management originates from different sources. Thereby, a lot of information is already available in different already existing software systems but has to be filtered and processed. This may happen automatically to maintain the information.

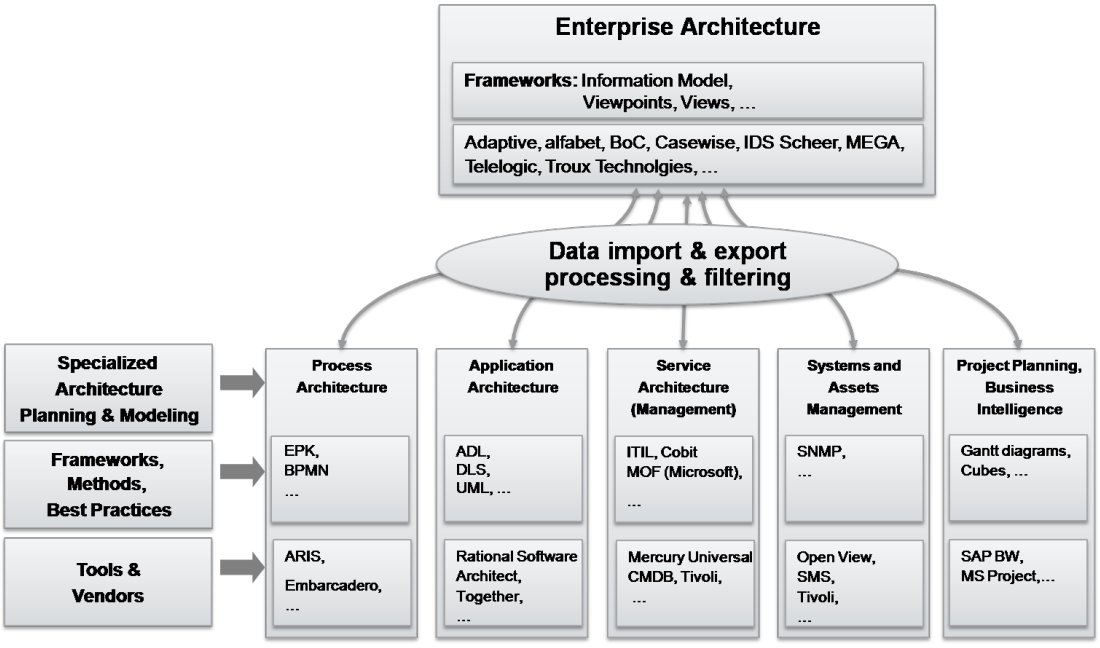


Figure 2.4.: Information filtering & processing [se05]

2.2. Introduction to compliance management

With respect to endeavors of regulations, e. g. Basel II [Ba06], Sarbanes Oxley Act [Se02] or Solvency II [Eu07] to observe given rules, compliance management has been a growing concern of companies all over the world in the last couple of years. Because of the big amount of such rules and the high complexity of the implementation and monitoring of these rules, software is needed to support the observation and adherence of them. The next section identifies the central aspects of the term compliance. The following Section 2.2.1 presents different definitions of compliance. Thereby, it sketches the definitions of legal, corporate, and technical compliance. Section 2.2.2 explains how different regulations in compliance management can be structured according to the source their are coming from and Section 2.2.3 explains how these sources are interrelated. Final Section 2.2.4 explicates how the term compliance management is used throughout this work.

2.2.1. What is compliance

According to [Co08] "compliance is about the fulfillment, equivalence, and/or conformance with national laws as well as with rules and specifications, with principles (ethical and moral) and procedures as well as with standards (e.g. ISO) and conventions, which were clearly defined. The fulfillment of the compliance can be based both on obligation (e.g. by laws) and on voluntariness (e.g. observance of standards)." Goal of the endeavors is risk reduction, improvement in efficiency, and effectiveness. These functions usually affect different departments within a company e. g. legal departments, controlling, internal revision, human resources, and the IT department. These leads to the need of coordination between these activities.

Another definition can be found in [Di08], there the term compliance refers to

1. the act of conforming, acquiescing, or yielding.
2. a tendency to yield readily to others, especially in a weak and subservient way.
3. conformity; accordance: in compliance with orders.
4. cooperation or obedience: Compliance with the law is expected of all.

The only thing both definitions have in common is the observation of guidelines, therefore this is the central point of compliance. Additionally, the first definition gives hints, about the origin of these guidelines, e.g. laws or observance of standards.

2.2.2. Types of compliance management

To get a holistic view on the field of compliance management it helps to structure it. There are different sources for compliance management, therefor one possibility to structure compliance management is to categorize it into the possible sources of regulations. Such sources can be e. g. legal, corporate, or IT. All types of compliance have an impact on the software systems of an enterprise, e. g. via the support of special reports.

2.2.2.1. Legal compliance

In the last few years several events, as e. g. Enron or WorldCom, forced the legislators to introduce laws to increase the liability of companies. Among these laws are e. g. the Sarbanes Oxley Act (SOX) [Se02] for companies, which are listed at US stock exchanges, Basel II [Ba06] for banks, or Solvency II [Eu07] for insurance companies.

The main focus of the 2002 in the USA introduced SOX is on financial reporting and tax returns. It was created in response to the corporate and accounting scandals. It increases the reliability of the reported information to the shareholder and raised the criminal penalties for white-collar crimes. It is a federal law in the US, but does not only affect the companies in the USA but also foreign companies that are listed on any US stock exchange. The main impact of SOX are increasing costs for audits and internal controls and furthermore a higher responsibility for CEOs and CFOs. The biggest impact on IT is contained in Section 404 of SOX, this Section contains responsibility for an assessment of the internal control including e. g. fraud risk assessment.

Basel II was created by the Basel Committee on Banking Supervision in 2004 with the purpose to assure that banks have appropriate equity bases and to create uniform competitive conditions for granting and trading of loans. Therefore, it is based on three pillars, which are Regulatory Capital, Supervisory Review, and Market Disclosure. In 2006 the EU adopted a directive with which Basel II applies to all banks and financial service providers in the EU (European Union) from the first of January 2007. Basel II aims at ensuring that capital allocation is more risk sensitive, separating operational risk from credit risk quantifying both, and attempting to align economic and regulatory capital more closely to reduce the scope for regulatory arbitrage. The most cost intensive is the rise of the equity base of a bank, the appropriate equity base is supervised by the national bank supervision.

In July 2007 the EU Commission announced a ground-breaking revision of EU insurance law more well known as Solvency II. It is designed to consumer protection, modernize supervision, deepen market integration, and increase the international competitiveness. [Eu07] In [RMRH06] is stated, that the biggest impact of Solvency II is on data management because a lot of data and reports is needed. This has an effect on EA management, because the used applications have to support the management and reporting of this data. Furthermore, these applications have to be handled and the responsibilities have to be clearly distributed.

2.2.2.2. Corporate compliance

In addition to the laws presented above, governments and industry have developed best practices which should be considered. There exist several of such best practices on different levels. One of them is the German Corporate Governance Code [Re08].

Although the German Corporate Governance Code was developed by the Federal Republic of Germany, it is not a law but is considered to be a behavior-recommendation to administration and control of a company. Therefore, the German Corporate Governance Code has to be reported and is important for the external communications of the company. The Code contains regulations for the board of directors, the executive board, and the cooperation of them. The companies have to consider the Code in many situations as e. g. when new executives or directors are introduced.

2.2.2.3. Technical Compliance

But the compliance influence may not come only from outside the company, also the company itself may establish guidelines for compliance management. Often the internal guidelines concern the information technology. One possibility to define guidelines for IT are standards. Standards can be implemented in several ways as e. g. product standards. Generally speaking standards avoid a magnitude of different objects which are used to reach the same goal. Such a magnitude of objects create a high amount of complexity because they have to be supplied and maintained. A standard prohibits such a magnitude a reduces complexity by using only one type of objects to reach the goal. The special case of product standards uses a certain or a small set of products for one purpose. Other possibilities as architectural patterns or architectural standards are described in detail in Section 2.3.

2.2.3. Relations between the types of compliance management

The different kinds of compliance as described above cannot be seen independent from each other, therefore, the dependencies between these types have to be considered. Additionally, the standards may depend on given laws or codes or circumstances, where the software has to be compliant to given standards.

2.2.4. Further usage of the term compliance management

Compliance management as used in this work refers to technical compliance. The major task of technical compliance is to support migration to a standard compliant technology landscape, and the creation of reports about the standardization degree for the organizational entity. Exemplary reports are “standard conformance”, “used applications”, “running

infrastructure". This enables better reuse options for already developed applications and libraries and can reduce costs of developing/adapting new applications. Furthermore, the "standard conformance" report helps to create a homogeneous application landscape by identifying application which are not compliant to any standard which then can be changed. The report about the "used application" can be used to get an overview about the existing application and maybe reuse one of them before developing a new one for the same purpose.

2.3. Information model

Following further options for IT compliance are detailed. Among them are architectural solutions which have been used for the implementation of this work.

The most simple option for IT compliance are product standards, as the name refers to, they define products, as e. g. Microsoft Windows XP Professional, MySQL 5.0, or Oracle 10g, which should be used during the development of a new or the change of an existing system. Such standards can be assigned to different abstract technologies as for example operating systems, frameworks, programming languages, or database management systems. The advantage of this possibility is the very easy implementation, it can be quickly introduced because it only needs one source where all existing standards are listed. Also the definition of new standards can easily be done, because only the existing standards have to be reviewed that no discrepancies exist. Although it is advisable to have a look at the systems already in use to avoid major migration efforts.

Architectural solutions consider, contrary to product standards, a complete stack of products, e. g. Java Applicationserver Stack which consists exemplary out of CentOS 5.2, JBoss 4.0.2, Oracle 10g, and Java 1.6. Therefore, the possible combinations of products in different abstract technologies are limited to the ones defined, this leads to a huge reduction of complexity compared to pure product standards, because of the decreased amount of interfaces between the systems. Figure 2.5 describes a possible information model for architectural solutions.

The following list describes the terms used in Figure 2.5 in detail:

- ArchitecturalBlueprint: A description of a software architecture (e.g. a client-server architecture), using so-called AbstractTechnologies as components.
- AbstractTechnology: A class of technologies offering similar, or even standardized functionalities. Examples are *Webserver* (with specific technologies then being Apache 2.0.53 or IIS 6.0) or *Database Management System (DBMS)* (with specific technologies then being DB2 6.0 or Oracle 9i).
- AbstractUsage: The usage of an AbstractTechnology in an ArchitecturalBlueprint.

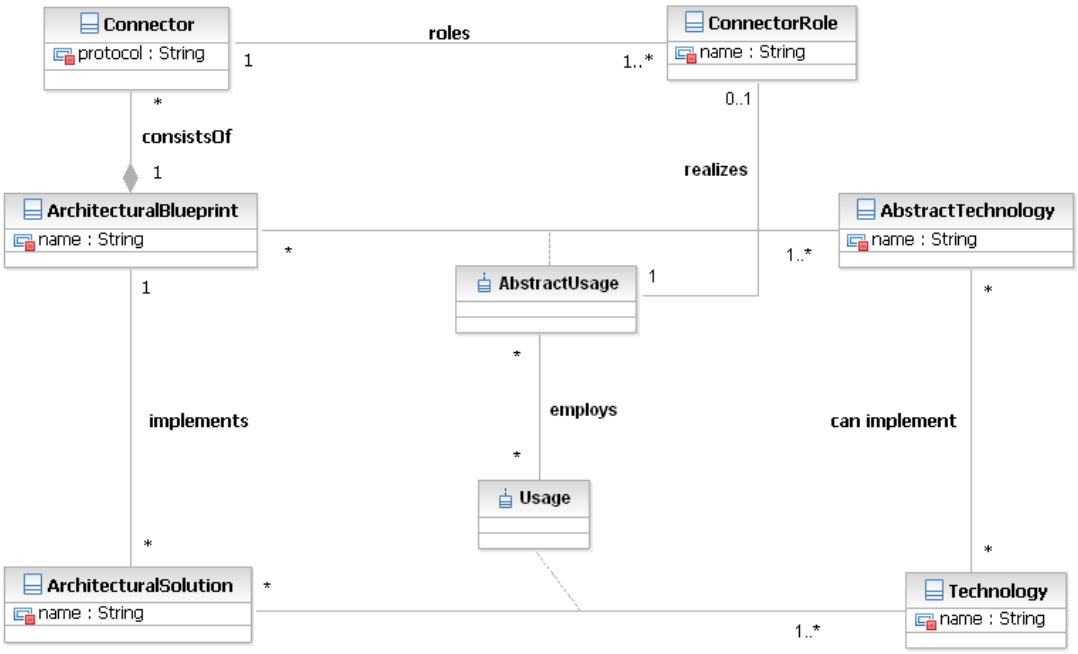


Figure 2.5.: Information model basis

- Connector and ConnectorRole: A Connector is a runtime pathway of interaction between two or more AbstractTechnologies. A ConnectorRole identifies the role taken by the respective AbstractUsage in the interaction.
- ArchitecturalSolution: A concretization of an ArchitecturalBlueprint, created by selecting a specific Technology for each AbstractTechnology of the respective ArchitecturalBlueprint. An architectural solution thus describes a basic architecture for an application, indicating of which components (technologies) it is made up, and how these interact.
- Technology: A specific technology implementing an AbstractTechnology (e.g. Apache 2.0.53, being a *Webserver*, or Oracle 9.2i being a *Database Management System*).
- Usage: Selection of an actual Technology for an AbstractUsage, in the context of a specific ArchitecturalSolution. For the subset of Usages belonging to one ArchitecturalSolution, each AbstractUsage of the corresponding ArchitecturalBlueprint has to be assigned *exactly one* Usage. AbstractUsages not belonging to this ArchitecturalBlueprint may not be referenced.

Both standards and architectural solutions do not define the inner structure of a system, but architectural patterns do so. The term architectural pattern as used in software engineering refers to:

> An architectural pattern expresses a fundamental structural organization schema for software system. It provides a set of predefined subsystems, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them. [Bu96]

Such architectural patterns help developers to understand software systems faster and allow the implementation of applications based on the same foundations. Thereby, making applications comparable. Another advantage of architectural patterns is that some used technologies, as e. g. databases can be reused in several different applications. Architectural patterns create benefit by defining the exact inner and outer structure of a system, which reduces the effort for developers to a minimum. But this limits the developer in its freedom in software development, therefore, the possibility that no appropriate pattern is defined for the given problem rises. Thus either the amount of patterns is enormous to address all possible problems or the systems are potentially suboptimal.

Because of the relative small work necessary to implement architectural solutions and the huge complexity reduction it leads to, the concept used in this work are architectural solutions. The information model used as basis is extracted from [Bu08b].

CHAPTER 3

Requirements of the German insurance company

After the introduction of the terms, now the situation in the German insurance company will be detailed. The requirements for the introduction of compliance management were gathered from the German insurance company. The main goal of the German insurance company was to perform a stepwise introduction from the already existing product standards to the usage of architectural solutions. Additional goals were to automate the generation of reports for management presentations with tool support and introduce an appropriate government process. These goals are detailed in the following sections.

The Section 3.1 describes the current state of compliance management within the German insurance company. In the Section 3.2 the actuators of the German insurance company are further detailed and goals mentioned. In the Section 3.3 the goals are explained in detail.

3.1. As-is state and motivation of German insurance company

The German insurance company is organized in a federated structure with a central organization and several OEs. Throughout a strategic dialog the goals and strategic implications are set globally. IT departments are centrally governed with a high degree of local decision freedom, therefore every OE can do the planning of its application landscape on its own but has to do this in respect to the globally set targets and strategies. This makes some kind of observation necessary to see if these specifications were incorporated.

Before starting the bachelors-thesis the German insurance company has done its compliance management on the level of product standards with an online tool, in which the OEs are able to search the product standards database. As already explained in 2.2.2.3 this has the disadvantage that relations between these standards cannot be defined and therefore no direct connection between several products was maintained and every layer in the stack was seen independent of the others.

The information about the current landscape was maintained in a text based online tool as plain data in a excel like flat data structure. There was only one option to create reports and consolidate information, this was to get an export of the database and create a report with the help of some tool as e. g. Microsoft Excel. Another drawback of the used tool was, that there was no possibility to store information about planned or target landscapes, therefore no comparisons between current and planned or target landscape was possible. Moreover there was no possibility to do any comparison between historical states and the current one.

Reports about the compliance of any OE have been evaluated by a survey once a year, whereby every OE specified to which degree a standards is fulfilled on a numerical scale from one to four. This information is adopted into an excel sheet and aggregated by a person into a small amount of kiviat-diagrams, which were presented to the management.

The driving forces for the transition from product standards to architectural solutions are detailed in the following Section.

3.2. Driving forces

The driving forces for this work were very diverse and were raised by several stakeholders. These driving forces can be divided into business drivers and technical motivation. These are further detailed in the next few paragraphs.

3.2.1. Business drivers

Business motivation summarizes all driving forces that help to increase efficiency. These are for example reduction of costs or reduction of management effort. The compliance management introduced in this work has three main goals, which are explicated subsequently.

The compliance management of the German insurance company should help to reduce costs on a global and on OE level by easy and effective homogenization. This means that the method introduced should provide mechanisms, reports, and visualizations supporting an architect to find possible inhomogeneities or homogenize already existing applications. Homogenization means that these applications are built on the the same technologies, as

e. g. databases, application servers, or middleware. Therefore, homogenization can be achieved by adopt an application to use e. g. another database.

In addition, the compliance management of the German insurance company should reduce the effort for managing the application landscape. Through the homogenization described before it should be less effort to maintain the application landscape and it should be easier to get an overview about the whole application landscape.

Furthermore, the collected information enables focused know-how creation and education of employees, which enables employees to have more in detail know-how for the applications in the landscape.

Another source of cost reduction is the automated generation of reports about compliance. In addition the creation of appropriate new architectural solutions and the maintenance of the already existing solutions has be supported by reports. Another potential source for savings is the reduction of request from architects by publishing the architectural solutions in an online tool.

Another driving force was a more objective comparability between the compliance of different OEs and reduction of know-how for the implementation and the operation of the systems should be reduced with the help of homogenization.

3.2.2. Technical motivation

But not only business reasons led to the migration from product standards to architectural solutions, also technical factors had an impact on the newly introduced system. Therefore, the compliance management of the German insurance company should not only meet the above described business motivation, but also closely related technical aspects. The following paragraphs give an insight view into theses technical aspects.

Strongly related to compliance management but not directly connected is the information about the used and installed applications. In respect to compliance management the questions *Where is an application installed?* and *Who uses an application?* are essential. These questions are needed in the context of impact analysis for decisions taken in compliance management.

Another information that is not directly content of compliance management, but important, is the amount of money invested into an application. This information is essential do decide which applications and related architectural solutions should be maintained further and which ones can be faded out.

For several questions of compliance management as e. g. *Are there architectural solutions for the purposes of a given business process?* an overview about the connections between application and business is important. Such an overview can give for example an overview about *Which application supports which business process?*.

All these questions are used to homogenize the technology platform of the application landscape. This is needed to support the business motivation and create better reuse options and ease further planning in the application landscape.

3.3. To-be state

Out of the as-is situation explained in 3.1 and the driving forces detailed above the to-be state can be defined. Following the most important key points are explained in detail.

The existing online tool is not capable of important features as e. g. report generation or handling of planned or target landscapes. Therefore, it is necessary to introduce a new tool that is capable of such things. This tool should not only handle planned and target landscapes but should also allow the comparison between different historic states to see in which direction an entity developed. The tool evaluation was already done by the German insurance company, therefore the tool was named by it.

Because of the introduction of the new tool the existing information model of the existing tool and the requirements of the introduced driving forces have to be integrated into a new model. In addition the import of the existing data into the new model has to be kept in mind. The newly introduced information model should also be capable of handling the information about the existing product standards for the reason of backwards compatibility. Another important feature is to consider special features such as virtualization.

Summarizing many diverse influences take effect on the implementation of the new tool. The explanation of the development of a new information model satisfying the requirements and integration with the existing information model can be found in chapter 4.

CHAPTER 4

Design: I-Patterns

Based on the requirements of the German insurance company an information model was created in the design phase. Thereby, an evolutionary approach was chosen in which several drafts were created and discussed with the stakeholders to ensure that the final model fits to the needs of the stakeholders. The different processed steps of the design phase are further detailed in the following sections.

The approach to create the information model by identifying interesting concerns and prior to the creation of visualizations was chosen because the persons involved had knowledge about data modeling. Furthermore, this approach allowed to have visible results in the tool very quickly. Besides, a partial information model already existed in the company and knowledge of UML [OM05] was spread along the stakeholders.

The following Section 4.1 explains how the first draft of the information model for compliance management was created. Based on the feedback received from the stakeholders in the first evolutionary cycle Section 4.2 details on the derived changes necessary to fulfill the requirements. Finally, Section 4.3 explicates the changes necessary to utilize the developed model in ARIS IT Architect.

4.1. First draft of the information model

The first draft of the information model was created by flipping through the EAMPC and identifying concerns and connected I-Patterns, which seem to be suitable to address the problems defined in Section 1.1.

Thereby, the following concerns from the EAMPC were identified:

- C-2: Where are architectural blueprints or architectural standards used, and are there areas where those standards are breached?
- C-4: Which technologies, e. g. programming languages, middleware, operating systems, database management systems, used in the application landscape should be replaced, which ones should be kept?
- C-9: Possibilities to reorganize the application landscape in respect to the used technologies should be outlined. Thereby, possible goals are: Reducing licensing costs, reducing maintenance costs, taking into account the support periods of the technology products, etc.
- C-19: Do the 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-46: Which knowledge about specific subjects, e .g. technologies, or programming languages, is currently available in the organization?
- C-50: How is an architectural blueprint/architectural solution made up?
- C-100: Analyze, to what extent individual and standard software is used in the application landscape.

Out of this list, the respective list of I-Patterns (I-47, I-48, I-63, I-66, I-67), which are connected with these concerns has been compiled.

In cooperation with the compliance manager the I-Patterns have been evaluated in order to decide which patterns should be integrated into a first draft of the information model. A second source likely to be integrated was an already existing database-model of the German insurance company.

Beneath the decisions concerning the integration, several general conceptual questions have been identified while developing the first draft of the information model, as e. g. *To which detail should the information been captured?* or *Which attributes should be mandatory?*. This question was evaluated for each attribute separately, mostly the existing guidelines of the existing model was adopted. One of the most difficult questions to answer was *How to reproduce historical data?* and *How should an application and its inner structure be modeled?*.

For the question of historical data several solutions, e. g. discrete historiography with the help of backups, new objects for every change, seamless with the help of an appropriate data model, or no historiography at all have been identified. After discussing the different opportunities, the use of a discrete historiography with the help of backups was chosen.

A further question was how to handle the inner application structure and their deployment. For handling different versions of an application a solution was introduced in the

EAMPC. Pattern I–26 explains how different versions of an application can be handled. Nevertheless, in the context of the German insurance company not only different releases have to be handled also different components of one system have to be modeled. This problem is only partially addressed in the EAMPC, several patterns introduce the concept `ApplicationComponent` but the relationship between the `BusinessApplication` and the `ApplicationComponent` is never detailed. One possible solution to close this gap is to model an application with different components, releases, and deployments and their relationships as illustrated in Figure 4.1. The model assumes, that an `Application` consists of several but at least one `Component` and one `Component` may have one or several `ComponentVersions`, which can be deployed. An application for example may have one component for user management and another for data management. Each of this components may have different `ComponentVersions`, because of several bugs that have been fixed in each module. Furthermore, this application is used several times independent from each other and therefore, it is installed on different servers, this is called deployment. Nevertheless, it should be mentioned that other companies may assume that one application has different releases and these releases use components. This leads to a exchange of `Component` with `Release` and `ComponentRelease` with `Component` in Figure 4.1. In this figure the above detailed relationships are untighten to allow for example `Applications` without `Components` and so on. This was done because while creating all the objects it could possibly be, that an `Application` has for a short time no `Component`. The model from Figure 4.1 can not be found completely in the model for compliance management, because this is not a central question of compliance management.

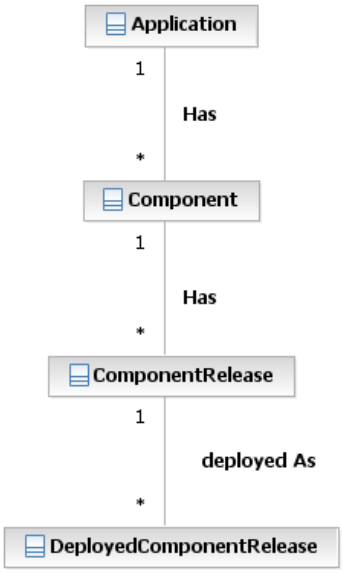


Figure 4.1.: The relationship between applications, components, versions, and deployment

After the initial design phase a first draft of the information model was developed by integrating the different patterns from EAMPC and sources available at the German insurance

company. Different approaches to integrate information model fragments exist, (see e. g. [Bu08a]) a naive approach for instance, would be to identify equal concepts and glue them together. [Bu08a] Although this approach seems to be very simple, there are several problems, which may arise in this context. For example is it complex to deal with different multiplicities of the same concepts, or to define equal concepts with different names. But this could possibly lead to problems with the multiplicities of different models. Therefore, a check if the multiplicities are still correct has to be performed. [Bu08a]

After finalizing the integration the created information model was sent out to the various stakeholders, e. g. compliance manager, business owner, etc. in the different departments. The first draft of the information model, which was sent out, is shown in Figure 4.2 (Due to the reasons of readability the compartments are removed in this figure.).

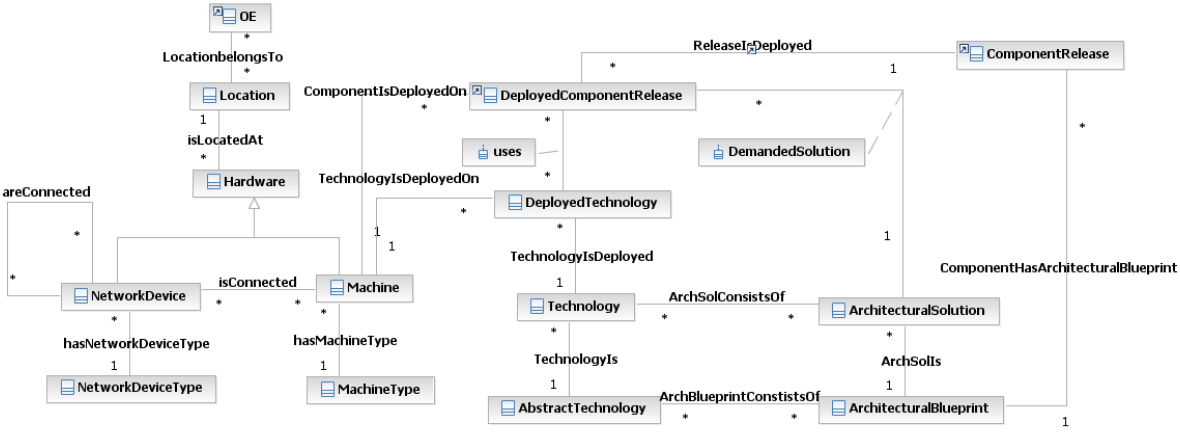


Figure 4.2.: First draft of the information model

4.2. Second draft and final state

The various stakeholders from the different departments reviewed the model in respect to their requirements and provided feedback in order to improve the model according to their needs. The following table shows the comments received from the stakeholders.

From this list of comments several changes have been derived. These were

- *Extension of the model with virtual machines*,
- *Extension of the model with classes for license management*, and
- *Removing network devices and routing tables.*

Stakeholder	Comment
Compliance manager	Virtualization should be considered (including used virtualization software)
Business owner	Services should be included
Compliance manager	Lifecycle information should be included
Compliance manager	License management should be included
Compliance manger, security advisor	network devices and simulation of routing is too complex and should be removed
Business owner	Applications should be grouped according to tasks

Table 4.1.: List of comments to the first draft of the information model

Stakeholder	Comment
Operator	Infrastructure information should be removed completely and instead a Configuration Management Database (CMDB) for infrastructure data should be used

Table 4.2.: List of comments to the second draft of the information model

After the identification of necessary changes the information model was adapted. The resulting model is shown in Figure 4.3. Following the evolutionary approach this model was further discussed in a first iteration with several OEs of the German insurance company. This discussion led to further changes derived from the comments, which are listed below.

The basic idea of integration of data from a Configuration Management Database (CMDB) as a already existing information source was already explained in Section 2.1.2. Because of the already existing connection between ARIS IT Architect and the CMDB the data model to use is predefined by the ARIS IT Architect and therefore not further considered in this model.

Following the evolutionary approach the improved information model was sent out to further OEs to receive more feedback. The received feedback was very positive and no changes to the model were considered to be necessary by the different stakeholders. The final state of the information model is shown in Figure 4.4.

Based on the experience gathered during the development of the final information model the following two new I-Patterns have been identified to extend the EAMPC.

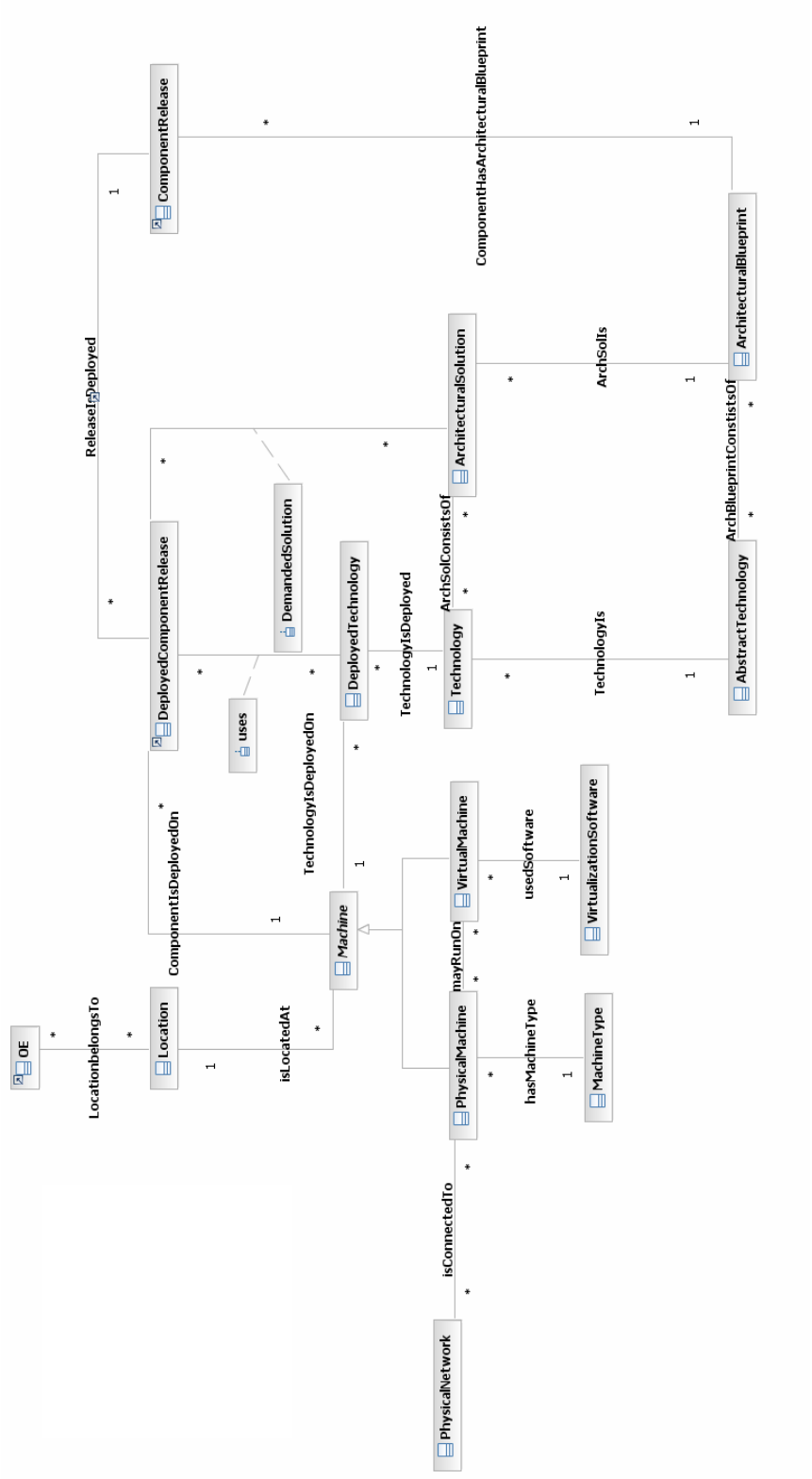


Figure 4.3.: Second draft of the information model

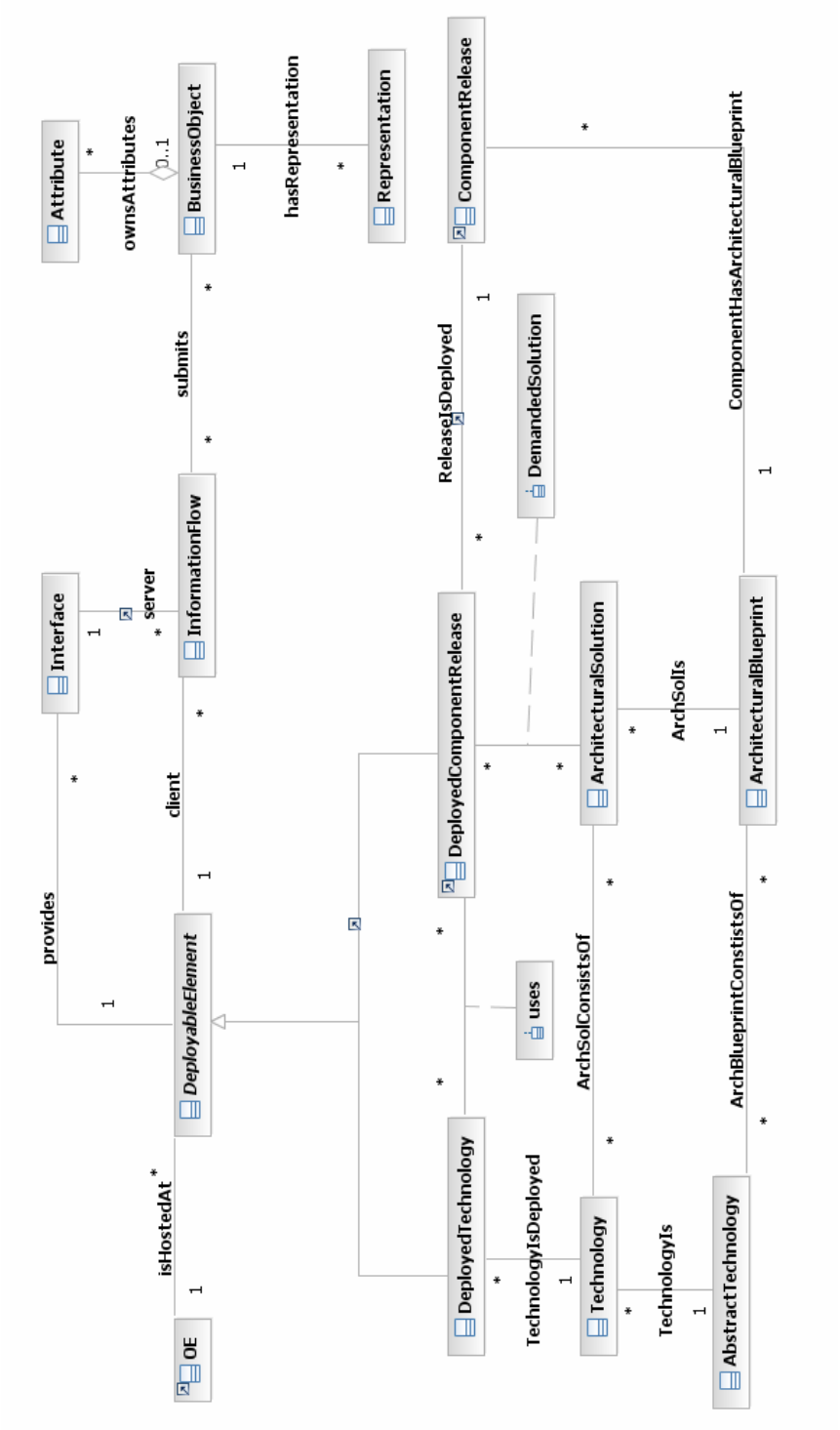


Figure 4.4.: Final state of the information model

4.2.1. Business Object Representation

Overview section:

Name:	Business Object Representation
Id:	I–84
Alias:	
Summary:	The I-Pattern provides a structure for organizing information about different representations, e. g. as a POJO, of a business object.

Solution section

One of the most valuable asset of todays company is information. Companies have to deal with a huge amount of information and process it throughout various workflow steps. By doing so it is potentially necessary to persist the information or transmit it over several devices. The goal of this pattern is to ensure that on every device in this step the business object is consistent, by defining its possible representations. For example the `BusinessObject` customer can be represented as POJO, which is referenced with the attribute `location` or an XML-stream, which is described by a XSD-file again referenced with the attribute `location`. Such a reference could be for example a URI. The consistency between the different representations can be assured either by the architect or by an automation process that compares the data in the information model (see I–Pattern I-46 from [Bu08b]) with the one in the different representations.

The benefit of the utilization of this pattern is the reduction of inconsistencies and a possible reuse option for implemented source code. This reuse goal can be reached when developers look up the already existing representations of the business objects they want to use in their applications.

The concepts of figure 4.5 are defined as follows:

BusinessObject: An BusinessObject represents a business entity (e. g. an invoice) that is used during the execution of a business process, which performs operations (CRUD) on the BusinessObject. [Bu08b] The only attribute of this class is the name of the business object.

Representation: A Representation can be any kind of storage possibility for the `BusinessObject`. This can be for example a XML-File, a POJO or a binary storage format. The attribute type defines the type of the representation, e. g. CSV-File or POJO and the attribute location allows to refer to a specific definition as for example an XML Schema Definition. The name attribute helps to keep different definitions apart.

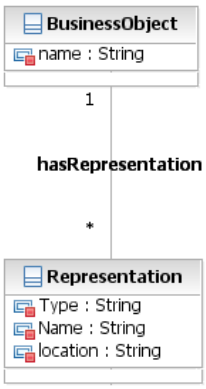


Figure 4.5.: Information model fragment for I–Pattern business object representation

The associations of the information model fragment serve the following purposes:

hasRepresentation: A `BusinessObject` can have several `Representations` for storage or transmission.

Consequence section:
This pattern builds an extension of pattern I–46 and needs at least the concept `BusinessObject` from that pattern. This pattern allows to identify what implementations and representations of a `BusinessObject` are already existing. The advantage of this is, that application developers avoid the development of the same storage possibility twice or ensure the use of the same schema for storage, as a consequence the redundant development of `Representations` for one `BusinessObject` is avoided and costs are cut. This leads to a minimized effort when using already defined `BusinessObjects`. A disadvantage of this pattern is the needed process to ensure that developers use the already existing `Representations` and do not create their own ones.

4.2.2. Virtualization

Overview section:

Name:	Operating System Virtualization
Id:	I–85
Alias:	
Summary:	The I–Pattern provides a structure for organizing information about virtual machines and the deployment of component releases on different physical and virtual machines.

Solution section
One of the most recent and radical changes in the software hosting business is an increased usage of virtualization. This leads to a reduction of physical machines, which demands a more efficient usage of resources. Figure 4.6 illustrates the information model fragment, which addresses problems of virtualization. This fragment can be found in the second draft of the information model, because of the replacement with a CMDB it was removed in the final model.

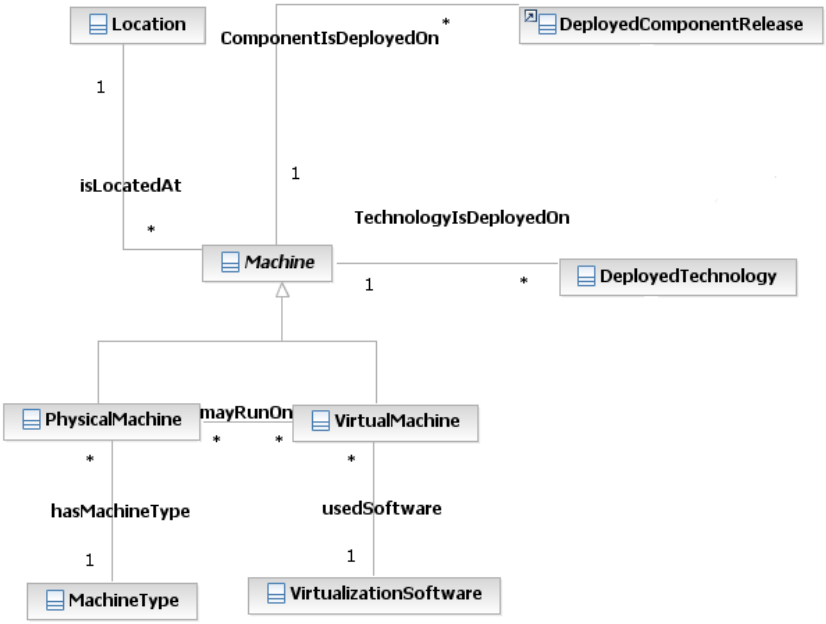


Figure 4.6.: Information Model for Virtualization

The concepts of Figure 4.6 are defined as follows:

Location: Describes a physical (e. g. Munich, London, New York) or virtual (e. g. Internet, Intranet) location. [Bu08b]

DeployedComponentRelease: This is a specific installation of a concrete release of a component.

DeployedTechnology: A `DeployedTechnology` is the installation of a technology on a certain instance of an operating system.

Machine: In this case `Machine` is an abstract class that can either be a physical machine or a virtual machine.

PhysicalMachine: This class represents a physical instance of a machine which may either directly host a deployed component release or a deployed technology or virtual machines can be running on it.

VirtualMachine: A virtual machine is a virtual instance of a machine. It can run on one or more physical machines and needs a certain virtualization software, which works as host application.

MachineType: The machine type indicates a certain configuration of a machine this can either be specific hardware components and/or software components.

VirtualizationSoftware: The virtualization software indicates a specific software for virtualization.

The associations of the information model fragment serve the following purposes:

ComponentisDeployedOn: This relationship indicates on which Machine (either physical machine or virtual machine a deployed component release is deployed. A deployed component release can only be deployed to a single machine.

isLocatedAt: Any machine is located at a certain location, e. g. a computing center.

TechnologyIsDeployedOn: Refer to `ComponentisDeployedOn`.

mayRunOn: This relationship indicates that a certain virtual machine can run on any of the related physical machines. This is a n–to–m relationship because a virtual machine can be moved between different physical machines.

hasMachineType: This relationship indicates that a physical machine is a certain type of a physical machine.

usedSoftware: This relationship defines that a specific software is used for virtualization.

Consequence section:
When using this I-Pattern the amount of collected data is quite large, because all physical machines, all virtual machines and their relationships to deployed component releases have to be stored. Besides the amount of information another challenge is the ongoing maintenance of the data caused by the high flexibility of the relationship between physical and virtual machines the information about the deployment becomes obsolete very fast.

This pattern allows the mapping of operating system virtualization to the information model, which is widely used in companies now. Therefore, it is necessary to integrate it in the application landscape management process in order to support planning stages. Furthermore, the planning of virtualization is supported by this pattern.

A possible further development of this pattern could be the extension to model information about application clusters and failover solutions. This pattern can also be easily integrated with Pattern I–66, because the technology on which a `DeployedTechnology` is based on is equal with the the `Technology` of Pattern I–66.

4.3. Mapping to ARIS IT Architect

The next important step was to introduce the developed model in the German insurance company. Prior to implementing the developed information model in the German insurance company a customization of the EAM tool established at the German insurance company need to be conducted. The mapping from the UML model to ARIS IT Architect showed that the information model used by ARIS IT Architect was not capable of supporting all concepts used. Although not scope of the compliance management the concept of license management is an example for this problem.

The validation of the final information model was performed in two steps. The first step was to map it to ARIS IT Architect to check if all used concepts are available. The second step and final approval takes place when it is used operatively by the various users and stakeholders to validate the completeness of the information model.

In summary the output of the design phase was a information model, which is mapped to the ARIS IT Architect, which is used by all users in the German insurance company.

CHAPTER 5

Implementation – V and M Patterns

Complementing the three-fold EAM pattern approach introduced in Chapter 2, this chapter details on the V– and M–Pattern, which use the information stored according to the I–Patterns presented in Chapter 4. The M–Patterns provide guidelines how to address the concerns identified using V–Patterns, which define the presentation of the data/information according to the needs of the involved stakeholders. Thereby, the development of the V– and M–Patterns was performed together in one step because of the tight integration between them both. The used proceeding is detailed in Figure 5.1.

The first step after the definition of the information model, was to map it to the information model of the ARIS IT Architect. Caused by limited customization possibilities of ARIS IT Architect, the mapping could not be performed without misusing several existing concepts. No concept for information flows for example does exist in ARIS IT Architect which lead to the misuse of the concept *information carrier*. The possible mapping alternatives have a big impact on the proceeding in ARIS IT Architect (see M–Pattern explained in Section 5.2). Therefore, the description of these tasks and the mapping was done at the same time in an iterative approach. The used approach led to a propagation problem, as every change in the mapping required a new evaluation of the modeling task. The same happened vice versa. In addition to the M–Pattern also V–Patterns have been defined at the same time, these V–Patterns are detailed in Section 5.1.

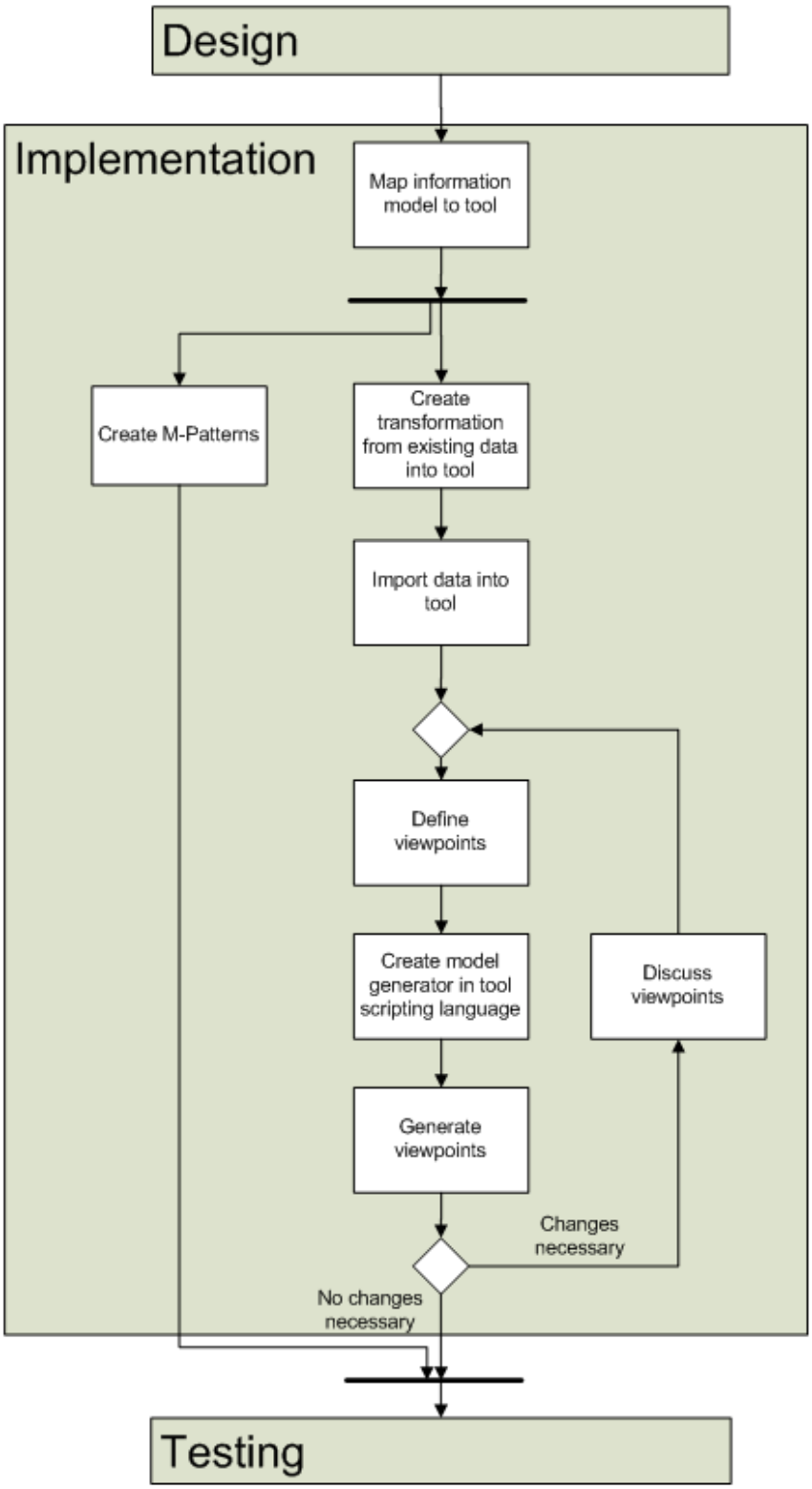


Figure 5.1.: Detailed proceeding of the implementation phase

5.1. Viewpoints

Even if a company stores all its gathered information about its application landscape in a huge database it is nearly useless, if it is not visualized or processed in a way that the stakeholders who need to work with the information can handle them without enormous effort. Therefore, various V–Patterns are needed to support the different steps of EA management.

The EAMPC defines a V–Pattern as follows:

> A Viewpoint Pattern (V–Pattern) provides languages used by M–Patterns. A V–Pattern proposes a way to present data stored according to one or more I–Patterns. [...] In order to ensure the understandability of a view, we regard a legend to be mandatory. [Bu08b]

According to the definition provided above all visualizations and reports, which present the stored data can be seen as Viewpoints. Because of the amount of data, that has to be visualized, most of the visualizations used in this work are defined as reports.

In order to fulfill of the requirements of the German insurance company it was necessary to extend and redefine the models of ARIS IT Architect. Thereby, the used models can be divided into three different types:

Navigation and structuring Models are used for navigation or structuring purposes, these models show the structure of the systems or groupings of systems and allow navigation to more detailed models of the selected part.

Application models illustrate detailed information about systems as e. g. application interfaces, user groups, or used technologies.

Reports are models, which are used for reporting, these models allow to transmit information about certain circumstances in an efficient way.

As ARIS IT Architect defines a huge set of different model types and supports the user in customization by defining a filter, which masks specific elements of the model, it is very easy to hide certain elements. Nevertheless, the extension of certain models regarding further object types is not allowed. Whereas, the customization using the filter functionality can be performed quite easily, a drawback can be found, as the built-in filter represents the only customization possibility provided by ARIS IT Architect for models, it was used to perform the customization.

The layouters of ARIS IT Architect does not work appropriately, when filtering is utilized. This leads to models were certain elements are displayed above each other or lines are not drawn as direct connections between the associated elements. This problem could only be solved by implementing appropriate layouting scripts in the scripting language of the tool.

In addition to models have been used as V–Patterns even more important where the reports that could be created on basis of the information model. Some of the V–Patterns seemed to be usable for any kind of organization, therefore two of them are detailed in the next paragraphs.

5.1.1. Kiviat diagram showing conformance levels

V–Pattern Overview	
Id	V-83
Name	Kiviat diagram showing conformance levels
Alias	
Summary	Kiviat diagram giving fast overview about the conformance levels of e. g. regions, OEs, or standard types.
Version	1.0

5.1.1.1. Solution Section

Figure 5.2.: Example of a kiviat diagram showing the conformance level grouped by regions

This type of diagram shows the conformance levels of logical units, e. g. regions, OEs, or standard types. Further grouping options are possible but not used in the view presented. An example of the viewpoint can be found in Figure 5.2.

The conformance levels values are calculated as aggregation of application. This could be, for example, *count of applications, which conform to at least one standard* in relation to *count of all applications of the region*. Of course it is possible to define other calculation options such as *count of applications that do not conform to any standard* in relation to *count of all applications of the region*. Thereby, the calculation formula of the report can be adapted to the grouping option and/or the available information.

5.1.2. Report giving an overview about the conformance levels for a specific standard

---

V–Pattern Overview

Id	V-84
Name	Report giving an overview about the conformance levels for a specific standard
Alias	
Summary	Detailed report about conformance levels for specific standards according to different groupings. Furthermore the different values are highlighted according to their value.
Version	1.0

5.1.2.1. Solution Section

	Region A	Region B	Region C	Region D	Region E
Standard 1	3,98	2,54	1,12	2,46	3,43
Standard 2	2,87	3,47	2,34	2,53	3,01

1,00-1,99	x,xx
2,00-2,99	x,xx
3,00-4,00	x,xx

Figure 5.3.: Example report showing the conformance levels of organizational entities for specific standards

The report shown in Figure 5.3 shows on the x-axis either the different OEs or different regions and on the y-axis it lists all standards. In the cells the conformance level is shown and the cell's background is colored according to the conformance level to give a better

overview. The report allows the usage of different calculation formulas. As stated above the calculation formula for the conformance levels can be adapted to the grouping option and/or other available information.

In addition to the two introduced V–Patterns several other reports and visualizations have been considered to be used in the context of compliance management during the conduction of this work. These V–Patterns can be found in Section A.

5.2. Methodology patterns

In addition to the I– and V–Patterns the steps needed to achieve certain goals have to be defined. In the EAMPC these steps are contained in the M–Patterns, which the EAMPC defines as follow:

> A Methodology Pattern (M-Pattern) defines steps to be taken in order to address given concerns. Furthermore, as a guidance for applying the method, statements about the intended usage context are provided, which include the concerns to which the M-Pattern can be applied. These concerns are addressed by procedures defined by the M-Pattern, which can be very different, ranging from e. g. visualizations and group discussions to more formal techniques as e. g. metrics calculations. Missing methodologies constitute a common issue in EA management information models.

Due to the fact that such M–Patterns are often very company-specific, it is not easy to develop M–Patterns that are best practice. Therefore, the result of this work is only one M–Pattern that seemed to be interesting for other companies and, which is detailed in the following subsection.

5.2.1. Maintenance of architectural solutions (M-35)

M–Pattern Overview

Id	M-35
Name	Definition of architectural solutions
Alias	
Summary	This M–Pattern describes how new and existing architectural solutions should be introduced and maintained.
Version	1.0

5.2.1.1. Problem Section

The methodology addresses the following concerns:

- C–2: Which technologies, e.g. programming languages, middleware, operating systems, database management systems, used in the application landscape should be replaced, which ones should be kept? [Bu08b]
- C–9: Possibilities to reorganize the application landscape in respect to the used technologies should be outlined. Thereby, possible goals are: Reducing licensing costs, reducing maintenance costs, taking into account the support periods of the technology products, etc.[Bu08b]
- C–46: Which knowledge about specific subjects, e.g. technologies, or programming languages, is currently available in the organization?[Bu08b]
- C–50: How is an architectural blueprint / architectural solution made up?[Bu08b]

The definition of new architectural solutions is one of the key factors for the future planning of an EA landscape and subsequently this is one of key the success factors of the future development.

5.2.1.2. Solution Section

The definition of new architectural solutions and the update of existing architectural solutions is a complex task, because architectural solutions have to consider not only architectural blueprints and the existing landscape but also possible future developments.

The methodology uses the following viewpoints:

- V–26: Time Interval Map visualizing Lifecycles of Applications
- V–83: Kiviat diagram showing conformance levels
- V–84: Report giving an overview about the conformance levels for a specific standard
- V–85: Report on all available Architectural Solutions

This methodology describes several ways to find possible new architectural solutions or update of architectural solutions. Furthermore, the implications of planned changes should be estimated before they are implemented. The different options and triggers for changes are detailed:

- Technologies that are widely in use may constitute possible candidates, this can be done for example with a report, similar to the one shown in Subsection A.1.
- Identification of architectural violations that occur on a regular basis may point on possible needs for action.

- Fade out, replacement, or introduction of a certain technology may lead to the need for updating architectural solutions.
- New architectural solutions can be chosen on basis of the knowledge of skilled designers.

The triggers for changes in the architectural solutions can be diverse. Changes in available technologies, for example can lead to changes in architectural solutions. The fade out of a technology may lead to a replacement of this technology for all its occurrences in architectural solutions. Another trigger may be impact from procurement, e. g. changes in the available technologies caused by expired contracts. Finally, architectural solutions have to be regularly updated because of new conceptions made for the architectural future of the application landscape.

The actions such a trigger can initiate are either to change an existing architectural solution, to introduce a new one, or to fade out an existing one. Most of the time an existing architectural solution is faded out, it is necessary to recommend an appropriate replacement. If this is not done, the amount of architectural violations could be enormous and the application developers will have to change the application to be in compliance with an architectural solution.

5.2.1.3. Consequence Section

An appropriate management of architectural solutions is very important for the future development of an application landscape. Nevertheless, not only the future development is affected by architectural solutions, an increasing homogeneity of the application landscape may also be a consequence of adequate architectural solutions. This leads to decreasing costs and a more efficient development of new applications. Last but not least the management of architectural solutions tries to avoid the uncontrolled growth of used technologies. Adequate administration of architectural solutions is the basis for an easy and usable evaluation of the standard conformity of an application landscape as explained in M–Pattern M–2 (Analysis of Standard Conformity of the Application Landscape) of the EAMPC.

The above described pattern was evaluated in the context of this work at a German insurance company. Based on an analysis of the as-is situation via the utilization of M–2 using V–83 and V–84, M–35 was successfully used and evaluated as detailed in the next Section.

5.3. Testing

The evaluation of the patterns introduced above was done by implementing them in ARIS IT Architect. Furthermore, a document, which was distributed among the participating OEs was used to support the stakeholder in the adaption phase. In addition access to the tool was granted to these OEs to act as test users.

At the beginning of the evaluation was the review of the document and the existing implementation by these OEs. Thereby, one OE submitted comments, which have been discussed and for the most part implemented in the second version of the document and models of the tool. Basically the received feedback concerning the used concepts was very positive and no major changes have been raised. Most of the changes concerned used models and the fact, that not every symbol was used uniquely. This remark was addressed by using different colored symbols for similar concepts.

Caused by the sole iteration of feedback as further iterations have not been in the scope of this work, no final approval of the usage was done until today. Maybe the fact that several other OEs started to use the output of this work, is an evidence that the defined I–, V–, and M–Patterns are practically relevant. Nevertheless, besides the different patterns developed, there are also some lessons learned, which can be derived from this work. These lessons learned are detailed in the last section.

CHAPTER 6

Findings

The goal of this work was to introduce a compliance management approach at a German insurance company by using the EAMPC. Thereby, special attention was given to the usability of the EAMPC, and possible extension of the EAMPC with further useful patterns for compliance management. Special attention *was thereby paid* on the used approach, because it is different to the one proposed in the EAMPC.

The EAMPC allowed a very fast orientation in the field of EA management and provided a strong basis on which it was possible to build on. Nevertheless, caused by the specific requirements of the German insurance company an approach different to the one described in the EAMPC was used. The reasons for this were that the German insurance company wanted to have a quick insight into the compliance management implementation, and besides this, a partial information model already existed in the company. The used approach had the advantage of a very early first look into the possible usage of the models. Furthermore, it allowed a quick first implementation in the used tool. Thereby, simplifying the considerations of the constraints of the tool in an early stage of development. In addition, it enabled the user to evaluate the later usage scenarios already during development. However, there are also some disadvantages, as for example the possible need for changes to the information model in later stages, when new requirements arise while developing models or visualizations. A change to the information model, which builds the foundation leads to changes in all following steps. Another disadvantage is, that all involved persons have to be familiar with data modeling, otherwise the persons are not able to communicate with each other and understand the implication of a certain model.

One major outcome of this work is the experience gathered which can be used for other companies and was therefore, converted into patterns for the EAMPC. Among these patterns is an the information pattern *Virtualization*(see Section 4.2.2) for representing virtual machines in the infrastructure of a company. The *Virtualization* pattern is very useful for companies that use virtualization and want to get an overview about the deployment of applications. This pattern was developed out of the requirements of a stakeholder of the German insurance company. Nevertheless, in a later stage of this work the German insurance company decided to pass this function on to a CMDB and therefore this pattern was not included in the final model. Another I–Pattern introduced is *Business Object Representation*(see Section 4.2.1), this pattern shows how information about different representations of the same business object can be stored.

Furthermore, two V–Patterns have been defined in the context of this work which give a quick overview about the compliance of different OEs of a company. The first one, called *Kiviat diagram showing conformance levels*(see Section 5.1.1), is a kiviat diagram showing conformance levels of different units, as e. g. OEs, regions, or standard types. This management report allows easy comparisons between the axes and helps to identify possibilities for improvement. The second V–Pattern *Report giving an overview about the conformance levels for a specific standard*(see Section 5.1.2) contains a report, which defines an overview about the conformance levels for a specific standard. Thus, it is a more in detail view on specific standards and their conformance in different regions or OEs. This report helps to identify the usage of certain standards and allows subsequently the identification of options for action. While defining the different visualizations it turned out to be complicated to visualize the huge amount of data available. Therefore, graphical visualizations turned out to be not the appropriate way to communicate. Tabular reports arose as an alternative able to display a lot more data in an arrangeable format and therefore, are widely in use.

One task, which arises regularly in the context of compliance management in general and compliance management in special is the introduction and maintenance of architectural solutions. The definition of architectural solutions has a great influence on the future evolution of the application landscape. Therefore, this work developed an M–Pattern called *Maintenance of architectural solutions*(see Section 5.2.1), which addresses the challenge of architectural solution maintenance. The pattern names possible triggers for a change of existing or new architectural solutions. This pattern was introduced in the German insurance company and lead to a standardized process for updating and introducing architectural solutions.

The patterns developed in this work will be introduced in the next revision of the EAMPC. Furthermore, this work builds a foundation for several further topics that deserve a closer look. An interesting topic for example are the impacts of the introduced automatic compliance reports on the behavior of the stakeholders who are responsible for it. The automatic and subsequently quick creation of such reports makes the control of the compliance cheap and easy. A further interesting empiric survey would be the analysis of the impact of au-

tomatic reporting on data quality and in series the further implications of data quality on the control of the compliance management. This could be the key point of further research. Every company needs to evaluate its projects, and therefore, needs to determine the benefits of e. g. the introduction of compliance management. The question arising here is *How to calculate the benefit of compliance management?*, which is an interesting question for further study. In addition, the developed EAM patterns have only been evaluated in the company where they have been developed at. The usage in further companies will show further experiences and possible enhancements.

APPENDIX A

Full listing of V–Patterns

Listing of considered V–Patterns, also the ones that are not explained in 5.

A.1. Viewpoint V-26

V–Pattern Overview	
Id	V-26
Name	Time Interval Map visualizing Lifecycles of Applications
Alias	
Summary	This V–Pattern visualizes the lifecycles of business applications, including the respective versions.
Version	1.0

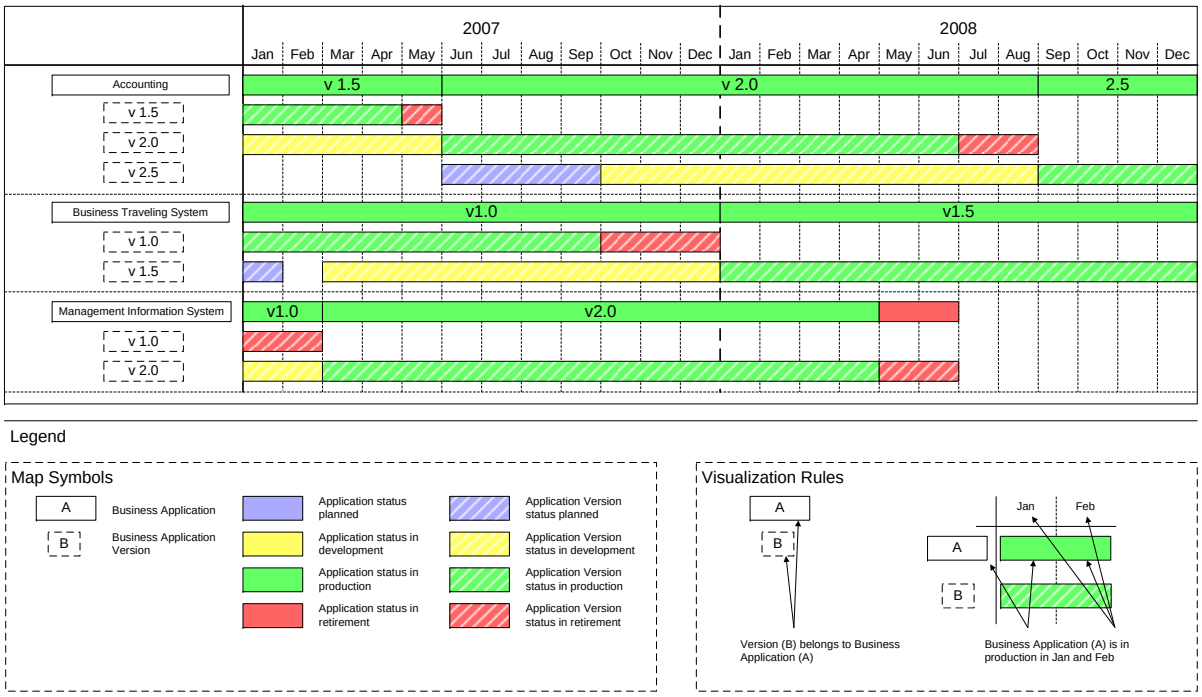


Figure A.1.: Viewpoint V-26

A.1.1. Solution Section

This V–Pattern is an interval map showing the lifecycles of business applications in an aggregating view, which is additionally detailed by the respective versions. The V–Pattern depends on I–Pattern I-26.

A.1.2. Consequence Section

The status of the business application in this V–Pattern has to be derived by the status of the corresponding business application versions. Thereby, an ordering of the business application version status can be used for the deduction, e.g. if there is a business application version, which is in state *in production*, this state overrules the other status and the

Business Application is assigned the status *in production*. The needed information can be gathered form the I–Pattern I-26.

A.1.3. Alternatives

An alternative visualization could possible be Figure A.2

Technology	Lifecycle	Number of uses
Oracle 9i	Introduction	15
Apache 2.2.8	Phase Out	8

Figure A.2.: Example report on Technologies with their lifecycle information as table

A.2. Viewpoint V-49

V–Pattern Overview

Id	V-49
Name	Communication Table
Alias	
Summary	This V–Pattern visualizes the data flow of business objects.
Version	1.0

A.2.1. Solution Section

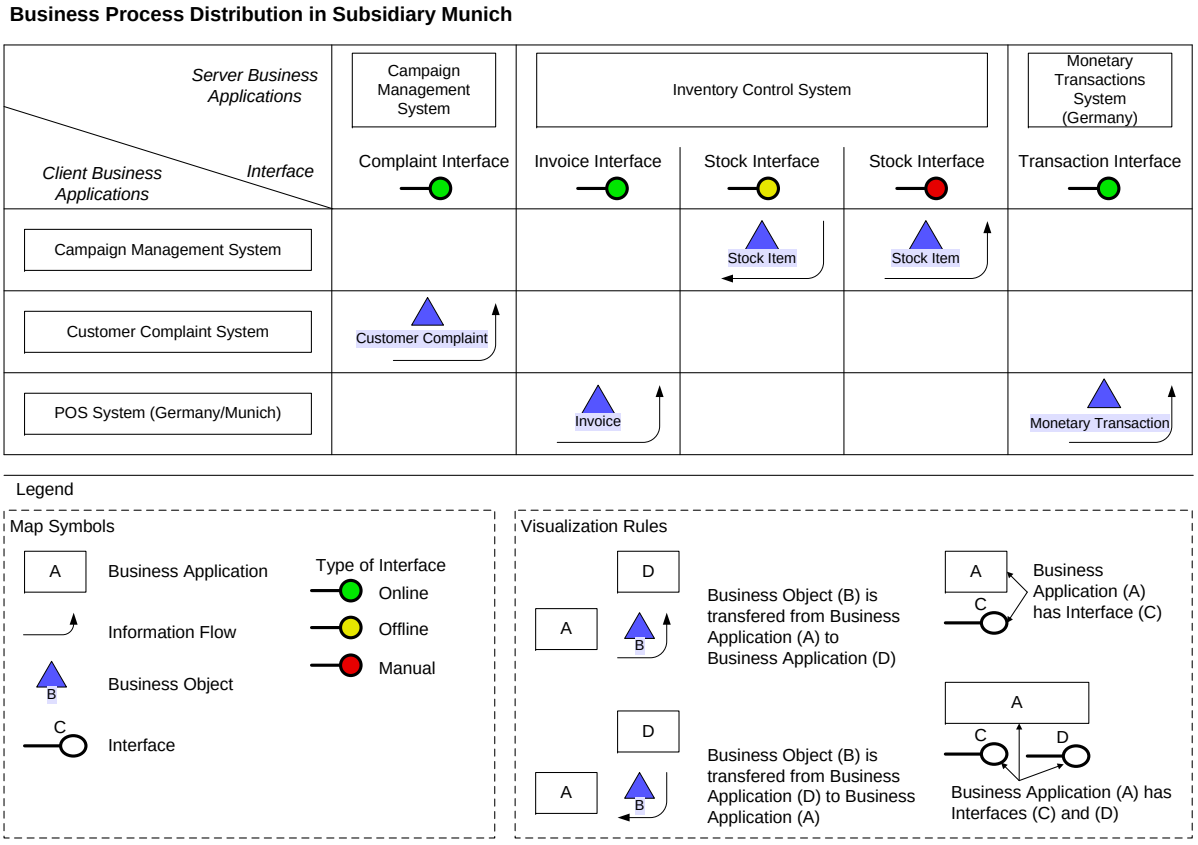
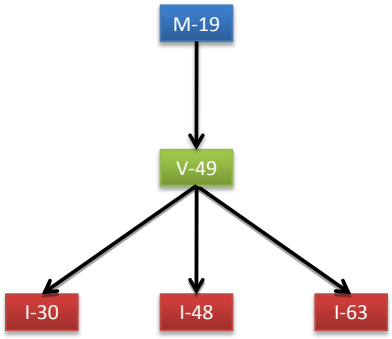


Figure A.3.: Viewpoint V-49

This V–Pattern shows in a table, how business objects are exchanged between business applications, thereby considering the interfaces offered by the business applications. This V–Pattern is based on I–Patterns I-30, I-48, and I-63.



A.2.2. Consequence Section

The I–Pattern s I-30 and I-63 can easily be integrated by the concept ”Business Application”, which can be found in both I–Patterns.

The resulting information model from the integration of I-30 and I-63 can be integrated with I-48 by the concept ”SupportRelationship”.

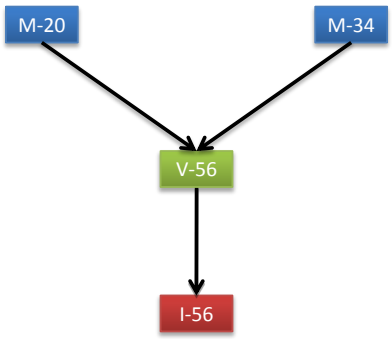
The information required for this V–Pattern is quite similar but more extensive to the information required for V–Pattern V-82. Therefore, V-82 may also be considered when modeling business objects that are transfered over interfaces.

A.3. Viewpoint V-56

V–Pattern Overview	
Id	V-56
Name	Infrastructure Usage
Alias	
Summary	This V–Pattern shows the infrastructure services offered by infrastructure software, which are used by business applications.
Version	1.0

A.3.1. Solution Section

This V–Pattern shows the infrastructure services offered by infrastructure software and used by business applications. This V–Pattern is based on I–Pattern I-56.



A.3.2. Alternatives

An alternative visualization could possible be Figure A.5

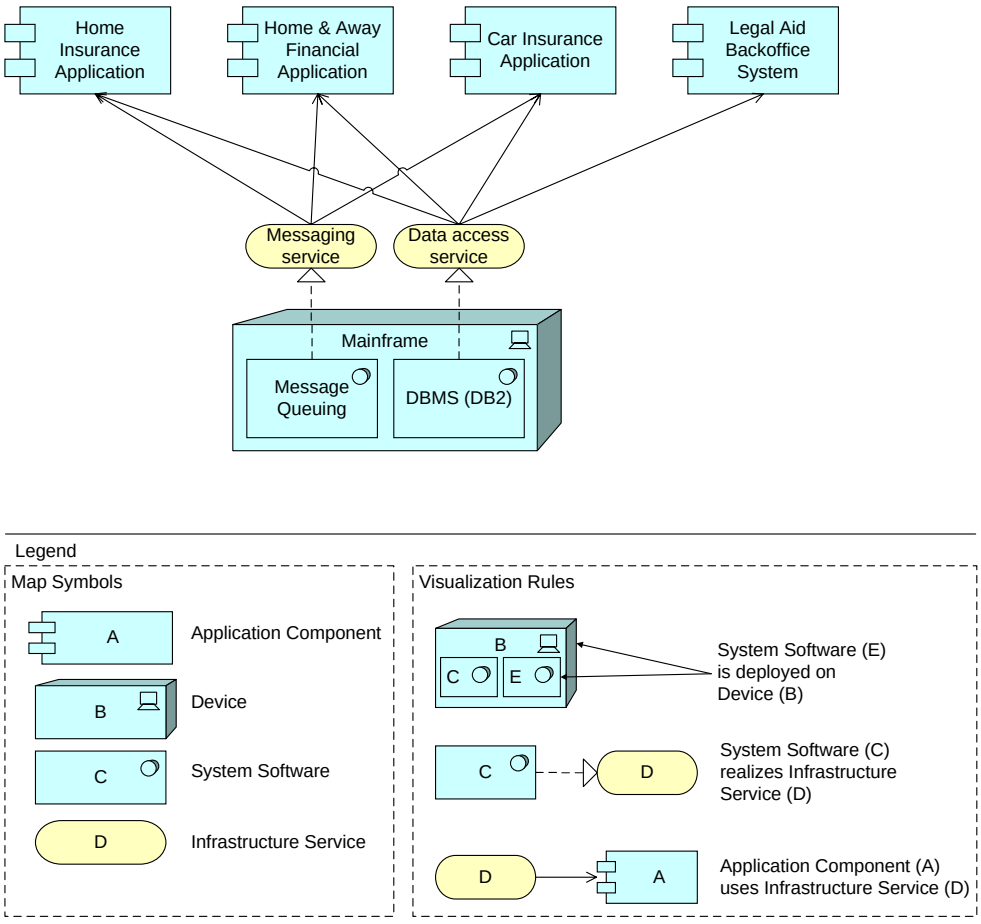


Figure A.4.: Viewpoint V-56

Hardware	Apps
Server213	Oracle 10g DB 2 Message Queuing
Server 345	TomCat 5

Figure A.5.: Example report listing hardware and the application running on it

A.4. Viewpoint V-75

V–Pattern Overview	
Id	V-75
Name	Business Application Deployments
Alias	
Summary	This V–Pattern uses the notation of an UML 2.0 deployment diagram and shows, how infrastructure components are used by business applications.
Version	1.0

A.4.1. Solution Section

This V–Pattern uses the notation of an UML 2.0 deployment diagram and shows, how infrastructure components are used by business applications. This V–Pattern is based on I–Pattern I-75.

A.4.2. Alternatives

An alternative visualization could possible be Figure A.7

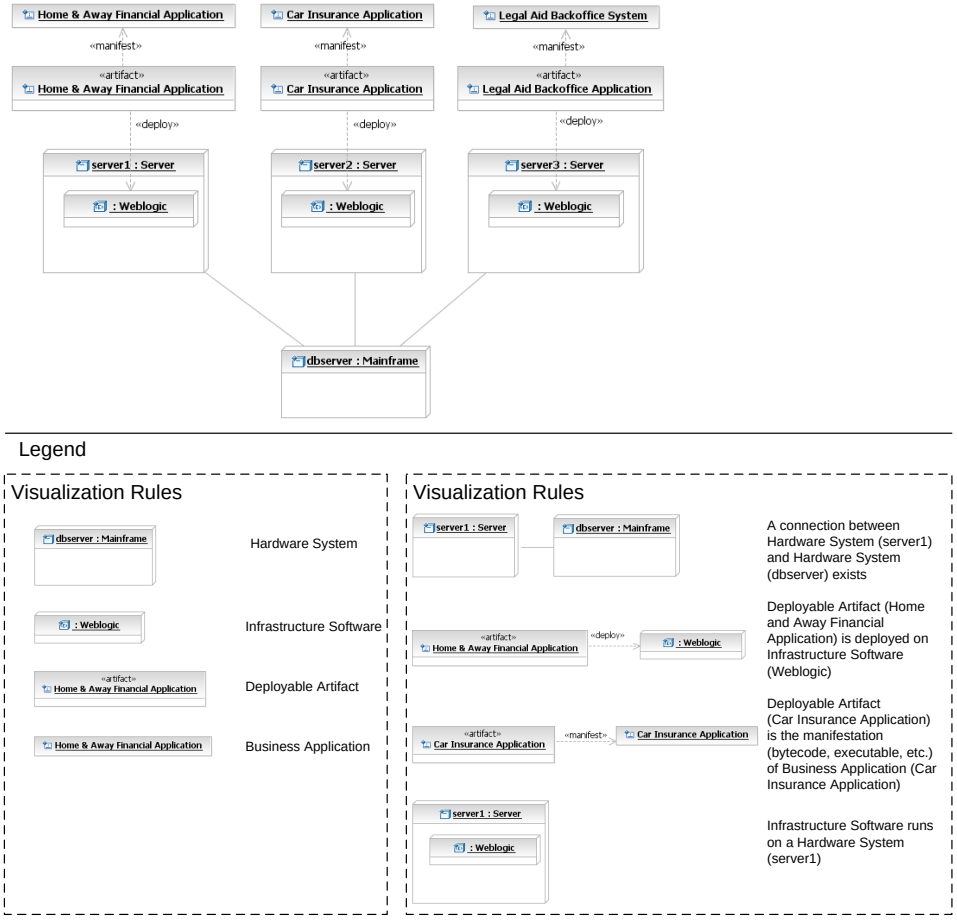


Figure A.6.: Viewpoint V-75

Hardware	Apps
Server213	Oracle 10g DB 2 Message Queuing
Server 345	TomCat 5

Figure A.7.: Example report listing hardware and the application running on it

A.5. Viewpoint V-85

V–Pattern Overview

Id	V-85
Name	Report on all available Architectural Solutions
Alias	
Summary	This report lists all currently available Architectural Solutions.
Version	1.0

A.5.0.1. Solution Section

Architectural Solution	Number of uses	Number of violations
OracleTomcat	15	2
MySQLJBoss	5	3

Figure A.8.: Example report on available Architectural Solutions as table

This Report can also be visualized as bar chart.

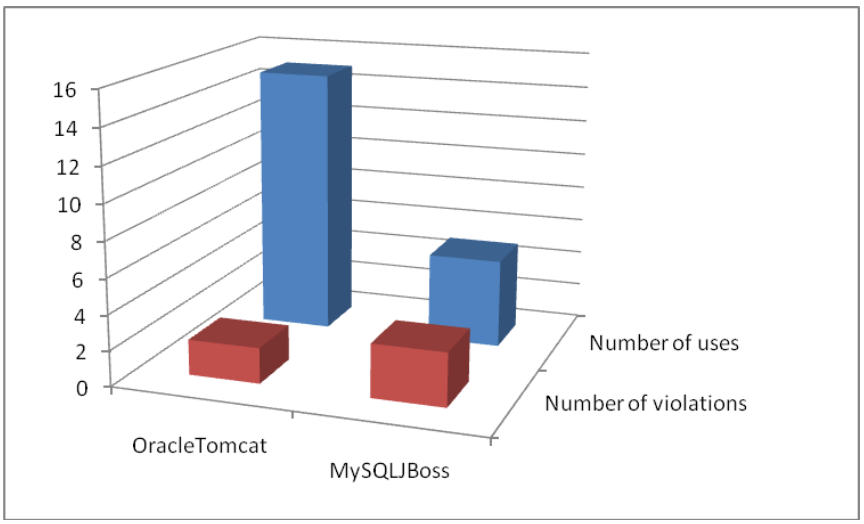


Figure A.9.: Example report on available Architectural Solutions as bar chart

This report lists all currently available Architectural Solutions, maybe with several attributes as e.g. Number of current uses, Number of violations or complete description of the Architectural Solution.

A.6. Viewpoint V-86

V–Pattern Overview

Id	V-86
Name	Report on Deployed Component Releases
Alias	
Summary	Report listing all Deployed Component Releases filtering by specifying criteria.
Version	1.0

Deployed Component Release	Technology	Support End Date
CRM System	Oracle 8g	10.5.2010
ARIS IT Architect 7.0.2	Oracle 9i	31.3.2013

Figure A.10.: Example report on Deployed Component Releases as table

Report listing all Deployed Component Releases filtering by specifying criteria as e.g. "All Deployed Component Releases using a Technology having a Support End Date within the next <x> months"; "All Deployed Component Releases based on a Component Release having a Support End Date within the next <x> months"; "All Deployed Component Releases using a specific Architectural Solution"

A.7. Viewpoint V-87

V–Pattern Overview

Id	V-87
Name	Report with count for each standard
Alias	
Summary	This report helps to identify the usage of certain standards.
Version	1.0

A.7.0.2. Solution Section

Standard	Count
Java	134
Cobol	14

Figure A.11.: Example report listing all Deployed Component Releases that are not associated with any standard

This report should list all available Standards and how often they have been used. This report should help to identify which Standards should be pushed and which ones should be faded out. The easiest way to visualize this information is to create a table.

A.8. Viewpoint V-88

V–Pattern Overview

Id	V-88
Name	Usage of certain Business Object
Alias	
Summary	This report shows all usages of a given Business Object.
Version	1.0

A.8.0.3. Solution Section

Business Object	Deployable Element client	Deployable Element server	Connection Type
Customer	CRM System	OS/2	read

Figure A.12.: Example report showing how certain Business Objects are used

This report shows all usages of a given Business Object. This report can be used to see whether a Business Object is still in use or can be deleted without questions. This report should be visualized as a tabular report as shown in Figure A.12.

APPENDIX B

Full listing of M–Patterns

M–Patterns of EAMPC [Bu08b] referenced in the work.

B.1. Analysis of Standard Conformity of the Application Landscape (M-2)

M–Pattern Overview	
Id	M-2
Name	Analysis of Standard Conformity of the Application Landscape
Alias	Analysis of Architectural Standards
Summary	This M–Pattern gives an overview, which business applications conform to architectural standards
Version	1.0

B.1.1. Problem Section

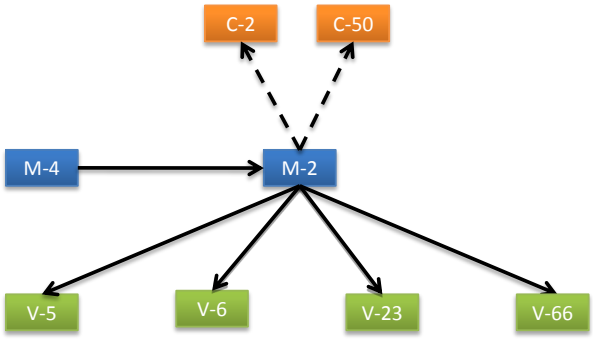
The methodology addresses the following concerns:

- **C-2:** Where are architectural blueprints or architectural standards used, and are there areas where those standards are breached?
- **C-50:** How is an architectural blueprint / architectural solution made up?

One of the fundamental problems to be addressed in this context is the growing complexity of the application landscape induced by the uncontrolled increase in used technologies, architectures, platforms, etc. Controlling this growth is thereby supposed to increase efficiency in IT operation and development, e.g. due the possibility to focus necessary knowledge on selected technologies, platforms, etc.

B.1.2. Solution Section

Architectural solutions and architectural blueprints consider homogeneity not only on the level of a specific kind of technology e.g. programming languages or middleware, but include architectural solutions and consider technologies at the level of standardized technology *bundles*.



The methodology uses the following viewpoints:

- **V-5:** Standard Conformity Layer for C-2
- **V-6:** Clustering by Standard, for C-2
- **V-23:** Technologies by Architectural Standard specifically for C-50
- **V-66:** Architectural Solution in detail (UML ; This viewpoint also addresses C-50. While it has not been selected via the online questionnaire, we reference this viewpoint nevertheless, as it illustrates the basic concepts behind architectural solutions and architectural blueprints as a way to create architectural standards.

This methodology describes basic steps for creating an overview of which (deployed) business application uses which architectural solution, and gives hints for analyzing this overview.

- For collecting information, it has to be noted that the employees operating a (deployed) business application might not always be totally aware of the respective application's architecture. Thus, the respective developers might have to be included into the data collection process. Of course, up-to-date blueprint and solution definitions are a prerequisite for this task (see methodology M-4). Additionally, an understanding about the blueprints should exist with the developers. This could possible require a more detailed overview than provided by V-23.
- The collected information should be verified. Also here different possibilities apply, ranging from automated plausibility checks to manual reviews, which could be tied into visualization creation. If necessary, missing or possibly erroneous information has to be delivered in addition or corrected.
- Creation of the visualizations (e.g. according to V-5 and V-6)

A V-23 - diagram can provide first background information about the existing architectural blueprints and solutions. It can give a first overview of the technologies included in a standard. This allows a first stage of the analysis: The set of standards might be to small (too restrictive) or too big (too permissive).

In analyzing diagrams according to viewpoints V-5 and V-6, the focus is likely to be on the business applications not conforming to the respective architectural standard. On the one hand, such business applications might be looked at specifically, considering e.g.:

- Does it require not to conform with the standard?
- How much are costs thus induced? Who bears these costs?
- Has the wrong standard been prescribed for the application?

On the other hand, analysis can also focus on the totality of the non-conforming business applications, e.g. looking at:

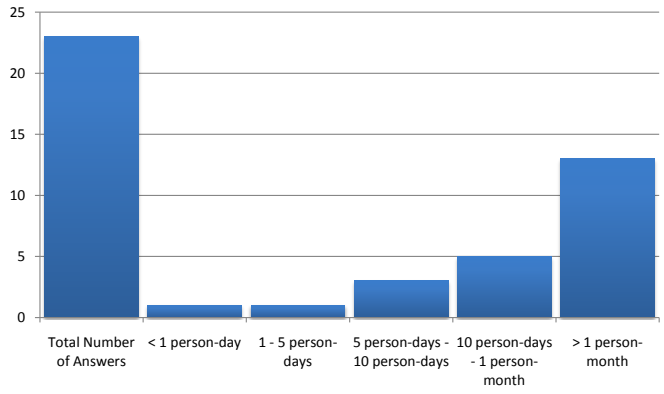
- What do they have in common?
- Are the standards inadequate for important parts of the application landscape?
- Are there organizational units for which there are no means of enforcing the standards?

Especially a V-6 - diagram might be helpful in getting an impression of the importance of the different architectural solutions. A standard only existing to serve a small proportion of the business applications might need a special justification.

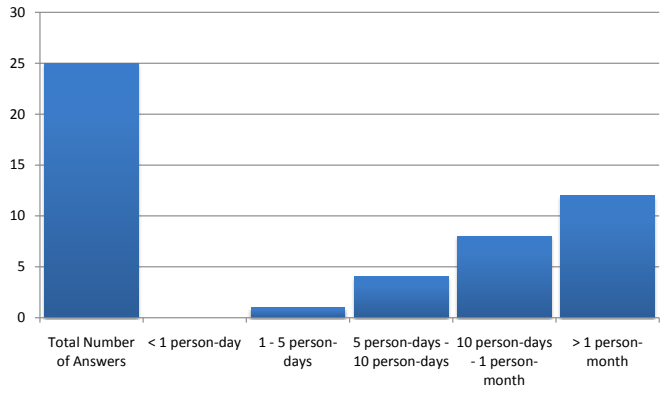
B.1.3. Consequence Section

The architectural blueprints and solutions need to live as *boundary objects*[1] in the communities of the software architects and the enterprise architects. Only if these two communities interpret the blueprints and solutions in a coherent way, they can be supposed to be an instrument towards a less uncontrolled growth of the technology zoo behind an application landscape.

The data collection effort per year for information about *business applications, architectural solutions, and assignment of architectural solutions to business applications* has been stated by practitioners using such methodologies as:



The data collection effort per year for information about *the structure of architectural solutions and its components or technologies* has been stated by practitioners using such methodologies as:



[1] A boundary object is an object which allows members of different communities to build a shared understanding in respect to certain things. Boundary objects are interpreted differently by the different communities, and realizing and discussing these differences can lead to a shared understanding. [SG89, St99]

Bibliography

[Ba06] Bank for International Settlements:. *Basel II: Revised international capital framework*. http://www.bis.org/publ/bcbsca.htm (cited 2008-09-15). 2006.

[Bu96] Buschmann, R. et al.:. *Pattern-Oriented Software Architecture*. John Wiley & Sons. 1996.

[Bu08a] Buckl, S. et al.:. *Enterprise Architecture Management Patterns – Exemplifying the Approach*. In *The 12th IEEE International EDOC Conference (EDOC 2008)*. München. 2008. IEEE Computer Society.

[Bu08b] 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. 2008.

[Co08] Compliance Magazin:. *Was ist Compliance?* http://www.compliancemagazin.de/compliancelexikon/wasistcompliance210708.html (cited 2008-08-09). 2008.

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

[De08] Department of Defense (DoD) USA:. *DoD Architecture Framework Version 1.5: Volume I: Definitions and Guidelines*. http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_I.pdf (cited 2008-03-19). 2008.

[Di08] Dictionary.com:. *Definition of compliance*. http://dictionary.reference.com/browse/compliance (cited 2008-08-09). 2008.

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

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

[Eu07] European Comission:. *Solvency II: EU to take global lead in insurance regulation.* `http://europa.eu/rapid/pressReleasesAction.do?reference=IP/07/1060\&format=HTML` (cited 2008-08-09). 2007.

[Gl08] Glenfis AG:. *ITIL Glossary.* `http://www.itil.org/en/shortcuts/glossarinhalt/glossarc.php` (cited 2008-10-08). 2008.

[HW05] Hafner, M.; Winter, R.:. *Vorgehensmodell für das Management der unternehmensweiten Applikationsarchitektur.* In (Ferstl, O. K. et al., Ed.): *Wirtschaftsinformatik.* pages 627–646. Physica-Verlag. 2005.

[ID06] IDS Scheer AG:. *ARIS Platform Method ARIS 7.0.* IDS Scheer AG. Saarbrücken. 2006.

[Jo05] Jonkers, H. et al.:. *A language for enterprise modelling.* In (Lankhorst, M., Ed.): *Enterprise Architecture at Work.* Springer. Berlin, Heidelberg, New York. 2005.

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

[Kr05] Krcmar, H.:. *Informationsmanagement.* Springer. Berlin. 4 edition. 2005.

[LW04] Langenberg, K.; Wegmann, A.:. *Enterprise Architecture: What Aspect is Current Research Targeting?* Technical report. Laboratory of Systemic Modeling, Ecole Polytechnique Fédérale de Lausanne. Lausanne. 2004.

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

[Of08] Office of Management and Budget:. *FEA Consolidated Reference Model Document Version 2.3.* `http://www.whitehouse.gov/omb/egov/documents/FEA_CRM_v23_Final_Oct%_2007.pdf` (cited am 2008-10-09). 2008.

[OM05] OMG:. *Unified Modeling Language: Superstructure, version 2.0, formal/05-07-04.* 2005.

[Re08] Regierungskomission Deutscher Corporate Governance Kodex:. *Deutscher Corporate Governance Kodex.* `http://www.corporate-governance-code.de/` (cited 2008-08-09). 2008.

[RMRH06] Romeike, F.; Müller-Reichart, M.; Hein, T.:. *Die Assekuranz am Scheideweg – Ergebnisse der ersten Benchmark-Studie zu Solvency II. Zeitschrift für Versicherungswesen.* 2006.

[Ro03] Ross, J. W.:. *Creating a Strategic IT Architecture Competency: Learning in*

Stages. MIS Quarterly Executive. 2(1). 2003.

[Se02] Senate of the United States of America:. *Public Company Accounting Reform And Investor Protection Act Of 2002.* Technical report. The Library of Congress – Congressional Record. Washington. 2002.

[se05] sebis:. *Enterprise Architecture Management Tool Survey 2005.* Technical report. München. 2005.

[SG89] Star, S. L.; Griesemer, J. R.:. *Institutional Ecology: "Translations" and Coherence: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology. Social Studies of Science.* Vol. 19(3):pages 387–420. 1989.

[St99] Strübing, J.:. *Soziale Welten – Arenen – Grenzobjekte.* In *29. Kongress der Deutschen Gesellschaft für Soziologie (Sektion Wissenschafts- und Technikforschung).* Freiburg. 1999.

[Th08] The Open Group:. *TOGAF 8.* `http://www.opengroup.org/togaf/` (cited 2008-03-19). 2008.

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

[Za08] Zachman Institute for Framework Advancement:. *Enterprise Architecture: A Framework.* `http://www.zifa.com/framework.pdf(cited2008-03-19)`. 2008.

List of Acronyms

CMDB

A Database used to manage Configuration Records throughout their Lifecycle. The CMDB records the Attributes of each CI, and Relationships with other CIs. A CMDB may also contain other information linked to CIs, for example Incident, Problem or Change Records. The CMDB is maintained by Configuration Management and is used by all IT Service Management Processes. [Gl08]

EAMPC

The objective of the EAM Pattern Catalog is to complement existing Enterprise Architecture (EA) management frameworks, which provide a holistic and generic view on the problem of EA management, and to provide additional detail and guidance needed to systematically establish EA management in a step-wise fashion within an enterprise. [Bu08b]

IT

The term IT is used in a wide variety. But it contains the whole for storage, processing and communication needed resources as well as the way, as these resources are organized. [Kr05]

sebis

sebis is the chair for Software Engineering for Business Information Systems at the Institute for Informatics of the Technische Universität Müunchen. sebis is headed by Professor Dr. Florian Matthes.

UML

The Unified Modeling Language is defined by the OMG [OM05] as a general-purpose modeling language in the field of software engineering.

List of Figures