



Technische Universität München
Fakultät für Informatik

Bachelor's Thesis in Wirtschaftsinformatik

**Evaluation and Extension of the EAM Pattern
Catalog in a German Insurance Company**

**Evaluierung und Erweiterung des EAM Pattern
Catalog in einer deutschen Versicherung**

Katharina Theresa Pflügler

Supervisor:	Prof. Dr. rer. nat. Florian Matthes
Advisor:	Alexander M. Ernst
Submission Date:	17.11.2008



I assure the single handed composition of this bachelor's thesis only supported by declared resources.

Rohrbach, 17.11.2008

.....

Abstract

In recent years, Enterprise Architecture (EA) management has emerged to one of the major challenges for enterprises. The increasing complexity of business transactions, the accelerated rate of change in business models, the growing regulatory framework and the increasing dependency on information technology demand a continuous alignment of business and information technology, which is the aim of EA management. Several EA management frameworks exist, however, they are often inapplicable in practice as they are either too extensive or too abstract. The EAM Pattern Catalog approach tries to address those deficiencies by providing different types of EAM Patterns which are based on best and proven practices.

Within the scope of this thesis, the EAM Pattern Catalog approach is applied for developing an EA management approach for a German insurance company. Thereby, additional Concerns are investigated and the EAM Pattern basis is extended accordingly with additional V- and I-Patterns. In a further step the set of identified and additionally developed I-Patterns is integrated to an organization-specific information model. The experience gained from these tasks forms the basis of an evaluation of the EAM Pattern Catalog approach.

Table of Contents

Table of Contents	IV
Table of Figures	V
Index of Tables	VI
1 Introduction and Overview	1
1.1 Motivation	1
1.2 Objectives of the Thesis	1
1.3 Course and Structure of the Thesis	1
1.4 Environment of the Thesis	2
1.5 Term Definitions	2
1.6 EAM Pattern Catalog	3
2 Investigation and Analysis of the Current and Former Approach	7
2.1 Former Approach	7
2.2 Current Approach	8
3 Applying and Extending the EAM Pattern Catalog	10
3.1 Investigation of Existing Concerns	10
3.2 Identifying Respective EAM Patterns	12
3.2.1 M-Patterns	14
3.2.2 V-Patterns	14
3.2.3 I-Patterns	15
3.3 Extending the EAM Pattern Catalog	18
3.3.1 Concerns	18
3.3.2 V-Patterns	18
3.3.2.1 Time Interval Map visualizing Lifecycles of Business Applications	19
3.3.2.2 Business Application Component Hierarchy	21
3.3.2.3 Business Application Component Hierarchy including Interfaces	23
3.3.2.4 Process Support Map for Products	25
3.3.2.5 Protection Requirements of Business Applications	27
3.3.2.6 Business Applications' State of Health	29
3.3.2.7 Cost Shares of Products in Business Applications	31
3.3.3 I-Patterns	32
3.3.3.1 Business Application Lifecycles	32
3.3.3.2 Interfaces and Information Flows	34
3.3.3.3 Business Application Component Hierarchy	36
3.3.3.4 Business Process Support for Products	38
3.3.3.5 Protection Requirements of Business Applications	40
3.3.3.6 Business Applications' State of Health	42
3.3.3.7 Cost Shares of Products in Business Applications	44
3.3.4 Relationships of additional EAM Pattern Catalog Elements	45
3.4 Integrating I-Patterns	47
4 Mapping Information Models	56
4.1 Current Approach Information Model	57
4.2 Iteraplan Information Model	58
5 Evaluation of the EAM Pattern Catalog Approach	60
6 Recapitulation and Prospects of the Thesis	62
6.1 Recapitulation	62
6.2 Prospects	62
A. List of Abbreviations	64
B. Bibliography	64

Table of Figures

Figure 1: UML class diagram describing the structure of the EAM Pattern Catalog. Source: [Bu08a].....	4
Figure 2: UML class diagram describing the relationships between Forces, Problems and EAM Patterns	5
Figure 3: Implementing an <i>EA Management Approach</i> based on <i>EA Management</i> patterns. Source: [Bu07]	6
Figure 4: Information Model Former Approach	7
Figure 5: Information Model Current Approach.....	8
Figure 6: Types of EAM Pattern Catalog elements	12
Figure 7: Types of relationships between EAM Patterns. Source: [Bu08a]	12
Figure 8: Overview graph containing relevant Concerns and deduced EAM Patterns.....	13
Figure 9: Exemplary view for <i>Time Interval Map visualizing Lifecycles of Business Applications and Business Application Versions</i> . Source: [Bu08a].....	19
Figure 10: Exemplary view for <i>Time Interval Map visualizing Lifecycles of Business Applications</i>	20
Figure 11: Exemplary view for <i>Business Application Component Hierarchy</i>	21
Figure 12: Exemplary view for <i>Business Application Component Hierarchy including Interfaces</i>	23
Figure 13: Exemplary view for <i>Process Support Map for Products</i>	25
Figure 14: Exemplary view for <i>Protection Requirements of Business Applications</i>	27
Figure 15: Exemplary view for <i>Business Applications' State of Health</i>	29
Figure 16: Exemplary view for <i>Cost Shares of Products in Business Applications</i>	31
Figure 17: Information model fragment for <i>Business Application Lifecycles</i> . Source: [Bu08a]	32
Figure 18: Information model fragment for variant of <i>Business Application Lifecycles</i>	33
Figure 19: Information model fragment for <i>Interfaces and Information Flows</i> . Source: [Bu08a].....	34
Figure 20: Information model fragment for variant of <i>Interfaces and Information Flows</i>	35
Figure 21: Information model fragment for <i>Business Application Component Hierarchy</i>	36
Figure 22: Information model fragment for <i>Business Process Support for Products</i>	38
Figure 23: Information model fragment for <i>Protection Requirements of Business Applications</i>	40
Figure 24: Information model fragment for <i>Business Applications' State of Health</i>	42
Figure 25: Information model fragment for <i>Cost Shares of Products in Business Applications</i>	44
Figure 26: Additional Types of relationships between EAM Patterns.....	45
Figure 27: Overview graph containing additionally developed EAM Pattern Catalog elements and related V-Patterns of the EAM Pattern Catalog	46
Figure 28: Integration Process.....	48
Figure 29: First Step Information Model	49
Figure 30: Second Step Information Model with highlighted EAM Pattern Catalog I-Patterns	52
Figure 31: Third Step Information Model with highlighted additional I-Patterns	53
Figure 32: Fourth Step Information Model	55
Figure 33: Source Model.....	57
Figure 34: Target Model 1	57
Figure 35: Target Model 2	58

Index of Tables

Table 1: Relevant Concerns grouped by EA Management Topics	11
Table 2: Comparison of total number of Concerns in the EAM Pattern Catalog and number of relevant concerns per EA management topic	11
Table 3: Deducted M-Patterns grouped by EA Management Topics	14
Table 4: Deducted V-Patterns	15
Table 5: Deducted I-Patterns.....	16
Table 6: Additional Concerns extending the EAM Pattern Catalog	18
Table 7: Matching Classes of Source Model and Target Model 2.....	59

1 Introduction and Overview

1.1 Motivation

In recent years companies have to cope frequently with the impacts of changes in their environment, which are according to [FAW07] due to challenges such as the increasing complexity of business transactions, the accelerated rate of change in business models, the growing regulatory framework as well as the increasing dependency on information technology. This forces the companies to continuously adapt their corporate strategies and to align their enterprise architecture to those strategic goals.

Organizations tend to have application landscapes which are the result of decades of one-on-one implementations of individual solutions [PS05]. Moreover, medium size or large organizations have a few hundred up to a few thousand business applications, which form the application landscape [Er08]. Thus documenting and managing the enterprise architecture is an advanced subject and therefore a structured approach must be taken in order to effectively introduce and maintain EA management. Such a structured approach is provided by the EAM Pattern Catalog which was developed at the chair for Software Engineering for Business Information Systems (sebis) of Prof. Dr. Florian Matthes at the Technische Universität München. In the course of this thesis the EAM Pattern Catalog was applied in practice in order to develop practice-oriented extensions and furthermore to evaluate the EAM Pattern Catalog approach.

1.2 Objectives of the Thesis

The application and evaluation of the EAM Pattern Catalog in practice is the main objective of this thesis. Thereby, the EAM Pattern Catalog is being assessed with regard to its usability, which means that it is tested whether the EAM Pattern Catalog is sufficient on its own or additional literature or assistance is required. Moreover, a goal is to identify potential problems or inconsistencies which might arise from applying the EAM Pattern Catalog approach. A further aim is to investigate the practice orientation of the EAM Pattern Catalog elements, for instance, it is examined to what extent the Concerns comply with the actual pain points of the company. In addition, this thesis is aiming at identifying potential difficulties in integrating EAM Patterns.

1.3 Course and Structure of the Thesis

The subsequent sections of the first chapter serve as an introduction to the topic of enterprise architecture management. Firstly, the environment of the thesis is described, followed by some definitions of main terms in order to provide a common terminology for the rest of the thesis. The last section of Chapter 1 presents the EAM Pattern Catalog [Bu08a], which constitutes the foundation of this thesis.

Chapter 2 is concerned with the investigation and analysis of the former and current enterprise architecture management approach of the insurance company. The information models of the two approaches are depicted and some information is given on the user interface as well as on implementation aspects.

Chapter 3 shows in Sections 3.1 and 3.2 how the EAM Pattern Catalog was used to determine the insurance company's Concerns and the respective EAM Patterns. Section 3.3 is concerned with developing additional EAM Pattern Catalog elements, which are Concerns (see 3.3.1), V-Patterns (see 3.3.2) and I-Patterns (see 3.3.3). The subsequent Section 3.3.4 describes the relationships of the additional EAM Pattern Catalog elements among each other and to elements of the EAM Pattern Catalog. And finally, Section 3.4 shows how an organization-

specific information model evolves from a set of already existing I-Patterns of the EAM Pattern Catalog and the I-Patterns that were developed in the course of this thesis.

At the beginning of Chapter 4 a general definition of mapping and its kinds is given. Subsequently the information model developed in Section 3.4 is first mapped to the information model of the current approach. Section 4.1 is followed by the mapping of the developed information model to the information model of the iteraplan EA management tool.

Chapter 5 evaluates the EAM Pattern Catalog approach and the thesis concludes with Chapter 6, which gives a short recapitulation of thesis and an outlook on future research directions.

1.4 Environment of the Thesis

The thesis was developed in cooperation with a German insurance company. The people involved in the development process of the EA management approach were a team of IT architects with their chief IT architect and a management advisor. The private insurance sector in Germany employs 460.000 people and has a total revenue of over 162 billion Euros¹. This sum makes up 7.5% of Germany's gross domestic product. In total the Germans have concluded 430 million insurance policies, which are over 5 insurance policies per capita. Germany's insurance industry can be subdivided into two sectors. On the one hand there is the social insurance sector which comprises the statutory pension insurance, the statutory health insurance, the statutory nursing care insurance, the statutory accident insurance and the statutory unemployment insurance. On the other hand there is the private insurance sector, which offers three classes of insurance: private pension and life insurance, private health insurance, indemnity and accident insurance. The latter class can further be subdivided into motor insurance, general liability insurance, property insurance and many more.

1.5 Term Definitions

The terms *EA management*, *EAM Pattern*, *software map* and *information model* will be defined in this section in order to provide a common understanding of those main terms of the thesis.

In this thesis *EA management* is abbreviated with EAM and is defined as follows:

“EA 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, but also business processes, business goals, strategies, etc. are considered in order to build a holistic and integrated view on the enterprise” [Er06].

As EA management considers both IT and business strategy, it should on the one hand be aligned to business strategy and on the other hand IT should take up the role of an enabler for the business strategy. Thus the alignment of business and IT can be seen as one of the main goals of EA management. A critical success factor for EA management is tool support as such an extensive and complex task can be better achieved by employing adequate EA management tools which provide, for instance, distributed access to consistent data [Er06].

¹ The quantity figures were taken from: <http://www.gdv.de/DatenUndFakten/Gesamtmarkt/inhalt.html>, accessed 10.11.08

A further very important term in this thesis is *EA management pattern* (abbr. EAM pattern), which is defined according to [Bu07]:

“**EA management patterns** [are] building blocks for the concept of an organization-specific support for EA management, consisting of a conceptual information model, viewpoints and methodologies for using the respective information”.

EA management patterns have the characteristics of being organization-independent and based on best and proven practices. They can be combined in one or several catalogs, which may constitute a starting point for implementing an EA management approach based on EA management patterns [Bu07].

The term *software map* is according to [Wi07] defined as follows:

A **software map** is a graphical model for documenting the architecture of the application landscape or of parts of the latter. A software map is composed of a base map and of layers, which build on the base map and visualize different characteristics.

Finally, a definition of the term *information model* in the context of EA management is required:

“An EA management **information model** [is] a model which specifies, which information about the enterprise architecture, its elements and their relationships should be documented, and how the respective information should be structured” [Bu07].

1.6 EAM Pattern Catalog

The objective of the EAM Pattern Catalog is to complement the existing EA management frameworks by providing additional detail and guidance for establishing an organization-specific EA management. It strikes a balance between a green field approach and a predefined approach given for instance by some EA management tools and frameworks. The EAM Pattern Catalog is composed of management concerns, management methodologies, viewpoints and information models which are given, except for management concerns, in the form of EA management patterns. The current version of the EAM Pattern Catalog comprises 43 concerns, 20 methodologies, 53 viewpoints and 47 information model fragments. Concerns and methodologies were clustered according to EA management topics in order to facilitate a better navigation and search. The following *EA management topics* were determined in [Bu08a]:

- *Technology Homogeneity* is concerned with analyzing and managing whether the application landscape is homogeneous with respect to technologies and architectures.
- *Business Processes* analyzes and manages the interactions of business applications, business processes and related entities at a high level of abstraction.
- *Application Landscape Planning* is concerned with analyzing the structure and planning the evolution of the application landscape. The focus lies on current, planned and target landscapes.
- *Support of Business Processes* analyzes and manages the IT support of business processes.
- *Project Portfolio Management* is concerned with planning and monitoring the portfolio of projects changing the application landscape.
- *Infrastructure Management* analyzes and manages the technical infrastructure on which the application landscape relies.

- *Interface, Business Object, and Service Management* is concerned with analyzing and finding services. Important analysis aspects are for instance data flows due to communication via services and business objects which are exchanged via service interfaces.

The conceptual UML class diagram in Figure 1 shows the relationships between the three EAM pattern types and Concerns. *Methodology patterns* feature an overview section, a problem section, a solution section and an optional consequence section. In the overview section id, name, alias and version information plus a summary of the M-Pattern is given. The problem section contains statements about the intended usage and furthermore the Concerns addressed by this M-Pattern are listed. A Methodology pattern's solution section specifies the V-Patterns used by the M-Pattern and defines the steps which have to be taken in order to address the Concerns. The optional consequence section mostly gives a hint at the annual data collection effort.

Viewpoint Patterns are used by M-Patterns and are composed very similar to M-Patterns. The overview section is identical whereas the problem section is missing in this EAM Pattern type. The solution section of a V-Pattern contains an exemplary viewpoint together with some textual explanations and references to the I-Patterns of which the information is visualized.

The structure of *Information Model Patterns* is again very similar to those of the other EAM Pattern types. An I-Pattern contains an overview section and a solution section, in which an information model fragment and the definitions of the contained classes are given. The optional consequence section is infrequent and if existing it contains statements concerning modelling or integration aspects.

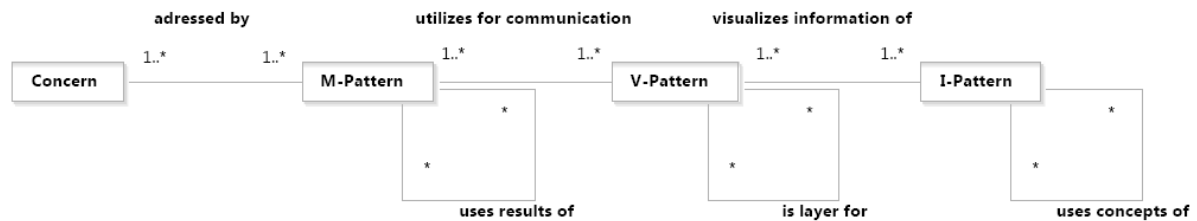


Figure 1: UML class diagram describing the structure of the EAM Pattern Catalog. Source: [Bu08a]

The structure described above was extended and adjusted in [Er08]. Concerns are no longer part of the EAM Pattern Catalog structure as they were split up in context, problems and forces. A further difference is that according to Figure 2 all types of EAM Patterns address exactly one Problem. Moreover, the EAM Patterns themselves became more extensive and self-contained as several additional sections were included. The additional sections are *example section*, *context section*, *implementation section*, *variant section*, *known uses section*, *see also section* and *credits section*. The problem section and the consequence section became compulsory for all EAM Pattern types. A further difference is that the id of the EAM Patterns became less important and consequently the EAM Patterns are now referenced by their name.

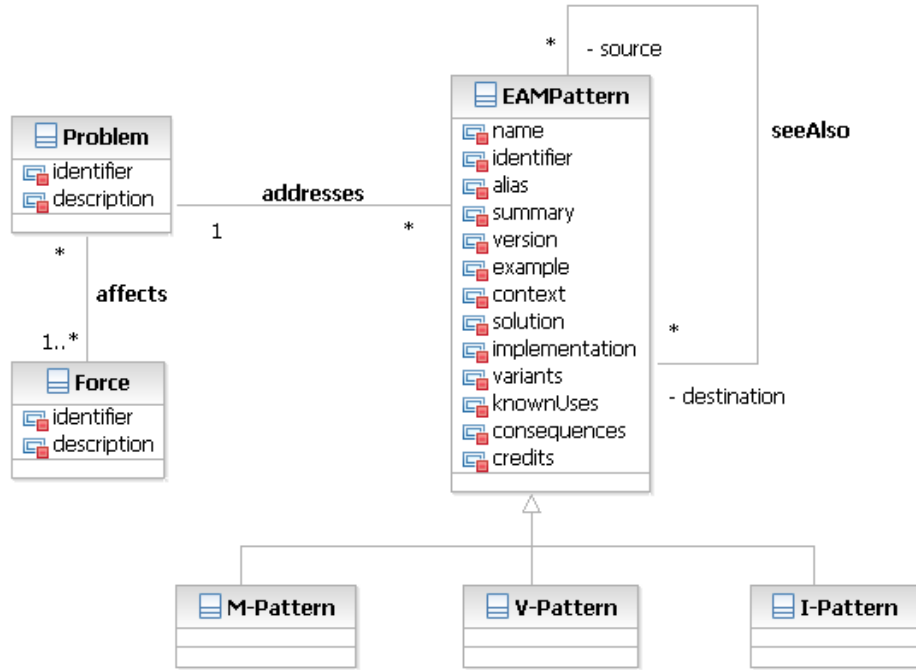


Figure 2: UML class diagram describing the relationships between Forces, Problems and EAM Patterns

The structure underlying the additionally developed EAM Patterns of this thesis is a compromise of the two structures described beforehand. The concept of Concerns remained and was not substituted by Problem. However, Concerns can now be addressed by all EAM Pattern types although the addressing by M-Patterns did not occur. The reason for this is that M-Patterns were not in the interest of the insurance company and were thus not developed in the course of the thesis. According to [Er08] the EAM Pattern themselves were supplemented with further sections such as the *see also* section, the optional *variant* section and the compulsory *consequence* section. Moreover, EAM Patterns referencing other EAM Patterns now use the name of a pattern in combination with its respective page number instead of the EAM Pattern id.

The EAM Pattern Catalog [Bu08a] states three different ways of using the EAM Patterns. The first use case is *establishing an organization-specific EA management through EAM Pattern Integration*. Firstly, all the Concerns of the company have to be identified, which is supported by the list of Concerns in Chapter 3 of the EAM Pattern Catalog. Subsequently the M-Patterns that address the selected concerns are identified by using the references given in the Concerns. Accordingly the respective V- and I-Patterns are identified. Besides the EAM Patterns of the EAM Pattern Catalog [Bu08a], patterns from other catalogs may be chosen to address a specific Concern. The last step is the integration of all identified EAM Patterns to an organization-specific EA management approach. Thereby special attention has to be paid to the conflicts and inconsistencies due to contradicting assumptions made by different EAM Patterns. M-Patterns are integrated by defining how a set of methodologies interact in order to address the selected concerns. This is achieved by creating a process model that consists of the steps to be taken for EA management. The integration of V-Patterns may seem to be the easiest integration task as V-Patterns are supposed to be fairly self-contained. As the V-Patterns of the EAM Pattern Catalog follow the layering principle originating from cartography, the integration task is to combine base maps and the so called layers, which for instance may visualize key performance indicators. Integrating I-Patterns is detailed in section 3.4 as the integration of I-Patterns was a main task of this thesis. A generalized

process on how to implement an EA management approach based on EAM Patterns shown in Figure 3.

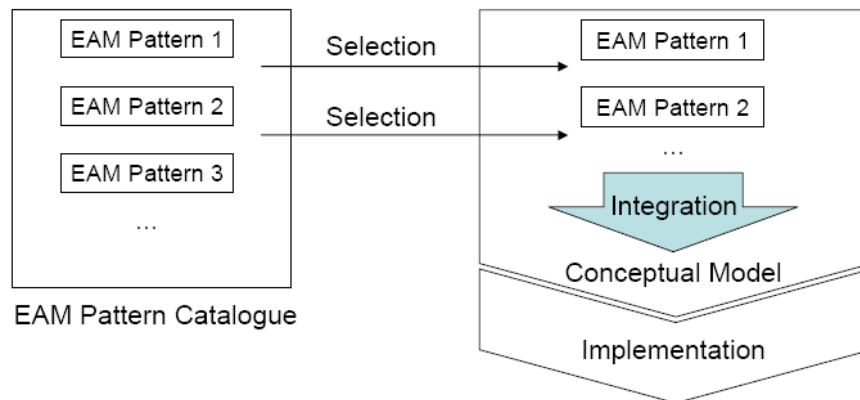


Figure 3: Implementing an EA Management Approach based on EA Management patterns. Source: [Bu07]

The second usage scenario is *inspiring and assessing an already implemented EA management approach*. Here the EAM Pattern Catalog serves as a reference book which is employed for the comparison of the own EA management approach to the best practices identified in the EAM Pattern Catalog. Thereby, M-Patterns, V-Pattern and I-Pattern that would be beneficial in the own approach are selected.

The third use case is the *EAM Pattern Catalog as a basis for academic research*. Currently, there is no EA management approach that can be iteratively extended and enhanced. This deficiency is addressed by the EAM Pattern Catalog as it facilitates the improvement and extension of single EAM Patterns without changing the overall EA management approach.

The usage scenario which was applied in this thesis can be described as a variant of the first way of using the EAM Pattern Catalog. Similarly, the Concerns were selected and were the starting point for the deduction of the respective EAM Patterns. Subsequently however, only the V- and I-Patterns were of interest and thus, only additional V- and I-Patterns were developed. At first the selected and additionally developed I-Patterns were integrated to an organization-specific information model. The resulting information model was the basis for a mapping to the information model of the already existing EA management approach in order to obtain the additional information need of the developed EA management approach. In a further mapping the developed information model was compared to the information model of an EA management tool. The purpose of this mapping was to determine to which extent the identified Concerns of the company are represented in the information model of the selected EA management tool. While integrating I-Patterns, the selected and additionally developed V-Patterns were integrated as well. After the completion of the I-Pattern integration process, some V-Patterns were no longer applicable due to the fact that their according I-Patterns were excluded during integrating. A list of still applicable V-Patterns was compiled and used as a basis for a comparison to the visualizations of the selected EA management tool. Thus, a further usage scenario of the EAM Pattern Catalog is specifying requirements for EA in order to select an EA management tool or to define the goals of an EA management approach.

The EAM Pattern Catalog itself and further information on the research project are currently available at <http://www.matthes.in.tum.de/wikis/sebis/eampc>. A wiki with the content of the EAM Pattern Catalog can be found at <http://www.systemcartography.info/eampc-wiki>.

2 Investigation and Analysis of the Current and Former Approach

In this chapter the insurance company's former approach to EA management is depicted first. This approach is characterized by a significantly larger information model in comparison to the one of the current approach, which is described in the second subchapter.

2.1 Former Approach

The former approach is based on the information model shown in Figure 4. The core of this model is an element called *Komponenten*, which is the equivalent for *BusinessApplication* of the EAM Pattern Catalog. The class *Komponenten* has a hierarchy relationship enabling the breakdown of a business application into subcomponents. Functional and technical responsibilities for certain business applications are expressed by the two relationships to the class *Verantwortliche*. The class *Anforderung* makes it possible to register requirements for the next release of a certain business application and is closely tied to the company's requirements management. *StammGruppen* enables the clustering of business applications into defined groups and thus represents the concept of domains. *Bestand* stands for portfolios of policies, which are assigned to their respective applications. The replacement of business applications can be described by the class *Abloesung* and its two associations to *Komponenten*. The first one is for the applications being replaced and the second one is for the applications replacing those. *Schnittstelle* is the equivalent of *Interface* in the EAM Pattern Catalog and has two associations to *Komponenten*. Thereby, the *von* association stands for offering and *nach* stands for using an interface.

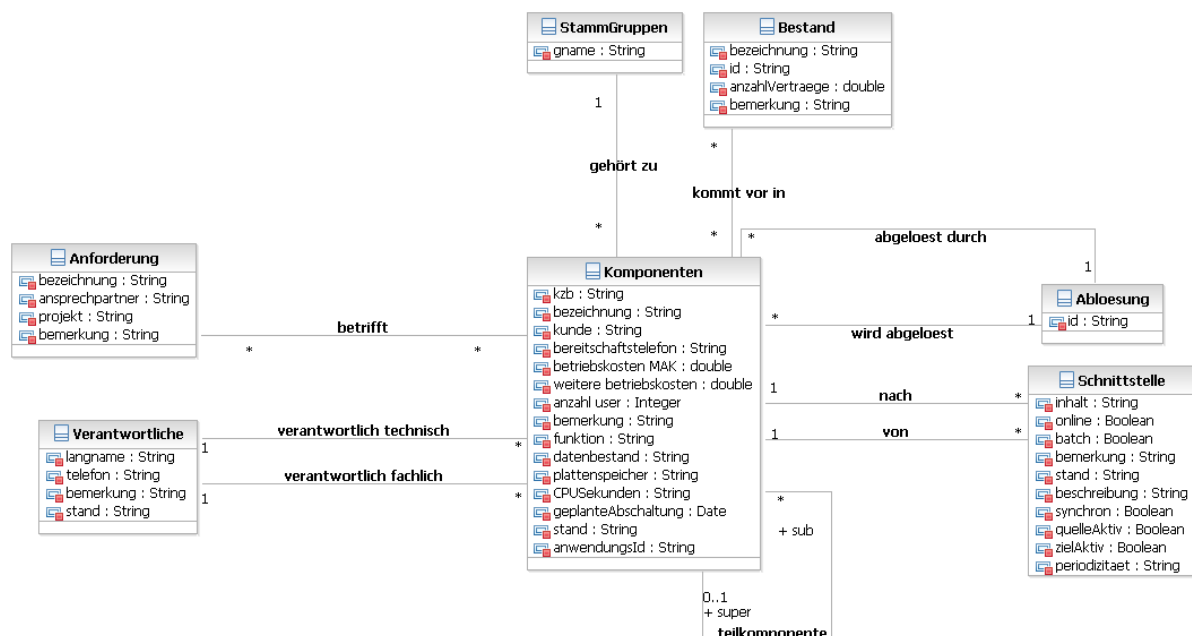


Figure 4: Information Model Former Approach

The class diagram shown in Figure 4 was deduced from the database schema of a MS Access database. This database schema contains a few more entities which were excluded as they were never filled with any information and also for reasons of clarity. The information contained in the database was collected by conducting interviews and was subsequently

exported into one big XML file. This file served as a basis for the web application, which was specifically developed for this purpose. The web application is still available in the insurance's intranet and provides several ways to access the desired information. There are lists of business applications sorted by domains, by technical responsibilities and by functional responsibilities. On selection a tabular representation of the business application containing links to the respective interfaces is displayed. Interfaces in turn are represented in a tabular form and contain links back to the respective business applications. Besides the lists of business applications a further way to access information is to visualize selected business applications together with their interfaces and the other business applications they are connected to. The visualizations are generated by using the graph editor yEd². Furthermore the web application enables the creation of a PDF-report for a selected business application. The former approach was solely used by the IT architects. On account of this no information about infrastructure or business aspects was included in the approach. The approach was created by the year 2000 and was in production until 2007 when it was decided to redefine which information is needed for architecture management. A decision was also made on the information to be published in the company's intranet. Thereby it was determined to downsize the information model of the former approach, which led to the smaller information model explained in the following subchapter.

2.2 Current Approach

The current approach is a combination of two seemingly contradictory approaches. The first part stores the information in a relational database and makes it accessible via a web representation available in the company's intranet. The second part of the current approach is a wiki available in the intranet and covering important topics. This separation emerged from a demand to have an unstructured part besides the strictly structured relational database approach. The rationale behind this is that certain information is much better conveyed using freely structured web pages rather than the predetermined form of a database schema. However, information retrieval and data analysis is hindered by such an approach.

The information model of the current approach resulted from a diminution of the former approach information model. Some parts were excluded as they were poorly filled with information and other parts were merged in order to obtain a smaller information model. Business applications, application components, portfolios of policies and domains were all united in a single class called *Objekt*. *Objekt* has a hierarchy relationship, which makes it possible to combine business applications, application components, domains and portfolios of policies. Furthermore it enables creating hierarchies for each of those four classes. *Schnittstelle* meaning interface remained, but the parts replacements, requirements and responsible persons were excluded. The resulting class diagram is shown in Figure 5.

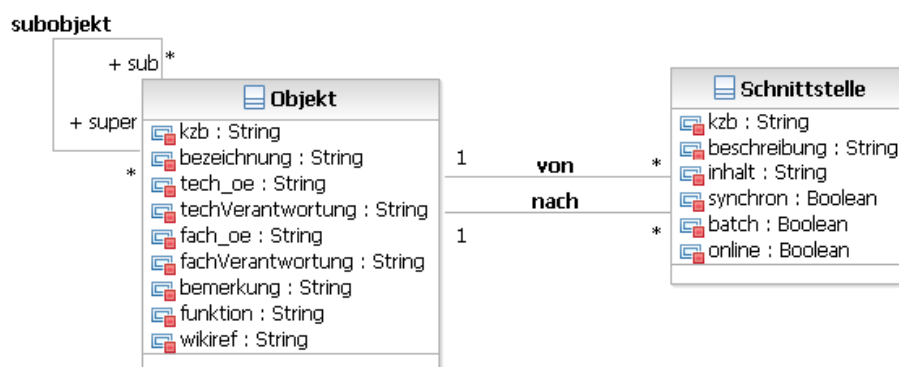


Figure 5: Information Model Current Approach

² http://www.yworks.com/en/products_yed_about.html

The basis of the current approach, namely the wiki as well as the relational part, is a MySQL database. The wiki was realized by employing TikiWiki, an open-source content management system. TikiWiki is based on PHP, ADOdb and smarty and can be used to create various kinds of web applications like sites, portals, intranets and extranets³. The wiki contains an overview of the IT architects and their responsibilities as well as an overview of important business applications. The 5 most important ones are described in detail. Links to the current and former web representations of the relational databases are included in the wiki. In addition it encompasses descriptions of the insurance company's standard architectures and a process support map containing important business applications.

The web representation of the relational database was created by using the open-source web application framework Grails, which is based on open-source technologies like Spring, Hibernate and SiteMesh⁴. As in the former approach the web application contains lists which are composed of links to the tabular representations of business applications and interfaces. There are only two lists, one for interfaces and one for objects. In this approach it is possible to select a domain in a drop-down-box and thus limit the list of applications to those of the selected domain. Unlike the former approach there is no possibility to create visualizations. However, with this web application the information may be edited and kept up to date by the IT architects.

The two parts of the current approach are connected by links in each part, which point to one another. The wiki includes a direct link to the web representation and there are links any time an interface or application is mentioned. In every tabular representation of a business application in the web representation of the relational database there is an attribute named wikiref. If there is a wiki topic on a business application, this attribute has a value which is a link to the respective page in the wiki.

³ <http://de.tikiwiki.org/tiki-index.php>, accessed 17.09.08

⁴ <http://www.oracle.com/technology/pub/articles/dev2arch/2006/10/introduction-groovy-grails.html>, accessed 17.09.08

3 Applying and Extending the EAM Pattern Catalog

The third chapter of the thesis is at first concerned with the application of the EAM Pattern Catalog. Initially, all the pain points, the so called Concerns, of the insurance company are compiled. The next step is to identify the respective EAM Patterns, namely M-Patterns, V-Patterns and I-Patterns. The subsequent Section 3.3 is concerned with the development of additional EAM Pattern Catalog elements and finally Section 3.4 describes in detail the integration of the identified EAM Pattern Catalog I-Patterns and the additionally developed I-Patterns to an organization-specific information model.

3.1 Investigation of Existing Concerns

The concerns considered relevant for the insurance company were compiled by using the EAM Pattern Catalog [Bu08a]. Chapter 3 of the Catalog gives a list of Concerns with respective M-Patterns, which address the Concerns. This list was used as a guideline for a meeting. Each individual Concern was rated as relevant or not. In the course of the meeting additional requirements came up which were reformulated as additional Concerns afterwards (see 3.3.1).

Table 1 contains the relevant Concerns with their respective M-Patterns on the right. For the purpose of clarity the Concerns were grouped according to the EA Management topics depicted in Section 1.6.

Concern	Addressed by
<i>Technology Homogeneity</i>	
C-2: Where are architectural blueprints or architectural standards used, and are there areas where those standards are breached?	M-2, M-4
C-19: Do the business applications currently used correspond to the architectural blueprints and architectural solutions (architectural standards)? If not, are there documented reasons for this, as e.g. strategic decisions?	M-4
<i>Application Landscape Planning</i>	
C-33: Which applications are used by which organizational units?	M-13
C-34: How does the long-term vision, the target of the application landscape, look like?	M-14
C-35: How does the application landscape look like at a specific date?	M-14
C-36: Which dependencies exist between business applications and are affected by current or planned projects? Which projects change the same business application? Are there changes on a business application that must be finalized before changes made by another project can be performed?	M-15
C-44: How can the operating expenses and maintenance costs be reduced, e.g. by identification of business applications providing the same functionality (redundancy)?	M-18
C-86: Which business applications are hosted by which organizational unit?	M-13
C-87: Which business processes are supported by which business application?	M-13
C-89: Which business applications will be affected by projects in the near future?	M-15
C-90: In which phase of its lifecycle is a business application at a certain point in time?	M-15
<i>Support of Business Processes</i>	
C-80: To which extent does the IT support the flexibility of the business	M-18, M-29,

processes? Where is the flexibility put at risk?	M-30
C-95: How can a more continuous IT support concerning business processes be realized?	M-29, M-30
<i>Infrastructure Management</i>	
C-41: Which infrastructure software is used by the business applications?	M-34
C-98: What is the impact of the shut-down of an infrastructure element? What other elements of the application landscape are affected?	M-34
<i>Interface, Business Object and Service Management</i>	
C-51: Which business objects are used or exchanged by which business applications or services?	M-19
C-52: What are the dependencies between the used business objects?	M-19
C-61: Which business objects are exchanged over which interfaces?	M-19
C-62: What are the domains of the application landscape?	M-20
C-64: How to find services within the development process of the application landscape?	M-20
C-65: Which services are offered by which business application?	M-20
C-66: Which business processes are supported by which services?	M-20
C-67: Which interfaces are offered/used by which business application?	M-21
C-68: What is the type, e.g. online, offline, batch, etc. of a specific interface? How is the interface implemented? What are its capabilities?	M-21
C-70: Which business applications are affected by the shut-down of an interface?	M-21
C-99: Which offered interfaces are affected by the removal of a business application?	M-21

Table 1: Relevant Concerns grouped by EA Management Topics

The relevant Concerns do not cover all EA Management topics mentioned in the EAM Pattern Catalog [Bu08a]. Table 2 shows that in the fields of Business Processes and Project Portfolio Management no Concern was considered to be relevant. Furthermore the topic of Technology Homogeneity was of little interest. On the other hand almost all of Application Landscape Planning, Infrastructure Management and Interface, Business Object and Service Management Concerns were rated as relevant.

	No. of Concerns in the EAM Pattern Catalog	No. of relevant Concerns	Percentage [%]
Technology Homogeneity	10	2	20
Business Processes	3	0	0
Application Landscape Planning	10	9	90
Support of Business Processes	3	2	67
Project Portfolio Management	3	0	0
Infrastructure Management	2	2	100
Interface, Business Object and Service Management	12	11	92

Table 2: Comparison of total number of Concerns in the EAM Pattern Catalog and number of relevant concerns per EA management topic

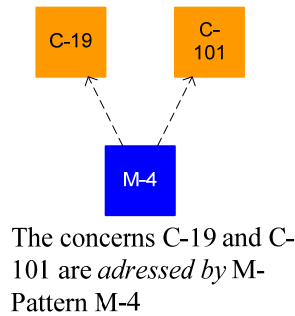
3.2 Identifying Respective EAM Patterns

After investigating the existing Concerns the EAM Pattern Catalog was consulted in order to obtain the respective M-Patterns, V-Patterns and I-Patterns. Figure 8 shows the EAM Pattern overview graph containing all identified EAM Pattern Catalog elements and their relations. Concerns and M-Patterns are grouped according to the EA management topics outlined in Section 1.6. The different types of EAM Pattern Catalog elements, which are part of the overview graph, are shown in Figure 6. The different types of edges were visualized according to [Bu08a] and are shown in Figure 7.

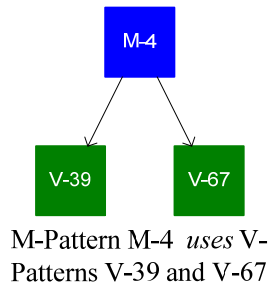


Figure 6: Types of EAM Pattern Catalog elements

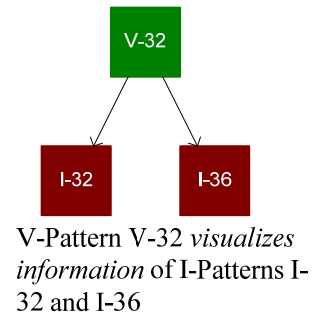
Relationship Type 1



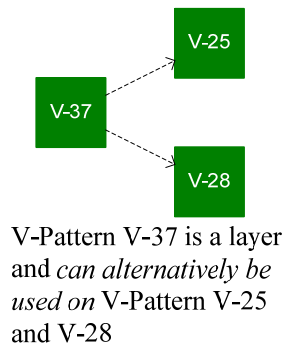
Relationship Type 2



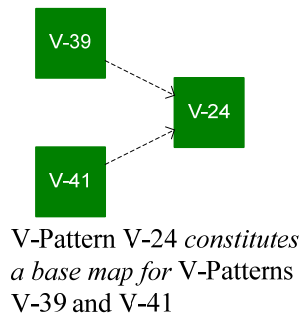
Relationship Type 3



Relationship Type 4



Relationship Type 5



Relationship Type 6

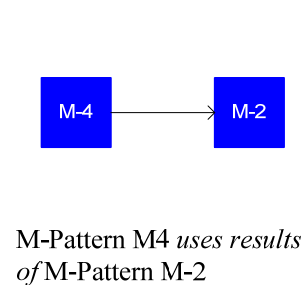


Figure 7: Types of relationships between EAM Patterns. Source: [Bu08a]

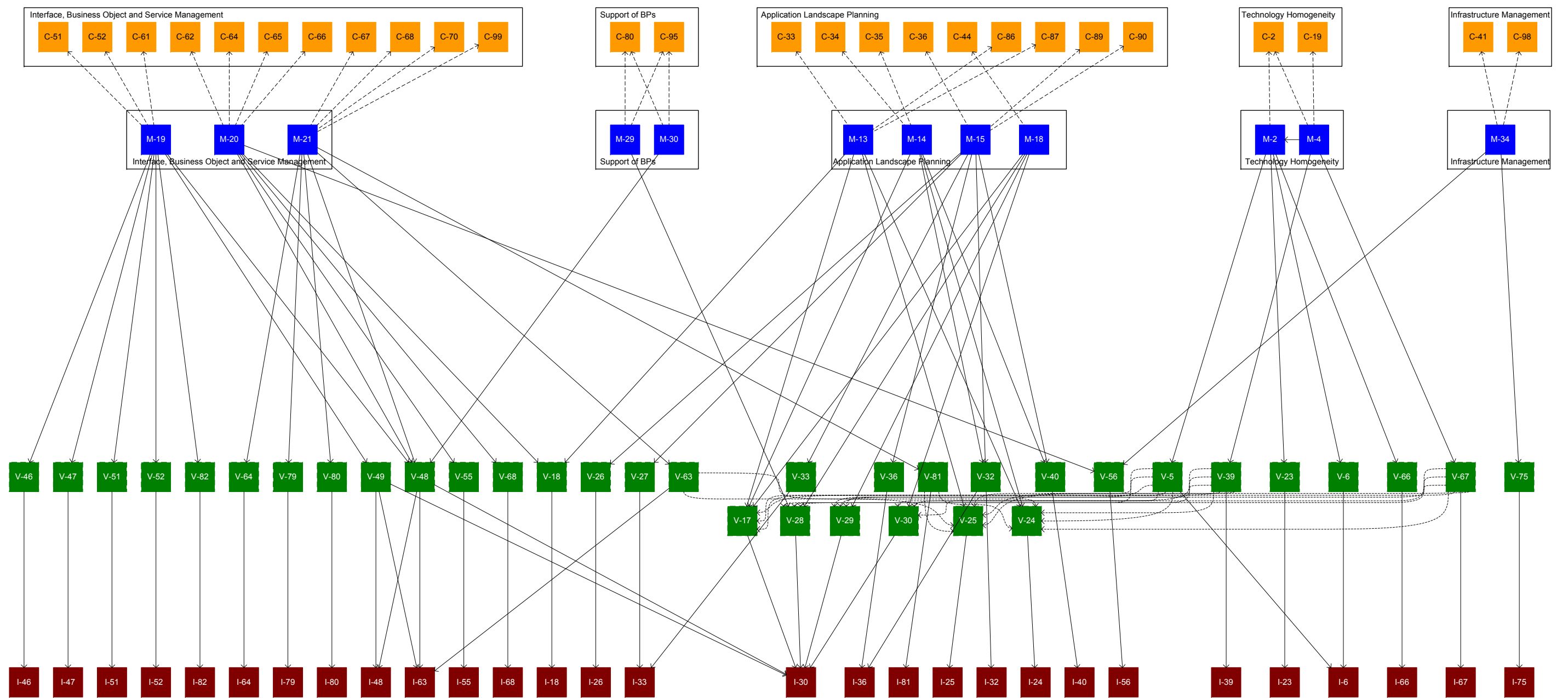


Figure 8: Overview graph containing relevant Concerns and deduced EAM Patterns

3.2.1 M-Patterns

The M-Patterns that were deducted from the Concerns outlined in Section 3.1 are shown in Table 3. In the right column of the table the associated V-Patterns are listed for each M-Pattern.

Id	Name	Uses V-Patterns
<i>Technology Homogeneity</i>		
M-2	Analysis of Standard Conformity of the Application Landscape	V-5, V-6, V-23, V-66
M-4	Management of Blueprint Conformity of the Application Landscape	V-39, V-67
<i>Application Landscape Planning</i>		
M-13	Analysis of the current application landscapes	V-17, V-18, V-24, V-25
M-14	Development of planned and target landscapes	V-17, V-24, V-32, V-40
M-15	Management of the application lifecycle	V-26, V-27, V-32, V-33, V-36, V-40
M-18	Horizontal and vertical integration	V-17, V-28, V-29, V-30
<i>Support of Business Processes</i>		
M-29	High level process support	V-28
M-30	Business Process data ow analysis	V-48
<i>Infrastructure Management</i>		
M-34	Infrastructure Failure Impact Analysis	V-56, V-75
<i>Interface, Business Object and Service Management</i>		
M-19	Management of Business Objects	V-46, V-47, V-48, V-49, V-51, V-52, V-82
M-20	Management of Business Services and Domains	V-18, V-48, V-55, V-56, V-68
M-21	Management of Interfaces	V-48, V-63, V-64, V-79, V-80, V-81

Table 3: Deducted M-Patterns grouped by EA Management Topics

3.2.2 V-Patterns

The V-Patterns were deducted from the M-Patterns in the previous section. In the right column of Table 4 the related I-Patterns are itemized for each V-Pattern.

Id	Name	Depends on I-Patterns
V-5	Standard Conformity Layer	I-6
V-6	Clustering by Standard	I-2
V-17	Process Support Map	I-30
V-18	Service-based Business Process Support Map	I-18
V-23	Technologies by Architectural Standard	I-23
V-24	Cluster Map for hosting Relationship	I-24
V-25	Cluster Map for using Relationship	I-25
V-26	Time Interval Map visualizing Lifecycles of Applications	I-26
V-27	Application Lifecycle Project Layer	I-33
V-28	Process Support Map visualizing horizontal Integration	I-30
V-29	Process Support Map visualizing vertical Integration	I-30
V-30	Process Support Map visualizing vertical and	I-30

	horizontal Integration	
V-32	Process Support Map visualizing Changes in Relations to their Time Horizon	I-32, I-36
V-33	Time Interval Map visualizing Projects and the affected Business Application	I-33
V-36	Overview over Lifecycle of Business Applications	I-36
V-39	Effects of a Project Proposal on the Application Landscape (detail)	I-39
V-40	Migration of Functionality	I-40
V-46	Business Object ER Diagram	I-46
V-47	Business Object Class Diagram	I-47
V-48	Cluster Map visualizing Business Object Flows between Business Applications	I-30, I-48, I-63
V-49	Communication Table	I-30, I-48, I-63
V-51	Process Overview	I-51
V-52	Business-level Communication Overview	I-52
V-55	Component Cluster Map	I-55
V-56	Infrastructure Usage	I-56
V-63	Information Flows	I-63
V-64	Applications and Interfaces	I-64
V-66	Architectural Solution in detail (UML)	I-66
V-67	Standard Conformity Exceptions	I-67
V-68	Process Support Map with Services	I-68
V-75	Business Application Deployments	I-75
V-79	Call Sequences	I-79
V-80	Application and Interface Migrations	I-80
V-81	Communicating Applications	I-81
V-82	Business Object Flows	I-82

Table 4: Deducted V-Patterns

3.2.3 I-Patterns

The I-Patterns deducted from the list of V-Patterns in the previous section are listed in Table 5.

Id	Name
I-6	Usage of Architectural Solutions
I-18	Services and Service Usage
I-23	Technology Usage
I-24	Hosting Business Applications
I-25	Using Business Applications
I-26	Business Application Versions
I-30	Process Support
I-32	Timed Process Support
I-33	Project Effects Business Application Version
I-36	Project Effects
I-39	Planned Proposal Effects
I-40	Functionality Migration
I-46	Business Objects and Business Object Types
I-47	Business Object Relationships and Attributes
I-48	Business Object Flow
I-51	Process Overview

I-52	High Level Information Flows
I-55	Domains and Components
I-56	Infrastructure Usage
I-63	Interfaces and Information Flows
I-64	Interfaces and Operations
I-66	Architectural Blueprints and Architectural Solutions
I-67	Demanded Architectural Solutions
I-68	Process Support by Service
I-75	Deployment Details
I-79	Call Sequences
I-80	Changing Business Applications and Interfaces
I-81	Communicating Business Applications
I-82	Interfaces and Business Objects

Table 5: Deducted I-Patterns

Some of those 29 I-Patterns are very similar and constitute modelling alternatives, which will be outlined in this section. Especially in the field of modelling interfaces many akin I-Patterns can be found. *Interfaces and Information Flows* (see page 221 in [Bu08a]) contains the class `InformationFlow`, which is a transfer of information between a server `BusinessApplication`, providing an `Interface` and a client `BusinessApplication`. *Communicating Business Applications* (see page 234 in [Bu08a]) is almost the same except for the fact that the `Interface` itself is not explicitly modelled. There are only `InformationFlow` and `BusinessApplication` acting as source and destination of the information transferred. *Interfaces and Operations* (see page 222 in [Bu08a]) supplements *Interfaces* with *Operations* and *Interfaces and Business Objects* (see page 235 in [Bu08a]) provides an association between `Interface` and `BusinessObject`. *Changing Business Applications and Interfaces* (see page 233 in [Bu08a]) proposes changing `BusinessApplications` as well as changing *Interfaces* which are realized by a `replacedBy` relationship.

In the field of *Infrastructure Usage* and *Deployment* two similar I-Patterns were identified, namely *Infrastructure Usage* (see page 217 in [Bu08a]) and *Deployment Details* (see page 229 in [Bu08a]). However, only some parts can be denoted as alike. In *Infrastructure Usage* `Device` and `SystemSoftware` are connected via an association class called `SystemSoftwareDeployment`. The corresponding classes for `Device` and `SystemSoftware` in *Deployment Details* are `HardwareSystem` and `InfrastructureSoftware` respectively. Between those two classes the class `InfrastructureSoftwareDeployment` can be found. `InfrastructureSoftwareDeployment` describes the deployment of an `InfrastructureSoftware` and runs on a `HardwareSystem`.

Similarities were also found in *Usage of Architectural Solutions* (see page 189 in [Bu08a]) and *Demanded Architectural Solutions* (see page 225 in [Bu08a]). In *Usage of Architectural Solutions* a `BusinessApplication` simply conforms to an `ArchitecturalSolution`. This topic appears in a more elaborate form in *Demanded Architectural Solutions*. There the association class `DemandedSolution` connects `BusinessApplication` and `ArchitecturalSolution`. The purpose of this class is

to indicate whether a business application conforms to a solution and if not it enforces justification why a violation was allowed.

Project Effects Business Application Version (see page 202 in [Bu08a]) and *Project Effects* (see page 204 in [Bu08a]) are almost alike. In *Project Effects* `BusinessApplication` and `Project` are connected via the association class `ProjectEffect`. In *Project Effects Business Application Version* a `Project` is connected to a `BusinessApplication-Version` and is not linked directly with the respective `BusinessApplication`.

The EAM Pattern Catalog contains several modelling alternatives concerning process support. *Process Support* (see page 199 in [Bu08a]) constitutes the basic version of this issue. `BusinessProcess`, `OrganizationalUnit` and `BusinessApplication` are assembled by a class named `SupportRelationship`. It states which `BusinessProcess` is supported by which `BusinessApplication` at which `OrganizationalUnit`. *Process Support by Service* (see page 226 in [Bu08a]) is exactly the same, only transferred to the context of services. Consequently `BusinessApplication` was replaced by the class `Service`. *Timed Process Support* (see page 201 in [Bu08a]) adds the time aspect. `BusinessApplication` and `BusinessProcess` are connected via the class `SupportRelationship`, which in this case indicates which `BusinessApplication` supports which `BusinessProcess` during which period of time.

Two similar I-Patterns were found in the field of technology usage. *Technology Usage* (see page 195 in [Bu08a]) is the simplified version. It suggests a simple using association between `ArchitecturalSolution` and `Technology`. The more elaborate *Architectural Blueprints and Architectural Solutions* (see page 223 in [Bu08a]) solves this problem similarly. Here `ArchitecturalSolution` and `Technology` are connected via the association class `Usage`. However *Architectural Blueprints and Architectural Solutions* additionally suggests modelling `ArchitecturalSolution` and `Technology` as the concretization of `ArchitecturalBlueprint` and `AbstractTechnology`. Those two classes are connected analogically by the association class `AbstractUsage`.

In the field of business object relationships two alternative modelling possibilities were identified. In *Business Object Relationships and Attributes* (see page 211 in [Bu08a]) a `BusinessObject` may have no or several `Attributes` and `RelationshipEnds`. Each `RelationshipEnd` belongs to exactly one `Relationship` and a `Relationship` may have two or more `RelationshipEnds`. *Business Objects and Business Object Types* (see page 210 in [Bu08a]) contains an extension, which allows a `Relationship` to have `Attributes`.

3.3 Extending the EAM Pattern Catalog

The extensions to the EAM Pattern Catalog which were developed in the course of the thesis are depicted in this chapter. Analogous to the EAM Pattern Catalog the Concerns are listed first. In Section 3.3.2 the additionally developed V-Patterns are presented and are followed by the subchapter containing additional I-Patterns. The last subchapter is concerned with the relation of all additional Concerns and EAM Patterns.

3.3.1 Concerns

The additional Concerns originate from a meeting with the responsible management advisor and the chief IT architect. At the beginning they were simply statements of what would be beneficial to have in an EA Management Approach and were reformulated as Concerns (see Table 6) afterwards.

Concerns

C-102: What are the costs of a business application regarding operation, maintenance and associated projects?
C-103: What are the quantity figures (number of users, number of contracts etc.) of a business application?
C-104: How many errors occur per lifecycle phase?
C-105: Which products are provided by which business applications? What cost share does a product make up?
C-106: How much turnover does a product generate?
C-107: What is the protection requirement category of a business application?
C-108: Which business applications are maintained by which organizational unit?
C-109: What is the state of health of a business application?

Table 6: Additional Concerns extending the EAM Pattern Catalog

3.3.2 V-Patterns

The V-Patterns in the following subchapters differ from the V-Patterns in the EAM Pattern Catalog [Bu08a] in the form of a more extensive structure. The EAM Pattern Catalog V-Pattern structure comprises a *V-Pattern overview*, a *solution section* and in some cases there is a *consequence section*. According to [Er08] the structure of V-Patterns in this thesis was supplemented with a *variants section* and a *see also section*. Moreover, the *consequence section* is now compulsory.

3.3.2.1 Time Interval Map visualizing Lifecycles of Business Applications

V-Pattern Overview

Name	Time Interval Map visualizing Lifecycles of Business Applications
Id	V-26
Alias	
Summary	This V-Pattern visualizes the lifecycles of business applications, including the respective versions.
Version	1.1

Solution Section:

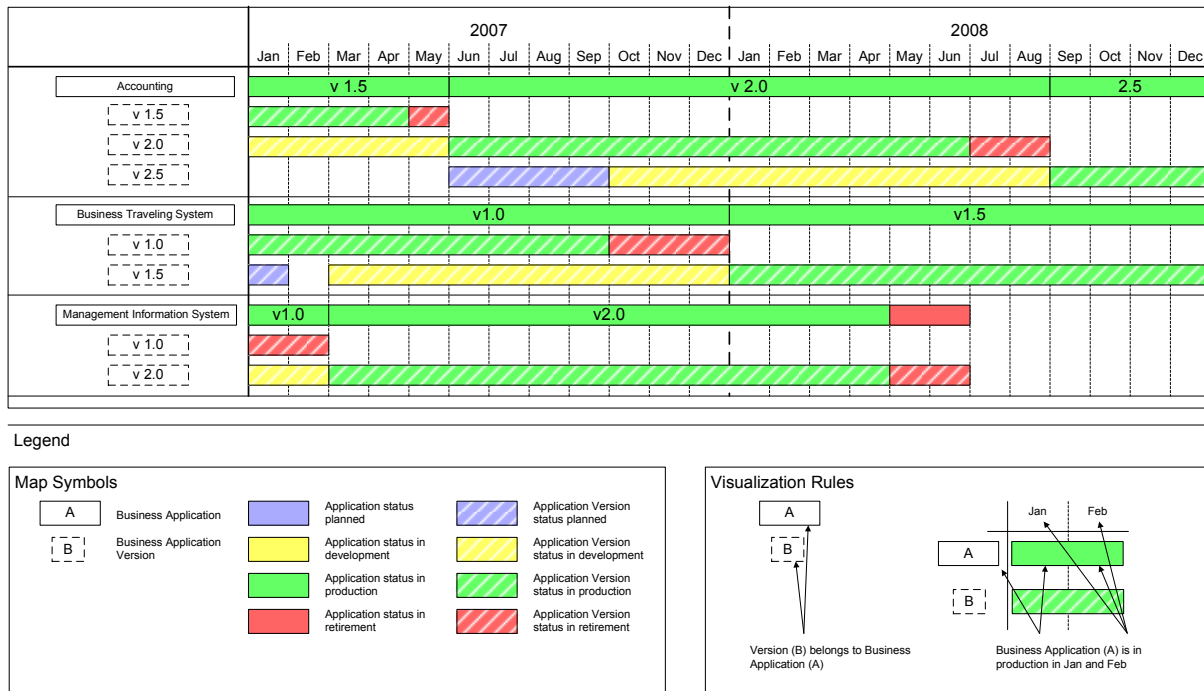


Figure 9: Exemplary view for Time Interval Map visualizing Lifecycles of Business Applications and Business Application Versions. Source: [Bu08a]

This V-Pattern belongs to the software map type *Cartesian Map*, more precisely a *Time Interval Map*. On the x-axis a certain period of time is shown and on the y-axis business applications and their respective business application versions can be found. This visualization expresses which business application or business application version is in which lifecycle phase at a certain point of time.

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.

Variants:

A variant of this V-Pattern is to leave out the business application versions and show the lifecycles only for business applications. This makes it possible to visualize the information for a bigger number of business applications while keeping the visualization clearly arranged. A further variant may be to partition the lifecycles differently and to alter their naming. An exemplary visualization is shown in Figure 10.

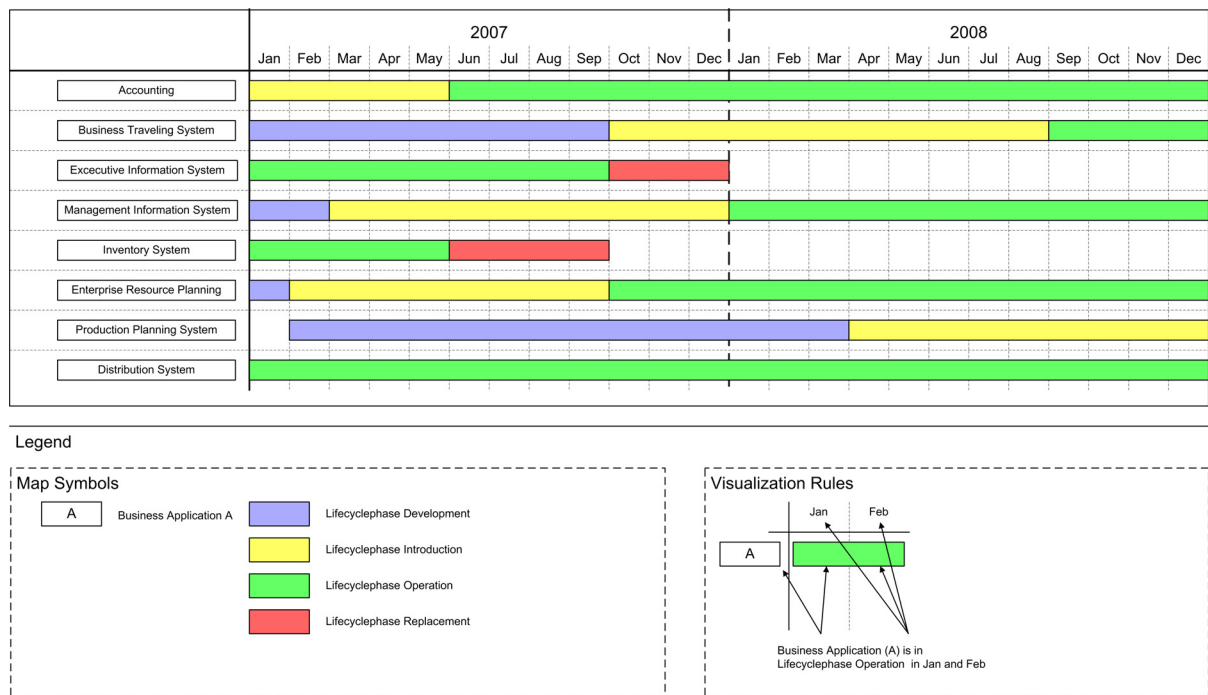


Figure 10: Exemplary view for Time *Interval Map* visualizing *Lifecycles of Business Applications*

Consequence Section:

A benefit of this V-Pattern is that it gives a concise overview of the changes to the application landscape. Moreover, it is easy to determine for a certain point of time which business application will be in which lifecycle phase and therefore this V-Pattern visualizes information which is very valuable for planning the future application landscape.

Drawbacks of this V-Pattern are costs and time involved to collect the needed data in order to be able to create visualizations. The collection of data is such an effort as for every single lifecyclephase of a business application the start and end dates have to be recorded.

See Also:

The V-Pattern is based on information according to *Business Application Lifecycles* and its respective variants (see page 32).

3.3.2.2 Business Application Component Hierarchy

V-Pattern Overview

Name	Business Application Component Hierarchy
Id	V-83
Alias	
Summary	This V-Pattern visualizes a business application with its application component hierarchy by using the concept of clustering.
Version	1.0

Solution Section:

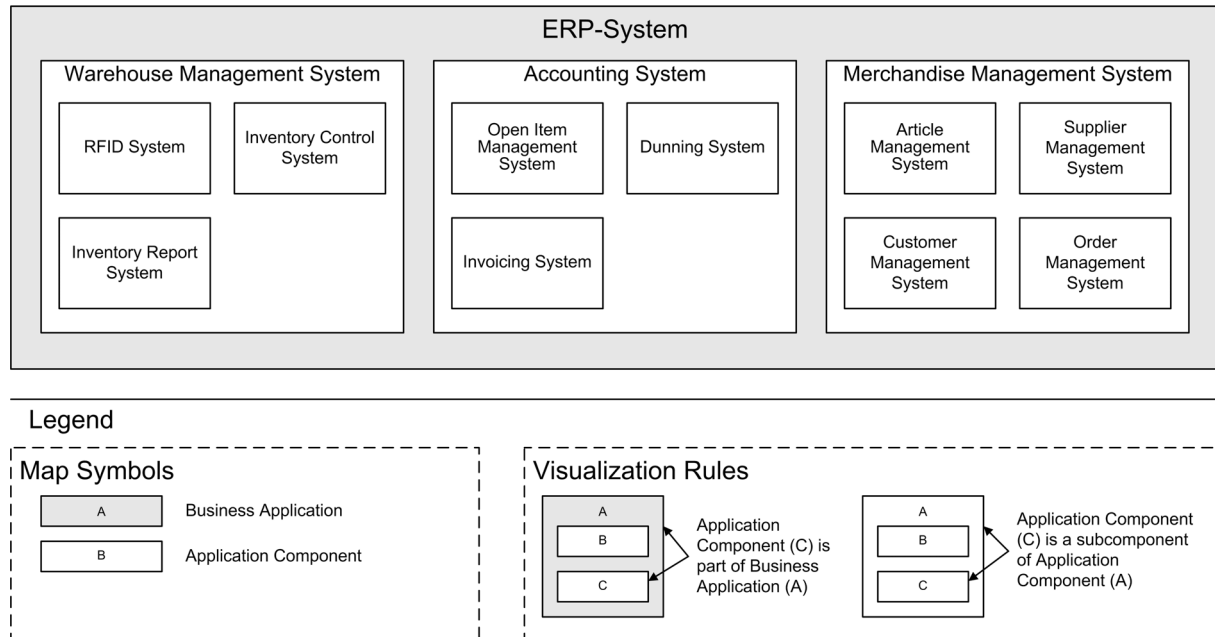


Figure 11: Exemplary view for *Business Application Component Hierarchy*

This V-Pattern is of the software map type *Cluster Map*, which uses the concept of grouping (clustering) of elements in order to express the relationships between them.

Here the concept of clustering is used to cluster application components in business applications and application components in application components. This expresses that a business application aggregates application components, which in turn may aggregate sub application components. Thus a hierarchy of business applications and its application components is visualized.

Variants:

The exemplary view shows the application component hierarchy for only one business application. A variant of this V-Pattern contains multiple business applications and their application component hierarchy, which makes it absolutely necessary to limit the number of levels of application components as otherwise the visualization would be too cluttered.

A further variant for this V-Pattern is to use a tree instead of clusters. However, this variant is only possible as long as the allocation of business applications and application components is distinct. The *root* of the tree is the business application, visualized in Figure 11 by a grey rectangle. The *nodes* of the tree stand for the application components and the *edges* between them visualize the relationships between the application components themselves and the business application. As mentioned for the previous variant, it is possible to show the hierarchy for multiple business applications, which in this case leads to several trees in one visualization.

Consequence Section:

A benefit of this V-Pattern is that creating visualizations is relatively simple and visualizations can even be created manually in a drawing tool. However, the clusters of application components can only be clearly arranged if the granularity has been chosen adequately in the data collection process. Too many levels of application components will lead to a glutted visualization, which results in a bad readability as the cognitive abilities of individuals will reach the limits.

See Also:

The V-Pattern *Business Application Component Hierarchy including Interfaces* (see page 23) is closely related. It additionally visualizes the interfaces of application components.

The visualized information is based on I-Pattern *Business Application Component Hierarchy* (see page 36).

3.3.2.3 Business Application Component Hierarchy including Interfaces

V-Pattern Overview

Name Business Application Component Hierarchy including Interfaces

Id I-84

Alias

Summary This V-Pattern visualizes a business application with its according application component hierarchy and the respective interfaces.

Version 1.0

Solution Section:

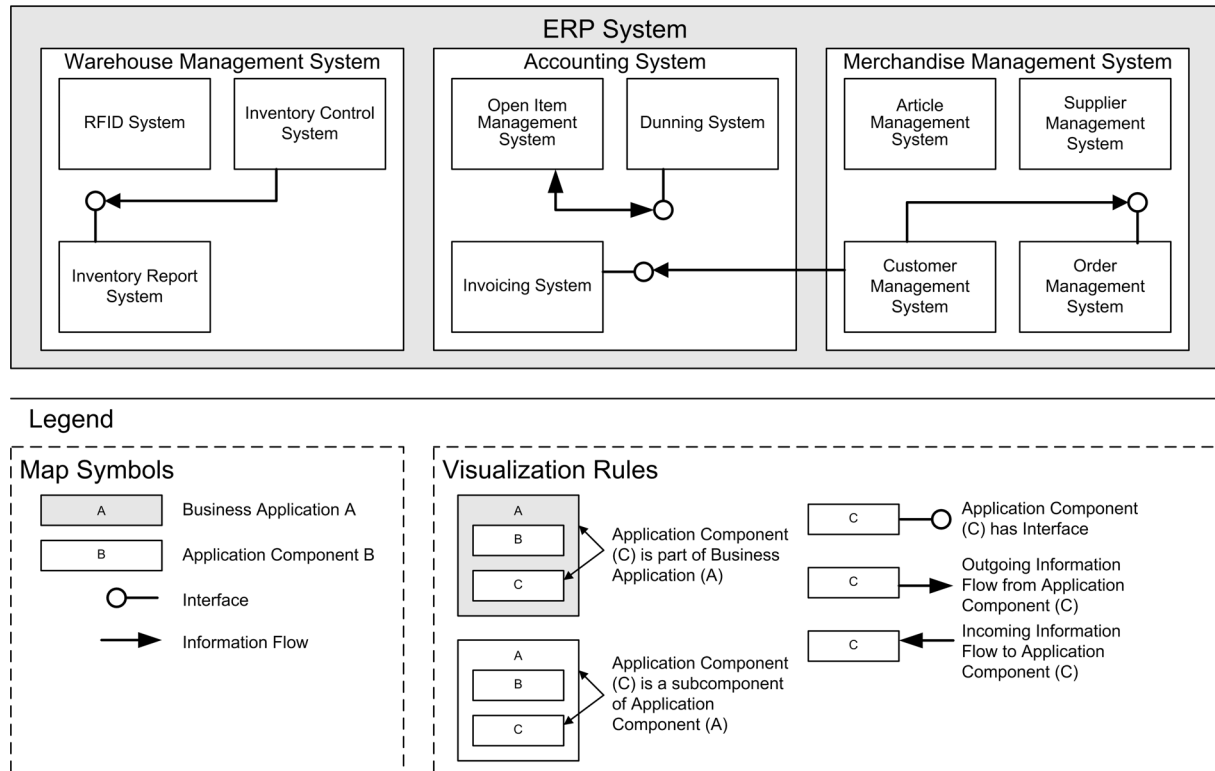


Figure 12: Exemplary view for *Business Application Component Hierarchy including Interfaces*

The V-Pattern, visualized in Figure 12 is closely related to V-Pattern *Business Application Component Hierarchy* (see page 21). The only difference is that this V-Pattern additionally visualizes interfaces, their relationships to application components and the respective information flows.

Variants:

A variant of this V-Pattern is to show the business application hierarchy with interfaces and information flows for several business applications. Thereby it is recommended to keep the number of application component levels adequately low as otherwise the visualization is likely to be cluttered.

Consequence Section:

A benefit of this V-Pattern is that creating visualizations is relatively simple and visualizations can even be created by hand. It is essential to choose an adequate granularity of application components as otherwise the visualization will contain unclearly arranged clusters of application components. If there are too many levels of application components, the visualization will be cluttered, which leads to a bad readability. The readability is also

influenced by the number of interfaces included in the visualization as with increasing the number of interfaces the probability of overlapping and crossing information flow arrows rises.

Furthermore it has to be decided on which level in the application component hierarchy the allocation of application components and interfaces is recorded. If this is not done uniformly, problems will occur when aggregating layers of the application component hierarchy.

See Also:

Another V-Pattern called *Business Application Component Hierarchy* (see page 21) is closely related. It visualizes the hierarchy of application components without interfaces.

The visualized information is based on I-Pattern *Business Application Component Hierarchy*, (see page 36) and on *Interfaces and Information Flows* (see page 34) and its variants.

3.3.2.4 Process Support Map for Products

V-Pattern Overview

Name Process Support Map for Products

Id V-85

Alias

Summary This V-Pattern visualizes the relationships between business processes, products and business applications. It shows which business application provides which product and supports which business process.

Version 1.0

Solution Section:

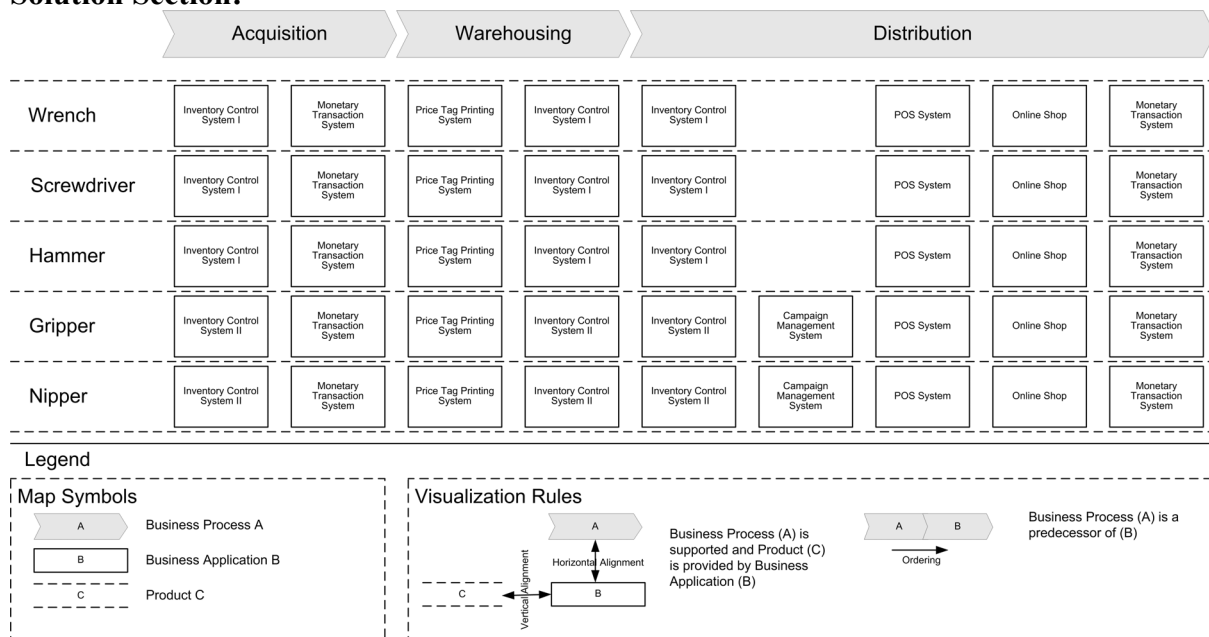


Figure 13: Exemplary view for Process Support Map for Products

This V-Pattern belongs to the software map type *Cartesian Map*, more precisely a *Process Support Map*. As this name implies, business processes are shown on the x-axis and products can be found on the y-axis.

Here the relationships between business processes, products and business applications are visualized. A business application is aligned horizontally according to the products it provides. The vertical alignment expresses which business process is supported by which business applications. A business application may appear multiple times if it is used in more than one business process or if it provides more than one product.

Consequence Section:

A benefit of this V-Pattern is that it contains a considerable amount of valuable information about the current situation of the application landscape and can therefore be used for planning the future application landscape. Due to the alignment along the axes displaying products and business processes, business applications providing redundant functionalities may be identified. On the other hand gaps in the layout of the business applications may hint at missing IT support. Consequently the visualization supports the identification of optimization potential and is very useful for planning the future application landscape.

See Also:

The V-Pattern is based on information according to I-Pattern *Business Process Support for Products* (see page 38).

Additionally, V-Pattern *Process Support Map for Products* is the basis for all V-Patterns which rely on visualizing the relationship between business processes, products and business applications together with other information. The following list gives an overview about these V-Patterns:

- *Standard Conformity Layer* (see page 99 in [Bu08a])
- *Effects of a Project Proposal on the Application Landscape* (see page 128 in [Bu08a])
- *Effects of a Project Proposal on the Application Landscape (detail)* (see page 132 in [Bu08a])
- *Cluster Map indicating standard vs. individual software* (see page 136 in [Bu08a])
- *Process Support Map, showing standard vs. individual software* (see page 140 in [Bu08a])
- *Expected Proposal Effects* (see page 157 in [Bu08a])
- *Standard Conformity Exceptions* (see page 169 in [Bu08a])
- *Technology Usage* (see page 178 in [Bu08a])
- *Protection Requirements of Business Applications* (see page 27)
- *Business Applications' State of Health* (see page 29)

3.3.2.5 Protection Requirements of Business Applications

V-Pattern Overview

Name	Protection Requirements of Business Applications
Id	V-86
Alias	
Summary	This V-Pattern visualizes the protection requirement category of a business application.
Version	1.0

Solution Section:

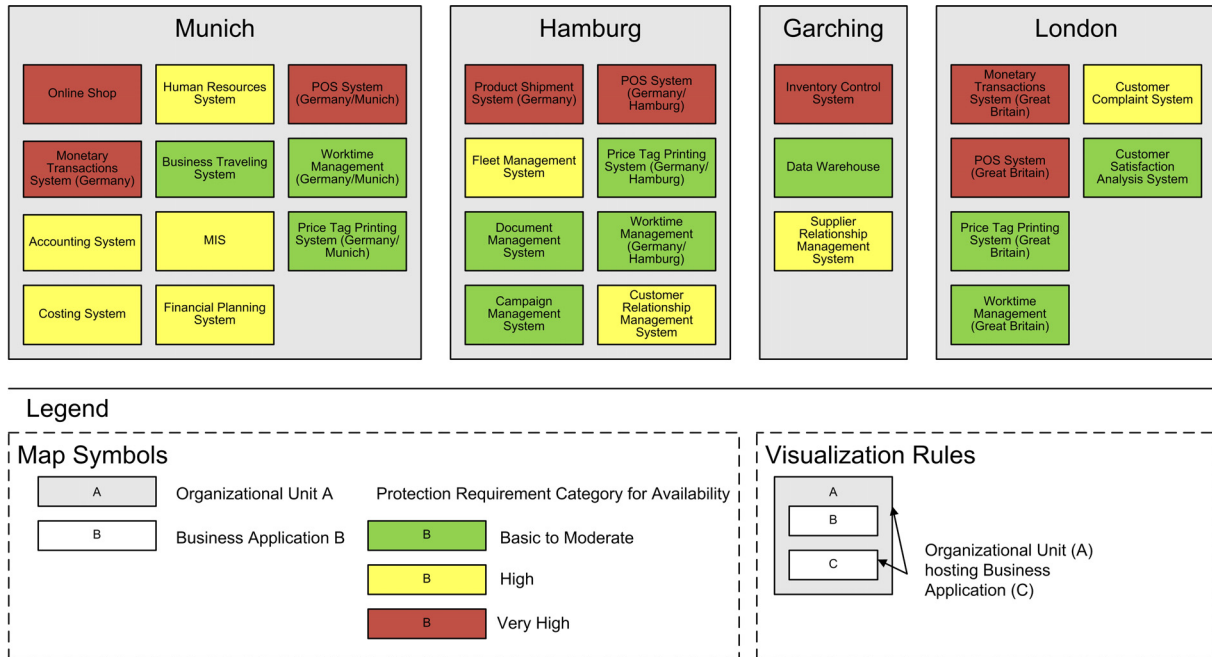


Figure 14: Exemplary view for *Protection Requirements of Business Applications*

This V-Pattern is based on the cluster map showing the relationships between business applications and organizational units of the V-Pattern *Organizational Unit Business Application Cluster Map* (see page 23 in [Er08]) and its variants. The visualization in Figure 14 visualizes the hosting relationship. Additionally, a layer shows which protection requirement category has been assigned to a business application in terms of availability. Thereby the three different colours stand for the three protection requirement categories suggested by [BSI04].

Variants:

Additional variants for this V-Pattern exist as protection requirement categories can be assigned in terms of the following aspects [BSI04]:

- Availability
- Confidentiality
- Integrity

Moreover, the V-Pattern *Organizational Unit Business Application Cluster Map* (see page 23 in [Er08]) contains multiple variants, which all constitute a basis for a different variant of this V-Pattern.

Furthermore the information which protection requirement category has been assigned to a business application can be visualized on a different software map like a Cartesian map, in

particular a process support map. The following V-Patterns of the type process support map also constitute a base map for a variant of this V-Pattern:

- *Process Support Map* (see page 105 in [Bu08a])
- *Process Support Map visualizing horizontal Integration* (see page 115 in [Bu08a])
- *Process Support Map visualizing vertical Integration* (see page 117 in [Bu08a])
- *Process Support Map visualizing vertical and horizontal Integration* (see page 119 in [Bu08a])
- *Process Support Map for Products* (see page 25)

Consequence Section:

The classification into protection requirement categories enables the achievement of adequate security levels for business applications. Furthermore, this V-Pattern makes it easy to identify the business applications with the highest potential loss or damage on failure, which may serve as input for risk management.

See Also:

Creating views based on this V-Pattern requires collecting information according to I-Pattern *Protection Requirements of Business Applications* (see page 40). Additionally, information about relationships between organizational units and business applications according to I-Pattern *Business Application and Organizational Unit Relationship* (see page 39 in [Er08]) is required for the visualization.

3.3.2.6 Business Applications' State of Health

V-Pattern Overview

Name Business Applications' State of Health

Id V-87

Alias

Summary This V-Pattern visualizes the state of health category for each individual business application.

Version 1.0

Solution Section:

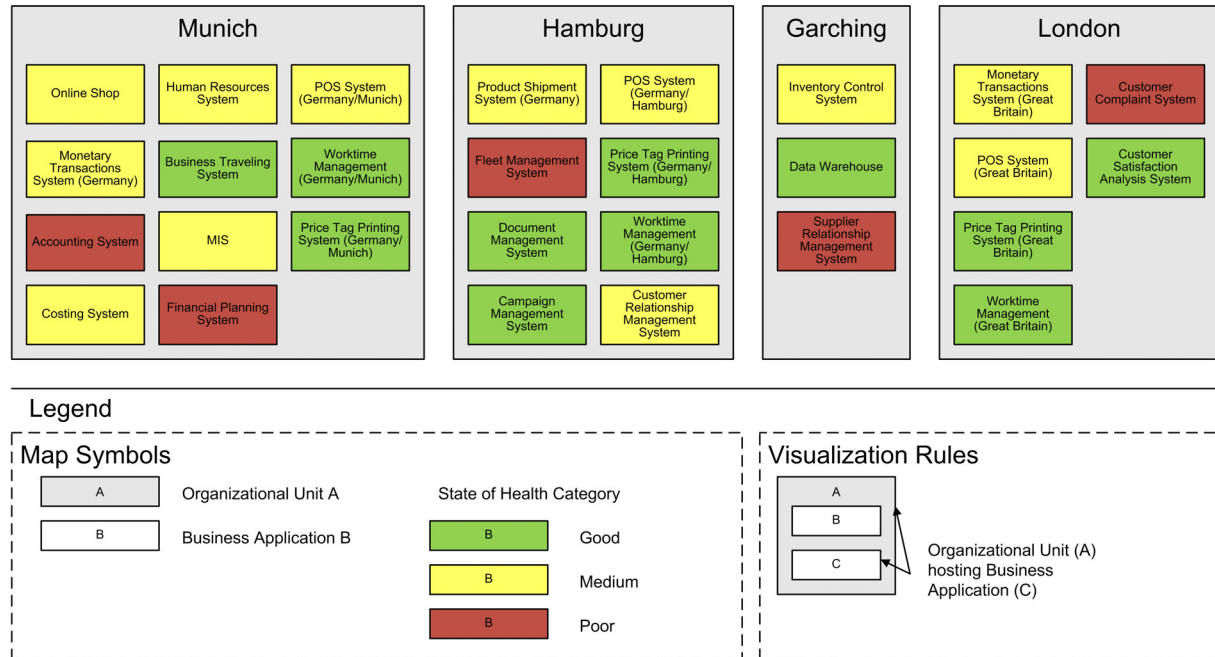


Figure 15: Exemplary view for *Business Applications' State of Health*

This V-Pattern is based on the cluster map showing the relationships between business applications and organizational units of the V-Pattern *Organizational Unit Business Application Cluster Map* (see page 23 in [Er08]) and its variants. The visualization in Figure 15 visualizes the hosting relationship. An additional layer shows a business application's state of health. Thereby the different colours stand for the different state of health categories. The exemplary view contains three different state of health categories: *good*, *medium*, *poor*.

Variants:

The V-Pattern *Organizational Unit Business Application Cluster Map* (see page 23 in [Er08]) contains multiple variants, which all constitute a basis for a different variant of this V-Pattern. Furthermore the information which state of health has been assigned to a business application can be visualized on a different software map like a Cartesian map, in particular a process support map. The following V-Patterns of the type process support map also constitute a base map for a variant of this V-Pattern:

- *Process Support Map* (see page 105 in [Bu08a])
- *Process Support Map visualizing horizontal Integration* (see page 115 in [Bu08a])
- *Process Support Map visualizing vertical Integration* (see page 117 in [Bu08a])
- *Process Support Map visualizing vertical and horizontal Integration* (see page 119 in [Bu08a])
- *Process Support Map for Products* (see page 25)

Consequence Section:

A benefit of this V-Pattern is that the business applications with a bad state of health can easily be identified by means of the red colouring. This provides a rather simple way to identify those business applications which need to be prioritized in projects in order to improve the overall state of health of the application landscape.

See Also:

The V-Pattern is based on information according to I-Pattern *Business Applications' State of Health* (see page 42). The base map showing relationships between organizational units and business applications requires the collection of information according to I-Pattern *Business Application and Organizational Unit Relationship* (see page 39 in [Er08]).

3.3.2.7 Cost Shares of Products in Business Applications

V-Pattern Overview

Name	Cost Shares of Products in Business Applications
Id	V-88
Alias	
Summary	This V-Pattern visualizes which product makes up which cost share in particular business applications.
Version	1.0

Solution Section:

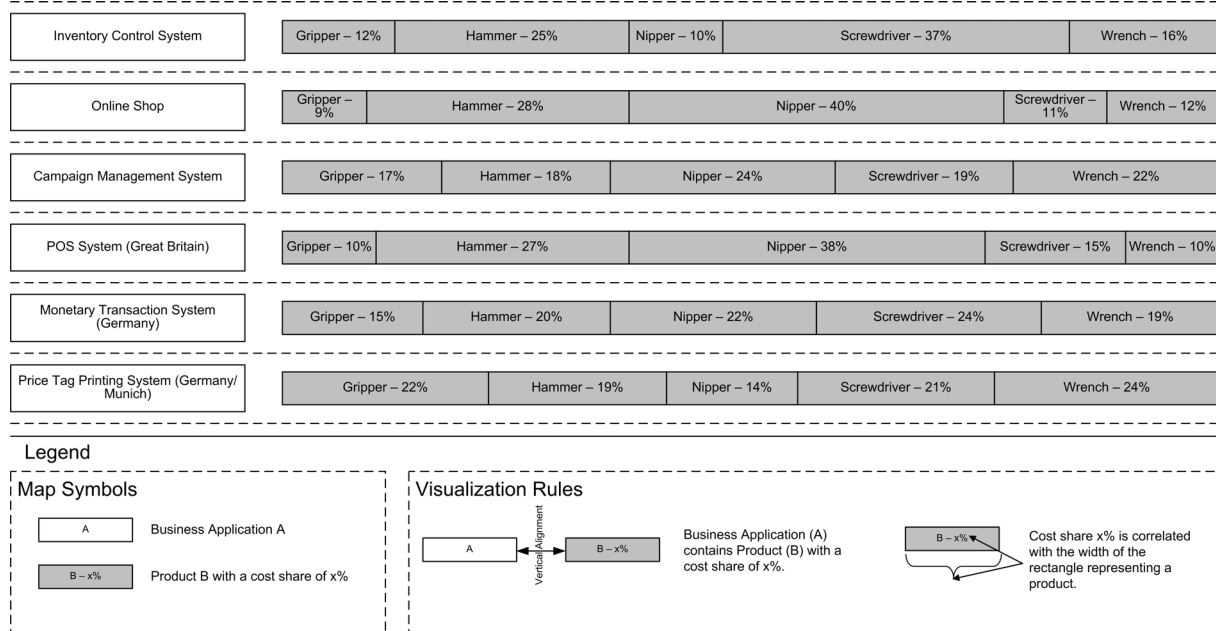


Figure 16: Exemplary view for *Cost Shares of Products in Business Applications*

This V-Pattern shows for each business application which products it provides and which share of the total costs of the business application a particular product makes up. The grey rectangles on the right side of a business application visualize the products and their width is correlated with the cost share of this product. The cost shares for all products of a business application sum up to 100%.

Consequence Section:

The benefit of this V-Pattern is its simplicity. It is equivalent to a plain pie chart with product cost shares for each business application. Furthermore, this kind of visualization can very intuitively be used to determine which products cause the major proportion of the total costs. Drawbacks of this V-Pattern are the time consuming processes of collecting the data and keeping the data up to date.

See Also:

The V-Pattern is based on information according to I-Pattern *Cost Shares of Products in Business Applications* (see page 44).

3.3.3 I-Patterns

The I-Patterns in the following subchapters also differ from the I-Patterns in the EAM Pattern Catalog [Bu08a]. As the V-Patterns in the previous section they show a more extensive structure. The EAM Pattern Catalog I-Pattern structure is only made up of a *V-Pattern overview* and a *solution section*. On the basis of [Er08] the structure of I-Patterns in this thesis was supplemented with a *variants section*, a *see also section* and a *consequence section*.

3.3.3.1 Business Application Lifecycles

I-Pattern Overview	
Name	Business Application Lifecycles
Id	I-26
Alias	
Summary	This I-Pattern describes the different lifecycles phases of a business application.
Version	1.1

Solution Section:

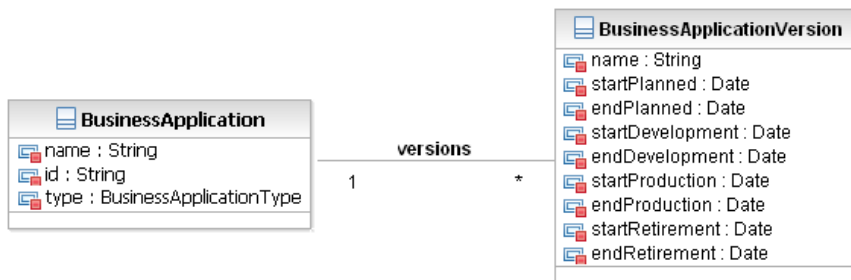


Figure 17: Information model fragment for *Business Application Lifecycles*. Source: [Bu08a]

This I-Pattern consists of two entities *BusinessApplication*, *BusinessApplicationVersion*, and one relationship *versions* and is visualized in Figure 17.

The entities and relationships can be defined as follows⁵:

- **BusinessApplication:** A business application is a software system, which is part of an information system of an organization. An information system is according to [Kr05] understood as a sociotechnical system, which is, besides the software system, made up of the infrastructure the software system is based on, and a social component, namely the employees or stakeholders concerned with it. Thereby, infrastructure and social component are not considered as belonging to the business application, while the characterization "business" restricts the term to applications that support at least one process of the respective organization. Thus, business application denotes here an actual deployment of a software.
- **BusinessApplicationVersion:** A specific version of a *BusinessApplication*, here meaning a specific release of this application. For versions, start and end dates of different lifecycle phases are recorded. The lifecycle phases *Planned*, *Development*, *Production* and *Retirement* are only exemplary and may be adjusted.

⁵ Definitions of classes which are already part of the EAM Pattern Catalog were copied and are not marked as quotations.

- **BusinessApplicationVersionVersionsBusinessApplication**: The association *versions* indicates which business application version is a version of which business application.

Variants:

An additional variant exists for this I-Pattern, in which the lifecyclephases are modelled explicitly and have an association to business application rather than being part of attributes of the class **BusinessApplicationVersion**. The information model fragment for this variant is shown in Figure 18.

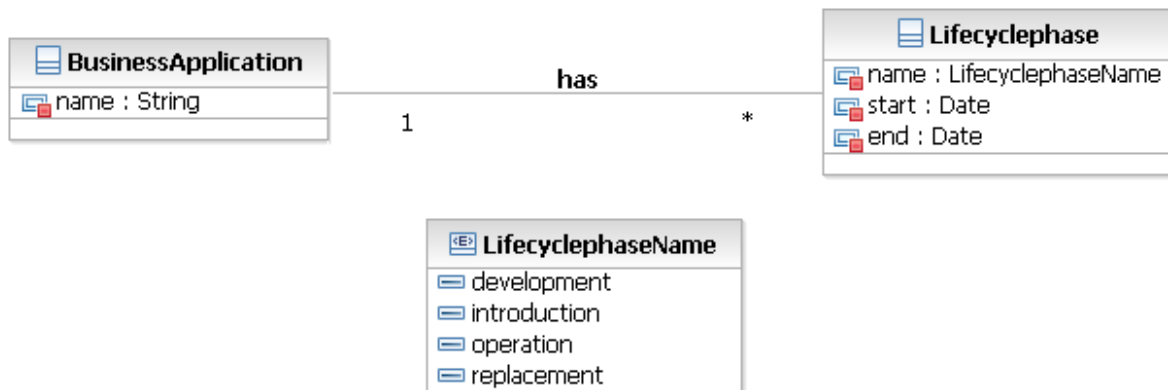


Figure 18: Information model fragment for variant of *Business Application Lifecycles*

The additional entities and relationships are defined as follows:

- **Lifecyclephase**: A lifecycle describes the process starting with the development and ending with the replacement of a business application. The lifecycle can be subdivided into several lifecycle phases.
- **LifecyclephaseName**: An Enumeration of different lifecycle phases. In this I-Pattern the four lifecyclephases *development*, *introduction*, *operation* and *replacement* are suggested.
- **BusinessApplicationHasLifecyclephase**: Each lifecycle phase with its specific start and end date belongs to exactly one **BusinessApplication**. A business application may have at most as many lifecycle phases as there are in **LifecyclephaseName**, in this case the maximum is four. Thereby, the name of the **Lifecyclephase** in the associations to a certain business application must be distinct.

Consequence Section:

A drawback of this I-Pattern is the amount of data that has to be collected to be able to reasonably analyze the data. The data collection is cumbersome as for every single lifecycle phase of a business application the start and end dates have to be recorded. However, on basis of the collected data very valuable analyses can be performed which contribute to the planning of the future application landscape.

See Also:

This I-Pattern is the basis for V-Pattern *Time Interval Map visualizing Lifecycles of Business Applications* (see page 19) and its variants.

3.3.3.2 Interfaces and Information Flows

I-Pattern Overview

Name	Interfaces and Information Flows
Id	I-63
Alias	
Summary	This I-Pattern describes the interfaces including information flows of business applications.
Version	1.1

Solution Section:

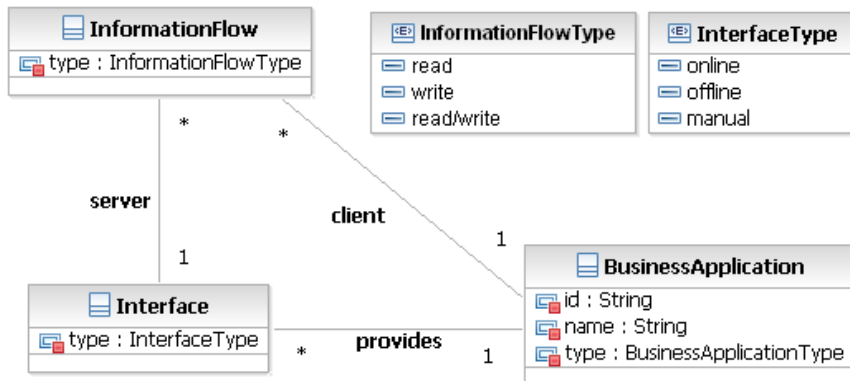


Figure 19: Information model fragment for *Interfaces and Information Flows*. Source: [Bu08a]

This I-Pattern visualized in Figure 19 is based on three entities and five relationships, which can be defined as follows:

- **BusinessApplication**: A business application is a software system, which is part of an information system of an organization. An information system is according to [Kr05] understood as a sociotechnical system, which is, besides the software system, made up of the infrastructure the software system is based on, and a social component, namely the employees or stakeholders concerned with it. Thereby, infrastructure and social component are not considered as belonging to the business application, while the characterization "business" restricts the term to applications that support at least one process of the respective organization. Thus, business application denotes here an actual deployment of a software.
- **InformationFlow**: Transfer of information between a **BusinessApplication** acting as a server, exposing functionality via an interface, and a client **BusinessApplication**, using this functionality. The type indicates the direction of the information transfer.
- **Interface**: An interface, via which a **BusinessApplication** can expose functionality for external usage.
- **BusinessApplicationProvidesInterface**: This association *provides* indicates that a business application exposes functionality via an interface.
- **BusinessApplicationClientInformationFlow**: This association *client* indicates that a business application acts as a client with respect to a information flow.
- **InterfaceServerInformationFlow**: This association *server* means that a business application via an interface acts as a server for an information flow.

- **InterfaceType**: Classifies different kinds of interfaces (e.g. online, offline, manual).
- **InformationFlowType**: Direction of a data flow between client and server: read means, that data is read from the server, write that data is transferred to the server, and read/write encompasses both.

Variants:

A possible variant is that interfaces may also be attached to application components instead of business applications. The information model fragment for this variant is shown in Figure 20.

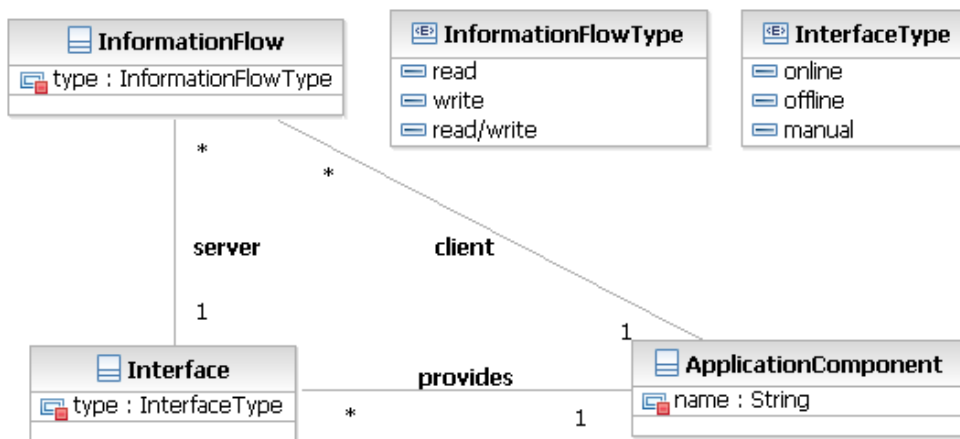


Figure 20: Information model fragment for variant of *Interfaces and Information Flows*

The additional entity is defined as follows:

- **ApplicationComponent**: Self-contained part of a system that encapsulates its content and exposes its functionality through a set of interfaces (definition from [Jo05]).

Consequence Section:

A drawback of this I-Pattern is the time-consuming collection of data in order to be able to visualize the information. Special attention has to be paid to the collection of data according to the variant of this I-Pattern. As application components usually form a hierarchy (see *Business Application Component Hierarchy*, see page 36) it is important to determine an adequate application component level on which the allocation of interfaces is to be done. It is also essential to do this allocation on the same level as otherwise the quality of analyses might be influenced.

See Also:

This I-Pattern is the basis for V-Pattern *Business Application Component Hierarchy including Interfaces* (see page 23) and its variants. It is also the basis for the following V-Patterns of the EAM Pattern Catalog [Bu08a]:

- *Cluster Map visualizing Business Object Flows between Business Applications* (see page 145)
- *Communication Table* (see page 147)
- *Information Flows* (see page 165)
- *Business Object Flows* (see page 186)

3.3.3.3 Business Application Component Hierarchy

I-Pattern Overview

Name	Business Application Component Hierarchy
Id	I-84
Alias	Breakdown of Business Applications
Summary	This I-Pattern describes business applications which consist of no or several application components. The application components may also be composed of application components.
Version	1.0

Solution Section:

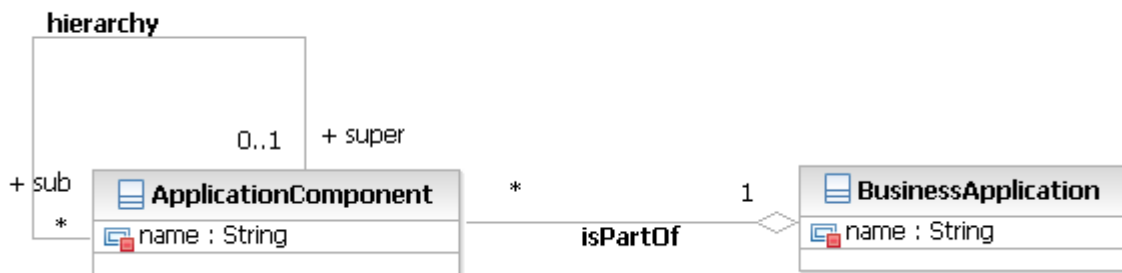


Figure 21: Information model fragment for *Business Application Component Hierarchy*

This I-Pattern, visualized in Figure 21 consists of two entities *BusinessApplication*, *ApplicationComponent*, and the relationship *isPartOf* and a hierarchy relationship of application component.

The entities and relationships can be defined as follows:

- **ApplicationComponent:** Self-contained part of a system that encapsulates its content and exposes its functionality through a set of interfaces (definition from [Jo05]).
- **BusinessApplication:** A business application is a software system, which is part of an information system of an organization. An information system is according to [Kr05] understood as a sociotechnical system, which is, besides the software system, made up of the infrastructure the software system is based on, and a social component, namely the employees or stakeholders concerned with it. Thereby, infrastructure and social component are not considered as belonging to the business application, while the characterization "business" restricts the term to applications that support at least one process of the respective organization. Thus, business application denotes here an actual deployment of a software.
- **ApplicationComponentIsPartOfBusinessApplication:** An *ApplicationComponent* is always part of a *BusinessApplication*. A *BusinessApplication* in turn aggregates no, one or several *ApplicationComponents*.
- **ApplicationComponentHierarchy:** *ApplicationComponents* form a hierarchy. An *ApplicationComponent* may have no or several sub application components and it may have no or one super application component.

Variants:

In this I-Pattern the allocation of an application component to a business application was chosen to be modelled as a distinct allocation. A variant of this I-Pattern is to suspend this

distinct allocation and allow for an application component to be part of several business applications. This however has an impact on the respective V-Pattern *Business Application Component Hierarchy* (see page 21) as the tree representation described in the variant section is no longer possible.

Consequence Section:

It is crucial for the usage of this I-Pattern to decide on the hierarchy levels, on which the application components are to be documented. If too many levels of application components are documented, the costs of doing so are in no relation to the benefits that can be drawn from analyzing the data. Furthermore, it is important to decompose the application components uniformly for all business applications in order to obtain comparable hierarchies and thus a feasible information basis for analyses.

See Also:

This I-Pattern is the basis for V-Patterns *Business Application Component Hierarchy* (see page 21) and *Business Application Component Hierarchy including Interfaces* (see page 23).

3.3.3.4 Business Process Support for Products

I-Pattern Overview

Name	Business Process Support for Products
Id	I-85
Alias	
Summary	This I-Pattern shows how information about and relationships between business processes, products and business applications can be stored.
Version	1.0

Solution Section:

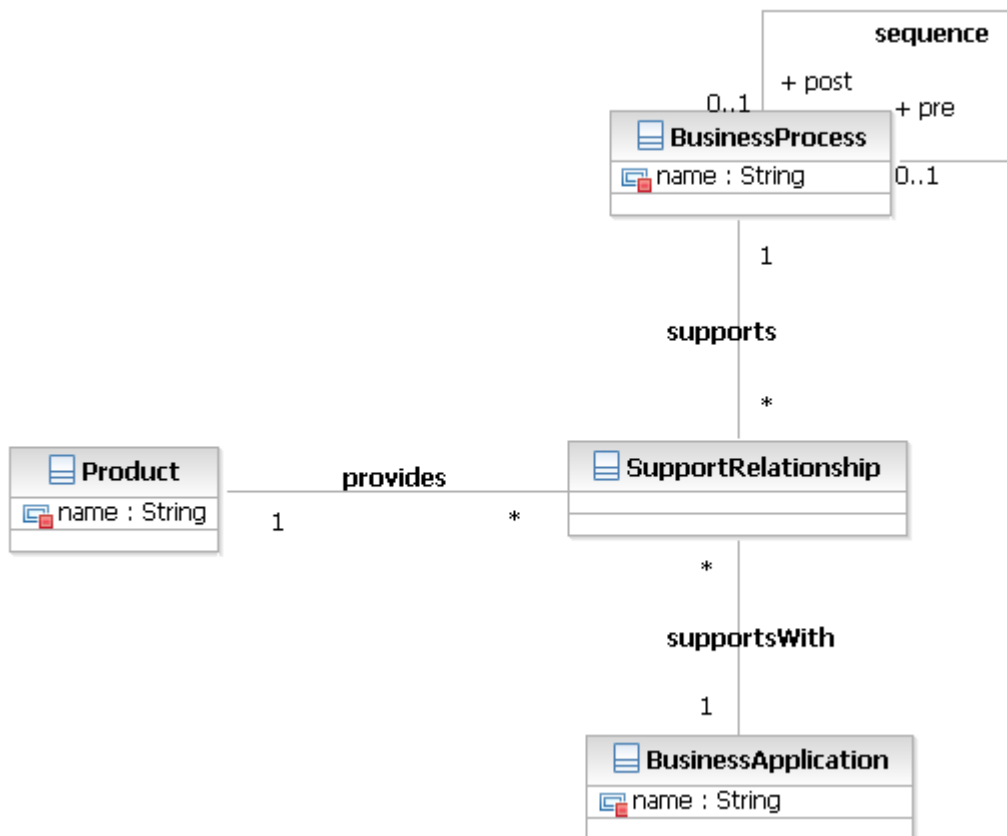


Figure 22: Information model fragment for *Business Process Support for Products*

The I-Pattern visualized in Figure 22 consists of the four entities *BusinessApplication*, *SupportRelationship*, *Product*, and *BusinessProcess*.

The entities can be defined as follows:

- **BusinessProcess:** According to [Kr05], defined as a sequence of logical individual functions with connections between them. [DFH03] states input and output factors and a defined process objective as important characteristics of a business process. The business process should not be identified with single process steps or individual functions, but with high-level processes at a level similar to the one used in value chains.
- **BusinessApplication:** A business application is a software system, which is part of an information system of an organization. An information system is according to [Kr05] understood as a sociotechnical system, which is, besides the software system, made up of the infrastructure the software system is based on, and a social component, namely

the employees or stakeholders concerned with it. Thereby, infrastructure and social component are not considered as belonging to the business application, while the characterization "business" restricts the term to applications that support at least one process of the respective organization. Thus, business application denotes here an actual deployment of a software.

- **Product:** A product is anything that can be offered to a person in order to satisfy a need or a desire. "Product" is a generic term for goods and services [KB05].
- **SupportRelationship:** Represents the support of a process and a specific product by a business application. Basically, it constitutes, together with its three associations, a ternary relationship between BusinessProcess, Product and BusinessApplication. This is necessary in order to be able to tell exactly which product is supported by which business application to support a given process.
- **BusinessProcessSequence:** A business process may have no or one predecessor and no or one successor.
- **SupportRelationshipSupportsBusinessProcess:** A support relationship supports exactly one business process, however, a business process may have no or several support relationships.
- **SupportRelationshipProvidesProduct:** A support relationship provides exactly one product, however, a product may have no or several support relationships.
- **SupportRelationshipSupportsWithBusinessApplication:** A support relationship is supported with one business application, however, a business application may support no or several support relationships.

Consequence Section:

A drawback is the effort of collecting data according to this I-Pattern. Organizations tend to have a large number of business processes, products and business applications, which have to be allocated to each other. Consequently the amount of data that has to be collected is considerably high and in particular the allocation of business processes, products and business application via support relationships is very time-consuming.

A benefit of this I-Pattern is that it facilitates the collection of data which describes the current situation of the application landscape and can therefore be used for planning the future application landscape. Data according to this pattern can be used to identify redundant functionality or missing IT support and thus is a very comfortable way to identify optimization potential.

See Also:

This I-Pattern is very similar to *Process Support* (see page 199 in [Bu08a]) and if those two I-Patterns are merged, they form a quarternary support relationship between BusinessApplication, BusinessProcess, OrganizationalUnit and Product. Furthermore it is the basis for V-Pattern *Process Support Map for Products* (see page 25).

3.3.3.5 Protection Requirements of Business Applications

I-Pattern Overview

Name	Protection Requirements of Business Applications
Id	I-86
Alias	
Summary	This I-Pattern shows how information about the protection requirement categories of business applications can be documented.
Version	1.0

Solution Section:



Figure 23: Information model fragment for *Protection Requirements of Business Applications*

This I-Pattern is visualized in Figure 23 and consists of the entity *BusinessApplication* and the enumeration *ProtectionRequirementCategory*.

The entity and the enumeration can be defined as follows:

- **BusinessApplication:** A business application is a software system, which is part of an information system of an organization. An information system is according to [Kr05] understood as a sociotechnical system, which is, besides the software system, made up of the infrastructure the software system is based on, and a social component, namely the employees or stakeholders concerned with it. Thereby, infrastructure and social component are not considered as belonging to the business application, while the characterization "business" restricts the term to applications that support at least one process of the respective organization. Thus, business application denotes here an actual deployment of a software.
- **ProtectionRequirementCategory:** For every business application, including the data held or used on it, the degree of protection is assessed in terms of confidentiality, integrity and availability. Each protection requirement is assigned to one of the three protection requirement categories. The *Bundesamt für Sicherheit in der Informationstechnik* suggests the protection requirement categories "basic to moderate", "high" and "very high" [BSI04].

Variants:

The protection requirement categories can be adjusted and thus new variants are created. The simplest adjustment is to change the naming of protection requirement categories. A further adjustment is to provide more detailed partitioning of protection requirement categories in order to facilitate more detailed analyses.

Consequence Section:

A drawback of this I-Pattern is that the preceding assessment of protection requirements is a very laborious process. The *Bundesamt für Sicherheit in der Informationstechnik* provides a process model for this purpose on its web page [BSI04]. The assessment of the protection requirement categories is laborious since for each business application a number of damage

scenarios with regard to confidentiality, integrity and availability has to be considered. Each damage scenario includes a number of questions which need to be answered. Moreover, in some cases it might be necessary to customize the damage scenarios, which entails even more work.

However, financial institutions are forced to assess their protection requirements due to regulations of the *Bundesanstalt für Finanzdienstleistungsaufsicht* (BaFin). The BaFin enjoins in [BAF07] a flexible and practical framework for the organization of risk management on financial institutions. This framework determines that IT systems and the according IT processes must ensure the integrity, availability, authenticity and confidentiality of data. In order to achieve this, the financial institutions should employ established standards, which are specified in the appendix as ISO 17799 of the *International Standards Organization* and the IT Grundschrift Manual of the *Bundesamt für Sicherheit in der Informationstechnik* [BSI04], in which the assessment of protection requirement is described.

See Also:

This I-Pattern is the basis for V-Pattern *Protection Requirements of Business Application*, (see page 27).

3.3.3.6 Business Applications' State of Health

I-Pattern Overview

Name	Business Applications' State of Health
Id	I-87
Alias	
Summary	This I-Pattern shows how information about the state of health of business applications can be stored.
Version	1.0

Solution Section:



Figure 24: Information model fragment for *Business Applications' State of Health*

The I-Pattern, visualized in Figure 24 consists of the entity *BusinessApplication* and the enumeration *StateOfHealthCategory*.

The entity and the enumeration can be defined as follows:

- **BusinessApplication:** A business application is a software system, which is part of an information system of an organization. An information system is according to [Kr05] understood as a sociotechnical system, which is, besides the software system, made up of the infrastructure the software system is based on, and a social component, namely the employees or stakeholders concerned with it. Thereby, infrastructure and social component are not considered as belonging to the business application, while the characterization "business" restricts the term to applications that support at least one process of the respective organization. Thus, business application denotes here an actual deployment of a software.
- **StateOfHealthCategory:** The state of health of a business application is a characteristic indicating how healthy this business application is. A poor state of health implies that the risk of a failure of this business application is high. The categories *poor*, *medium* and *good* are suggested.

Variants:

Variants of this I-Pattern can be created by adjusting the state of health categories. The simplest adjustment is to change their naming and if it is required to do more detailed analyses it might become necessary to provide a more detailed partitioning of state of health categories.

Consequence Section:

A drawback of this I-Pattern is that the assignment of stateOfHealthCategories might not be objective if for example the responsible IT architect for this business application makes the assignment. It is therefore recommended to assign the assessment of the stateOfHealthCategory to a neutral person. The drawback can be avoided by deducting the stateOfHealthCategory automatically. Thereby measurements depending for example on the number of bug fixes or breakdowns of a system, have to be created.

A benefit of is that data according to this pattern facilitates the identification of those business applications which need to be prioritized in projects in order to change or improve them.

Therefore, this I-Pattern contributes largely to the improvement of the overall state of health of the application landscape.

See Also:

This I-Pattern is the basis for V-Pattern *Business Applications' State of Health* (see page29).

3.3.3.7 Cost Shares of Products in Business Applications

I-Pattern Overview

Name	Cost Shares of Products in Business Applications
Id	I-88
Alias	
Summary	This I-Pattern shows how information about cost shares that products make up, can be documented.
Version	1.0

Solution Section:

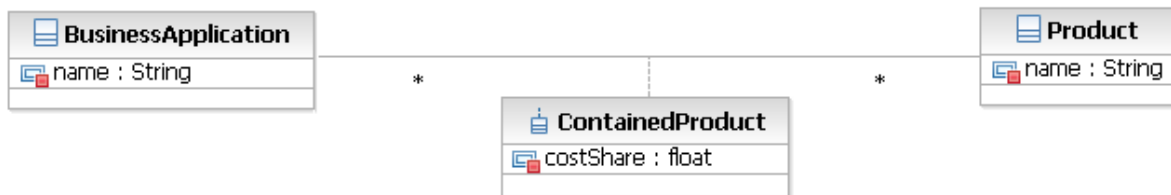


Figure 25: Information model fragment for *Cost Shares of Products in Business Applications*

This I-Pattern is visualized in Figure 25 and consists of the entities *BusinessApplication*, *Product* and the association class *ContainedProduct*.

The entities and the association class can be defined as follows:

- **BusinessApplication:** A business application is a software system, which is part of an information system of an organization. An information system is according to [Kr05] understood as a sociotechnical system, which is, besides the software system, made up of the infrastructure the software system is based on, and a social component, namely the employees or stakeholders concerned with it. Thereby, infrastructure and social component are not considered as belonging to the business application, while the characterization "business" restricts the term to applications that support at least one process of the respective organization. Thus, business application denotes here an actual deployment of a software.
- **Product:** A product is anything that can be offered to a person in order to satisfy a need or a desire. "Product" is a generic term for goods and services [KB05].
- **ContainedProduct:** Assigns a certain Product to the appropriate BusinessApplication and indicates what cost share the Product makes up in this BusinessApplication.

Consequence Section:

A drawback of this I-Pattern is the amount of information that has be collected and kept up to date. Furthermore, it is quite laborious to determine the costShares for the various products provided by a business application.

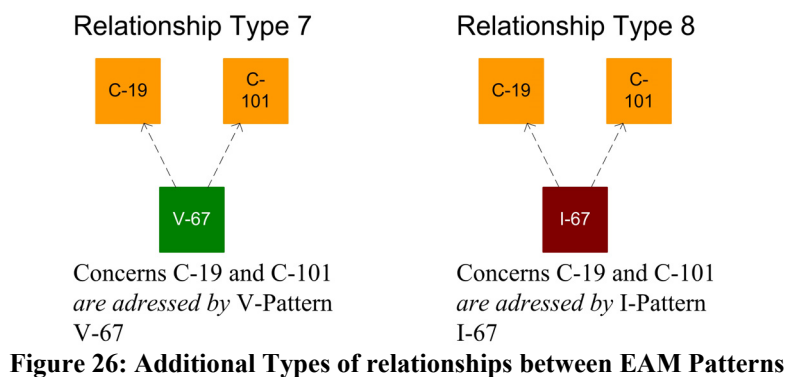
A benefit of this pattern is that it presents an easy way to document cost structures concerning products and business applications. It is therefore a good staring point to introduce costing information in an EA management information model.

See Also:

This I-Pattern is the basis for V-Pattern *Cost Shares of Products in Business Application*, (see page 31).

3.3.4 Relationships of additional EAM Pattern Catalog Elements

The overview graph in Figure 27 contains the additionally developed EAM Pattern Catalog elements, i.e. Concerns, V-Patterns and I-Patterns. Additional M-Patterns were not developed in the course of this thesis as they were not in the interest of the insurance company. For that reason the relationship types (see Figure 7) of the overview graph in 3.2 had to be adjusted. All relationship types concerning M-Patterns were excluded, which applies to relationship types 1, 2 and 6. Relationship types 3 to 5 are still contained in the overview graph of the additional EAM Pattern Catalog Elements. As the meta model of EAM Pattern Catalog elements and their relationships was subject to changes [Er08] and the EAM patterns developed in this thesis were to some extent adapted to the new meta model, it became necessary to also adjust the relationship types of the overview graph. The concept of Concerns was superseded by the slightly different concept of Problems with their respective forces. In addition a Problem is now not only addressed by M-Patterns, but also by V- and I-Patterns, which lead to the two additional relationship types shown in Figure 26.



Besides the additionally developed elements the overview graph of the additional EAM Pattern Catalog elements in Figure 27 contains elements of the EAM Pattern Catalog [Bu08a] and [Er08]. The elements taken from those two sources are marked with a * in order to make them distinguishable from the EAM Pattern Catalog elements that were developed in this thesis. V-24* (*Organizational Unit Business Application Cluster Map*, see page 23 in [Er08]) and I-24* (*Business Application and Organizational Unit Relationship*, see page 39 in [Er08]) address an additional Concern and are the only patterns taken from [Er08]. The elements added from the EAM Pattern Catalog [Bu08a] are solely V-Patterns which were identified while integrating the additionally developed V-Patterns.

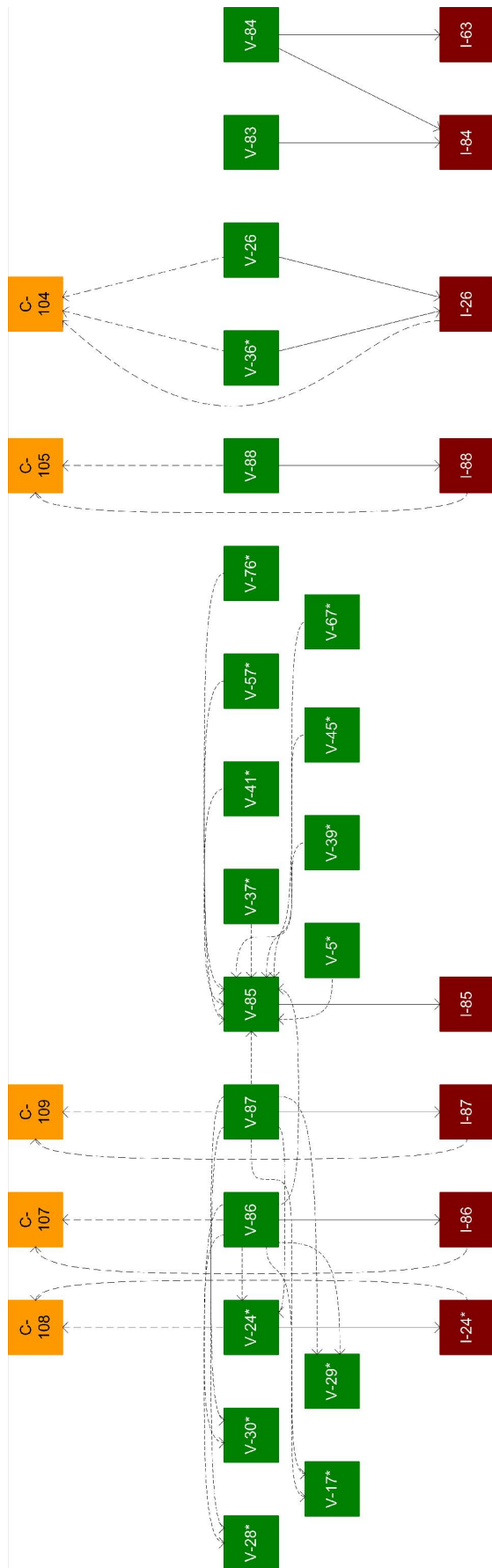


Figure 27: Overview graph containing additionally developed EAM Pattern Catalog elements and related V-Patterns of the EAM Pattern Catalog

3.4 Integrating I-Patterns

Integrating I-Patterns “strongly relies on the integrator’s skills in conceptual modelling” [Bu08b]. Steps taken for integrating I-Patterns are similar to integration steps in software engineering. [Bu07] sketches two integration approaches. The first approach can be applied if two information model patterns are to be integrated and one or more identical classes have been identified in both patterns. Those identical classes can then be used as a point of integration. However, it must be stated that identifying similar class names is not sufficient for identifying identical classes. Common sense implies that different concepts ought to be named clearly different. This is often not the case, particularly if the patterns to be integrated origin from various catalogs and therefore have different authors. Hence integrating information model patterns is not as easy as it may seem at first sight [Bu08b]. The second approach is to introduce „a new relationship between two classes from different patterns“ [Bu07]. This potentially involves inheritance as a class of the first pattern may inherit attributes of the respective class of the second pattern.

A different approach is suggested by [Kü04], who provides a number of metamodel integration patterns. An advantage of using such integration patterns is that one benefits from experiences from former metamodel integration projects [Kü03]. Applied to this case it means that integration patterns are used for integrating I-Patterns. In this thesis two of the metamodel integration patterns will be depicted, namely the *merge pattern* and the *extension pattern*. The *merge pattern* applies if two or more source metamodels are to be used concurrently. Parts from one or more source metamodels are taken and merged into a new concept which is part of the target metamodel. The *extension pattern* applies if new concepts are to be integrated into a source metamodel. Thus it is ensured that a metamodel evolves over time and can be adapted to new requirements.

In Figure 28 the integration process which can be subdivided into four steps is visualized. The four steps contain a number of tasks that were performed in order to obtain an information model at the end of each step. When looking at the integration process visualization the task *Integration of additional information model patterns* is eye-catching as it occurs three times. The reason for this is that additional I-Patterns were developed throughout the integration process due to Concerns emerging from discussions on the developed information models.

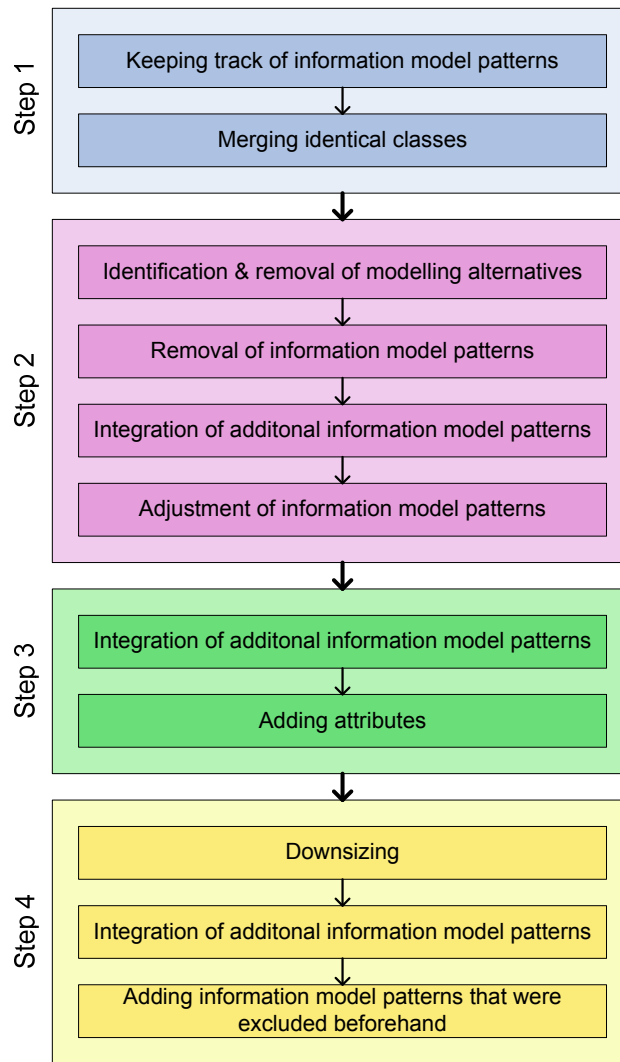


Figure 28: Integration Process

The I-Patterns to be integrated in the first step can be found in Table 5 in Section 3.2.3. All of those I-Patterns were integrated into one large information model without taking into account the modelling alternatives that were outlined in Section 3.2.3. As all those modelling alternatives are part of the information model, some redundancies occur in Figure 29. The following tasks were performed in this step:

- *Keeping track of information model patterns:*
In order to keep track of the I-Patterns used, each class was provided with a note indicating all I-Patterns, in which this class appears.
- *Merging identical classes:*
Identical Classes with identical names were merged. This means that those identically named classes were replaced by a single class and all associations of the identically named classes were turned into associations of the remaining class. Furthermore the attributes of identically named classes were summed up.



The information model in Figure 29 served as basis for the second step. The following tasks were performed:

- *Identification and removal of modelling alternatives:*
First the modelling alternatives outlined in 3.2.3 were identified. Afterwards the I-Pattern that fitted the requirements best was chosen and subsequently the other information model patterns were removed from the information model
- *Removal of information model patterns:*
I-Patterns were removed for several reasons. Firstly if they were considered to contain too detailed information and therefore cause extensive maintenance efforts in keeping the information basis up to date. Secondly an I-Pattern was excluded if it was too specific for a certain V-Pattern.
- *Integration of additional information model patterns:*
In this step additional I-Patterns from Section 3.3.3 were integrated into the information model.
- *Adjustment of information model patterns:*
The most time was spent on this task. It involves the adaption of associations, multiplicities and names. Moreover in some cases only parts of the I-Pattern were included in the information model.

Identification and removal of modelling alternatives

I-6 was omitted because *I-67* contains the same information plus the association class `DemandedSolution`.

I-Pattern *I-23* was omitted due to the more detailed *I-66*.

I-33 constitutes an alternative of *I-36* and would have been excluded anyway as it was decided to leave out versioning.

I-Pattern *I-75* was omitted as it basically contains the same information as *I-56*. Only some of the namings were borrowed in the modified *I-56*.

The I-Patterns *I-64*, *I-81* and *I-82* were skipped as they constitute alternatives to *I-63*, which is part of the information model.

Removal of information model patterns

I-26 was skipped as versioning was considered to cause a lot of maintenance effort and it would also make the information model more complex.

I-32 was omitted because it was felt that the timed process support would cause extensive maintenance efforts.

`ProjectProposal` were considered to be too time-consuming to be maintained and therefore *I-39* was excluded.

I-40 was omitted as `Milestones` and `MigrationActivity` are too detailed and would cause massive maintenance expenses.

Relationships between `BusinessObjects` were considered to be rather uninteresting and on account of this *I-46* and *I-47* were excluded.

I-51 was excluded because it was specifically designed for *V-51* and didn't fit into this information model.

I-52 was omitted as `BusinessRole` and `BusinessFunction` were rated as uninteresting and too detailed.

I-79 was excluded as call sequences were considered to be nonrelevant.

I-80 was omitted in the first place as changing interfaces were not rated as important.

Integration of additional information model patterns

I-Pattern *I-63* (see page 34) in its basic form and the additionally developed *I-84* (see page 36) are part of the second stage information model shown in Figure 30, but are highlighted in Figure 31. *I-63* was modified to the extent that `BusinessApplication` was replaced by `SupportProvider`.

Adjustment of information model patterns

I-30 and *I-36* were modified to the extent that `BusinessApplication` was replaced by `SupportProvider`, which is the super class of `BusinessApplication` and `Business Service`. A similar adjustment was made for *I-55*. In this case it was the class `Component` that was replaced by `SupportProvider`. The I-Patterns *I-24* and *I-25* of [Bu08a] were replaced by *I-24* (Business Application and Organizational Unit Relationship) of [Er08]. In order to distinguish this I-Pattern a * was added to the naming. *I-24** was also modified by replacing `Business Application` by `SupportProvider`.

From I-Pattern *I-48* only the transfers relationship between `BusinessObject` and `InformationFlow` was included in the second stage information model.

I-Pattern *I-66* is contained in a reduced form. `Connector` and `ConnectorRole` were removed and the association classes `Usage` and `AbstractUsage` were turned into normal uses associations.

I-56 was the I-Pattern that was most modified. First some renaming took place. `Device` became `HardwareSystem`, `SystemSoftware` became `InfrastructureSoftware` and the association class `SystemSoftwareDeployment` was renamed as `InfrastructureSoftwareDeployment`. Furthermore this association class was replaced by the class `InfrastructureSoftwareDeployment` and the associations `runsOn` and `deploymentOf`. Later on `InfrastructureSoftware` was merged with the class `Technology` of *I-66* as they constitute the same item. The single remaining class was then named `Technology`.

Included information model patterns

I-18 occurs twice in the second stage information model. Firstly the hierarchy association of `BusinessProcess` was included. Secondly the exposes relation between `BusinessService` and `BusinessApplication` became part of the information model.

I-Pattern *I-68* is already indirectly part of the information model as the modified *I-30* represents the process support by service.

I-41 and *I-67* are completely part of the information model.

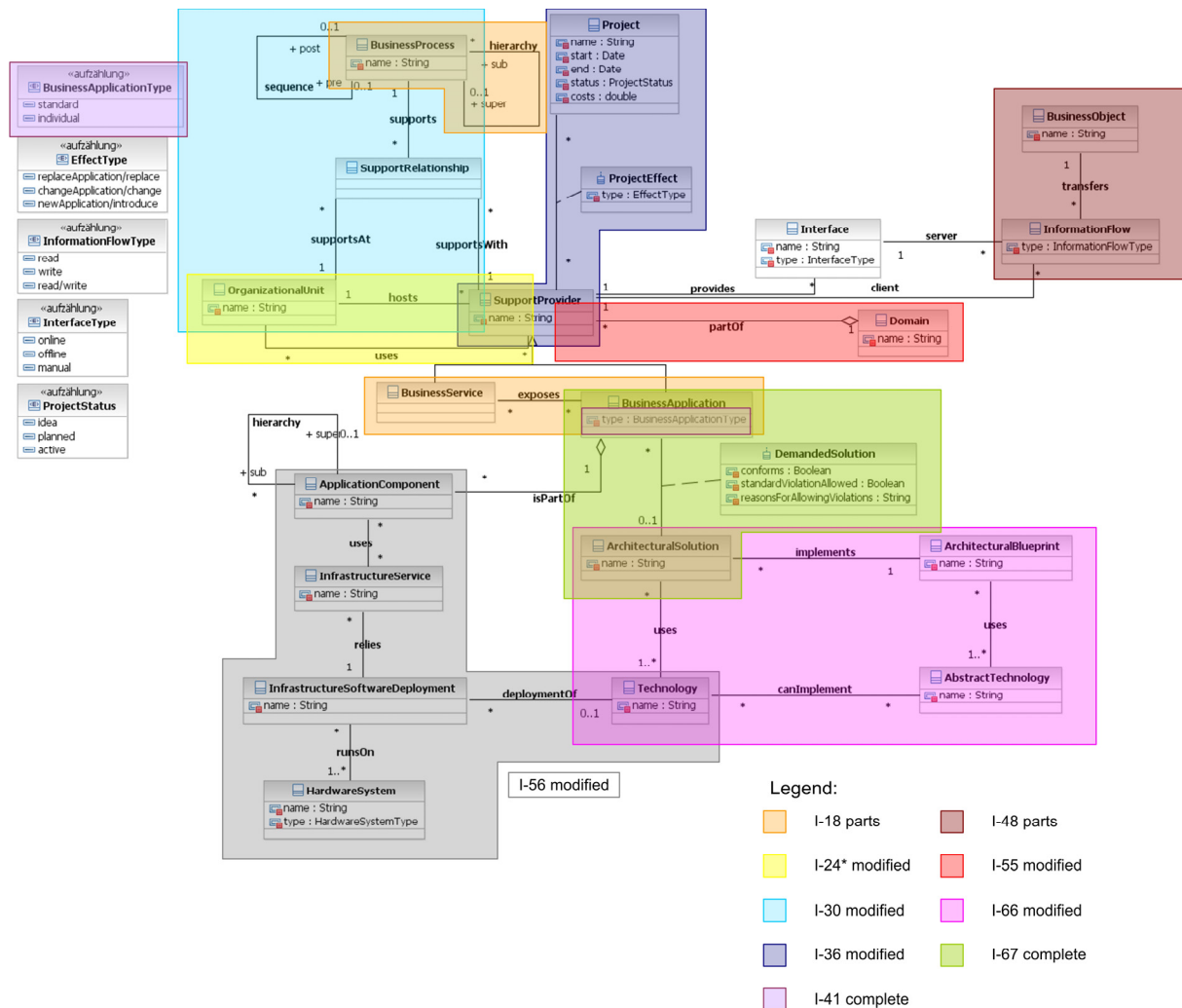


Figure 30: Second Step Information Model with highlighted EAM Pattern Catalog I-Patterns

The second stage information model of Figure 30 was the starting point for the following tasks:

- *Integration of additional information model patterns:*
In this step almost all remaining additional I-Patterns were integrated into the information model. As the I-Patterns in Section 3.3.3 are based on the final information model they may occur in a modified form.
- *Adding attributes:*
A number of attributes originating from the additional concerns (see 3.3.1) were added.

Integration of additional information model patterns

The additional I-Pattern **I-86** was integrated and for this reason the three attributes availability, integrity and confidentiality were appended to SupportProvider instead of BusinessApplication. Analogously **I-87** was integrated.

Concern C-108 demands a maintains-relationship between BusinessApplication and Organizational Unit. This relationship is contained in a variant of the I-Pattern **I-24*** of [Er08]. Thus the association maintains was added and again BusinessApplication was replaced by SupportProvider.

The variant of *I-26* and I-Pattern *I-88* were in the first place attached to SupportProvider in lieu of BusinessApplication.

Adding attributes

According to Concern C-102 the attributes operatingCosts and maintenanceCosts were added to SupportProvider and the attribute costs was added to Project. Concern C-103 was the cause of attributes numberOfUsers and numberOfMainUser of SupportProvider. As the attribute numberOfContracts applies to applications for portfolio of policies and not to BusinessServices, it was decided to attach this attribute to BusinessApplication. Due to Concern C-104 the attribute numberOfErrors was added to the attributes of Lifecyclephase. On account of Concern C-106 the attribute turnover was added to Product. The attribute shortName was added to BusinessApplication as in the current approach application has this attribute (see Section 2.2).

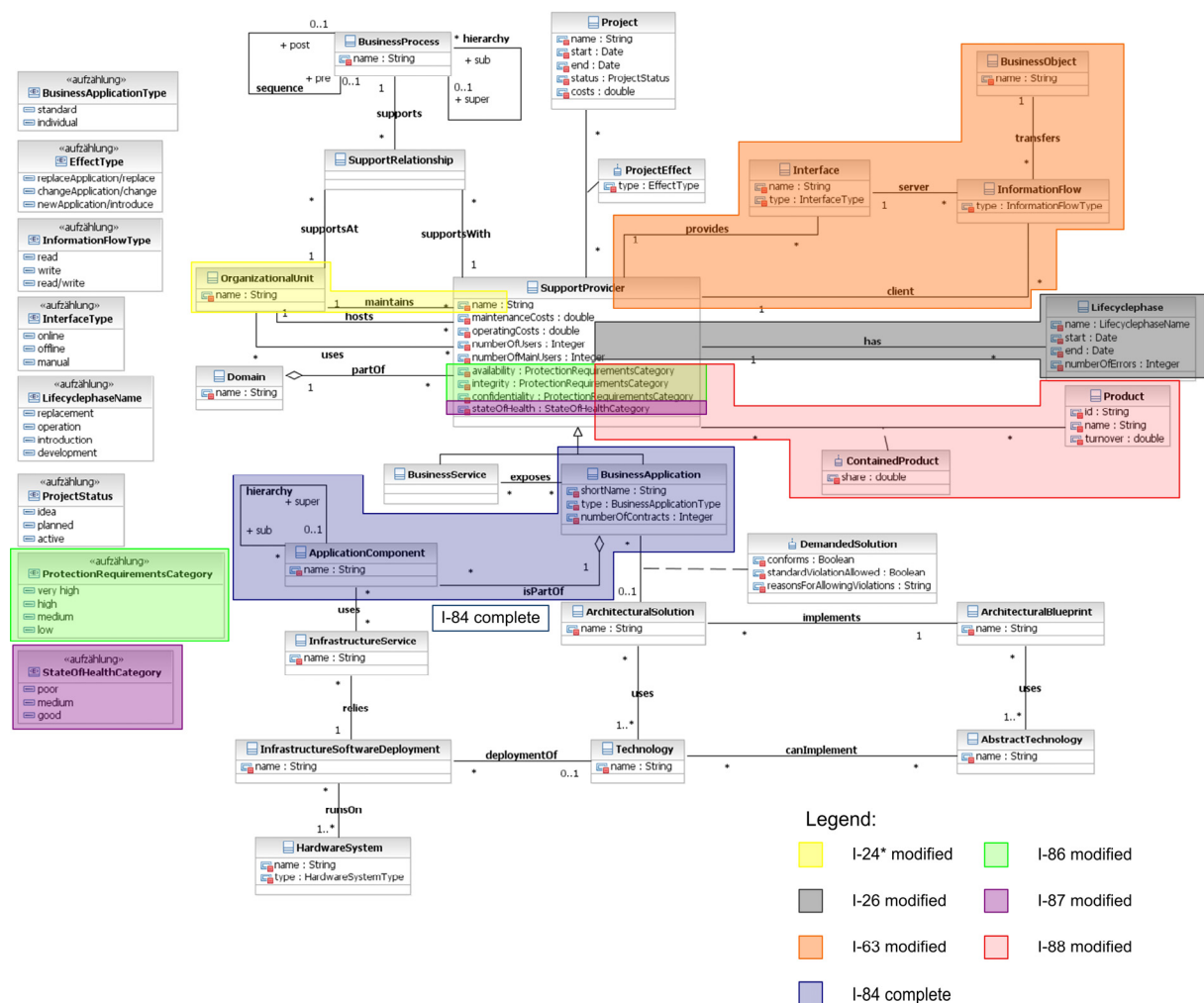


Figure 31: Third Step Information Model with highlighted additional I-Patterns

The fourth step is again based on the previous information model shown in Figure 31 and leads to the final information model shown in Figure 32. The following tasks were performed in this step:

- *Downsizing:*
A number of classes were subject to downsizing and were consequently removed from the information model. Furthermore an elaborate and detailed I-Pattern was substituted by its slimmer alternative.
- *Integration of additional information model patterns:*
In this step the last remaining additional I-Pattern was integrated into the information model. As the I-Patterns in Section 3.3.3 are based on the final information model they now occur in their described form.
- *Adding information model patterns that were excluded beforehand:*
After discussing the third stage information model it was requested to add parts of an I-Pattern, which was formerly excluded.

Downsizing

In the fourth step the information model was mainly subject to downsizing. The most striking difference is the absence of `SupportProvider` and `BusinessService`. The latter was removed on request of the insurance company and as a consequence `SupportProvider` became unnecessary and was also deleted. All associations going into `SupportProvider` were transferred to `BusinessApplication`. Furthermore all attributes of `SupportProvider` became attributes of `BusinessApplication`. As a result the additionally developed I-Patterns (see Section 3.3.3) are now contained in their original form. The downsizing also stroke `InfrastructureService`. This class was deleted and as a result `ApplicationComponent` is directly connected to `Infrastructure-SoftwareDeployment` by a `uses` association.

The modified I-Pattern *I-66* was replaced by the smaller *I-23* which only contains a `uses` relationship between `ArchitecturalSolution` and `Technology`.

Integration of additional information model patterns

Besides the downsizing attempts the additional I-Pattern *I-85* was integrated into the information model. It was merged with *I-30* and *I-88*. Consequently the `SupportRelationship` now constitutes a quarternary relationship between `BusinessApplication`, `OrganizationalUnit`, `BusinessProcess` and `Product`.

Adding information model patterns that were excluded beforehand

As requested a further addition was made and the `Interface`'s `replaceBy` association of I-Pattern *I-80* became part of the information model.

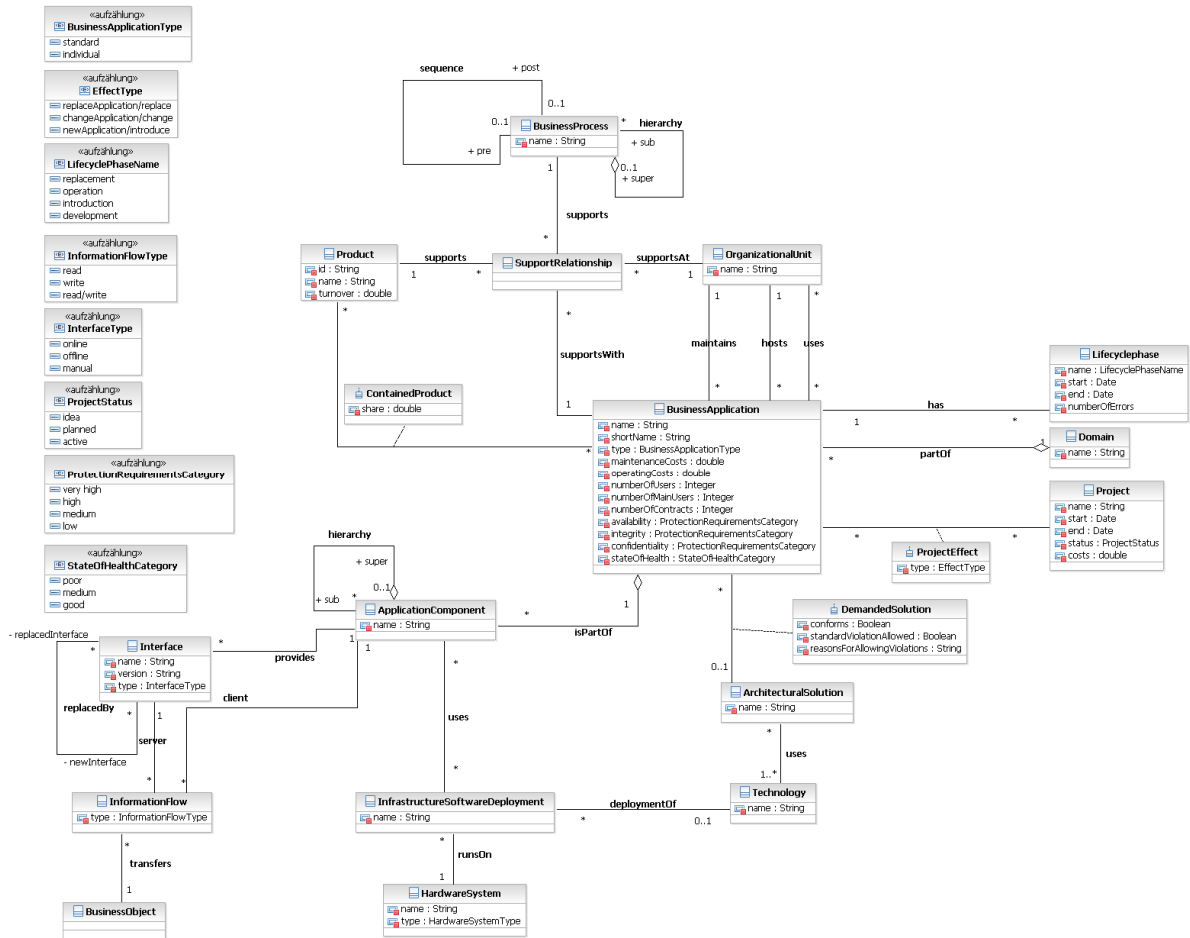


Figure 32: Fourth Step Information Model

The information model in Figure 32 is considerably smaller than the first stage information model in Figure 29. However, it is no longer possible to create all visualizations according to the V-Patterns of Table 4 as several I-Patterns were excluded in the integration process.

The final version was the starting point for the mappings with the information model of the current approach and the iteraplan information model, which are depicted in the following chapter.

4 Mapping Information Models

In this chapter the information model, which was developed in the preceding chapter is mapped to the information models of the current approach (see Section 2.2) and the Iteraplan⁶ open-source EAM Tool. At first a definition of mapping will be given in order to ensure a common understanding of this term.

Let $m(s)/f$ be the model m of the system s in the formalism f . A mapping can be defined as a transformation of a source model $m_1(s)/f_1$ into a target model $m_2(s)/f_2$, shortened $m_1/f_1 \rightarrow m_2/f_2$ [CS03].

There are two kinds of mappings:

- In *endogenous mapping* source and target model are expressed in the same formalism, thus $m_1(s)/f_1 \rightarrow m_2(s)/f_1$. An example for this kind of mapping is $m_1/UML \rightarrow m_2/UML$. In order to obtain the model m_2 a suite of modelling actions, for instance Create ModelElement or Delete ModelElement, has to be applied to the source model m_1 .
- In *exogenous mapping* source and target model are expressed in different formalisms ($m_1(s)/f_1 \rightarrow m_2(s)/f_2$). $m_1/UML \rightarrow m_2/C++$ constitutes an example for this type of mapping. Exogenous mapping requires to compare the metamodels of f_1 and f_2 , $mm_1=m(f_1)/f$ and $mm_2=m(f_2)/f$, in order to obtain transformation patterns from source pattern p_1/f_1 to target pattern p_2/f_2 .

In this case only the endogenous mapping is applicable as all information models are UML class diagrams. However, for the mapping of the information models a less formal and more intuitive approach was employed, which can be described by simply comparing classes and associations. Thereby, similar classes were marked with similar colours and the mappings of different concepts were described in natural language. The source model for the mappings is given in Figure 33.

⁶ <http://www.iteraplan.de/>

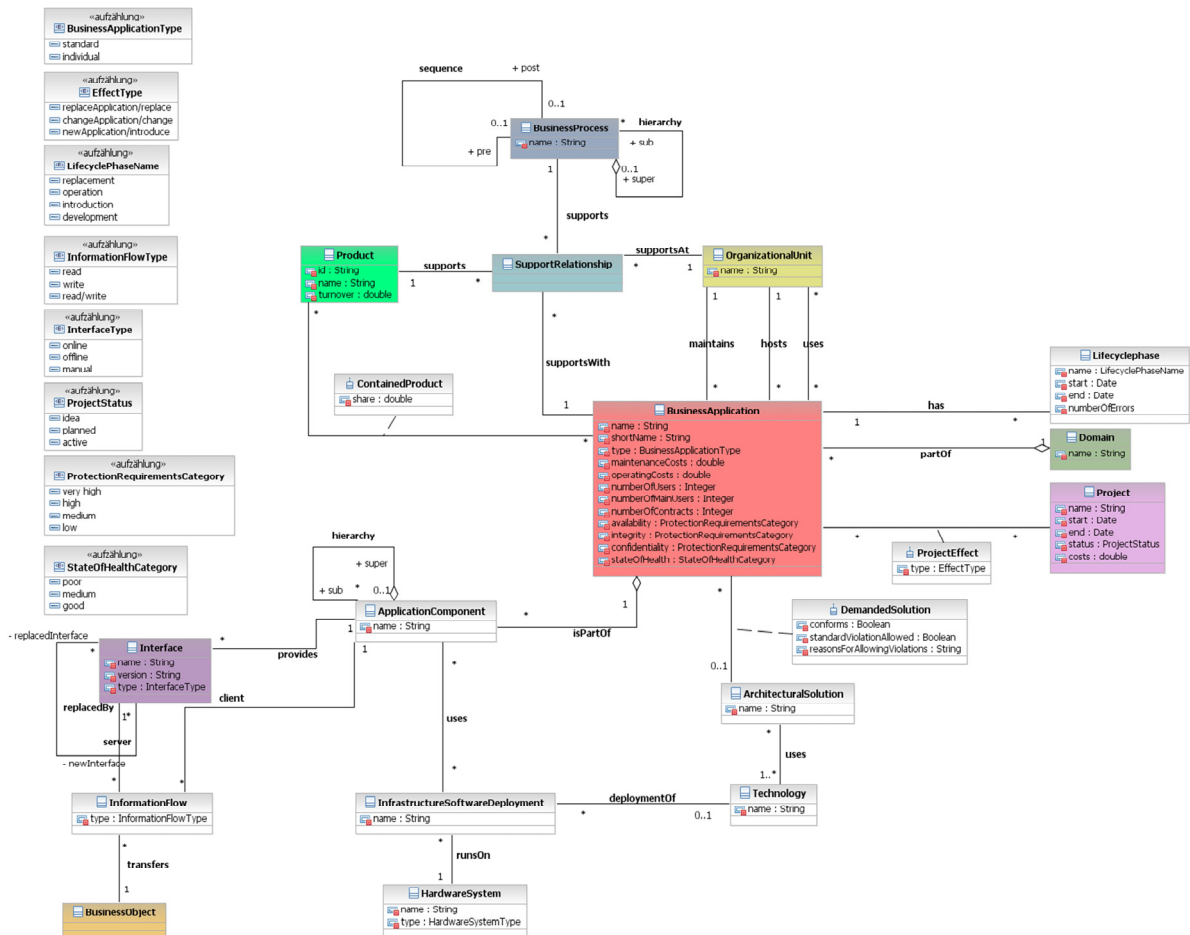


Figure 33: Source Model

4.1 Current Approach Information Model

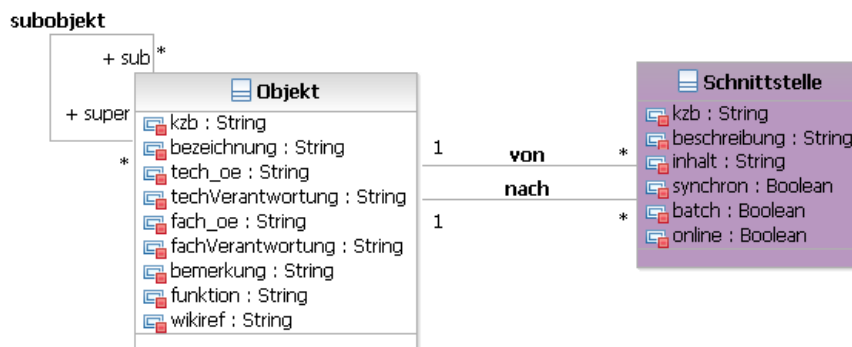


Figure 34: Target Model 1

As mentioned in Section 2.2 the current approach unites the concepts of business applications, application components, portfolios of policies and domains in a single class called Objekt. Objekt is thus the equivalent of BusinessApplication, ApplicationComponent and Domain. Schnittstelle matches with Interface in the source model whereas all other classes of the source model have no equivalent in the information model of the current approach.

The associations between Interface, InformationFlow and ApplicationComponent are completely different in the target model. The most striking

difference is the absence of the class `InformationFlow`. The client and provides associations match with the von and the server association is the equivalent of the nach association between `Objekt` and `Schnittstelle`.

The subobject relationship of `Objekt` in the target model fulfills the purposes of the following associations in the source model: hierarchy association of `ApplicationComponent`, `partOf` association between `Domain` and `BusinessApplication`, `isPartOf` association between `ApplicationComponent` and `BusinessApplication`. It must be taken into account that the cardinality of the subobject association is many to many whereas the source model suggests that a business application is part of only one domain and that an application component can only be part of one business application. In principle it is furthermore possible to have multiple parents for one `ApplicationComponent`, which is not allowed in the source model.

4.2 Iteraplan Information Model

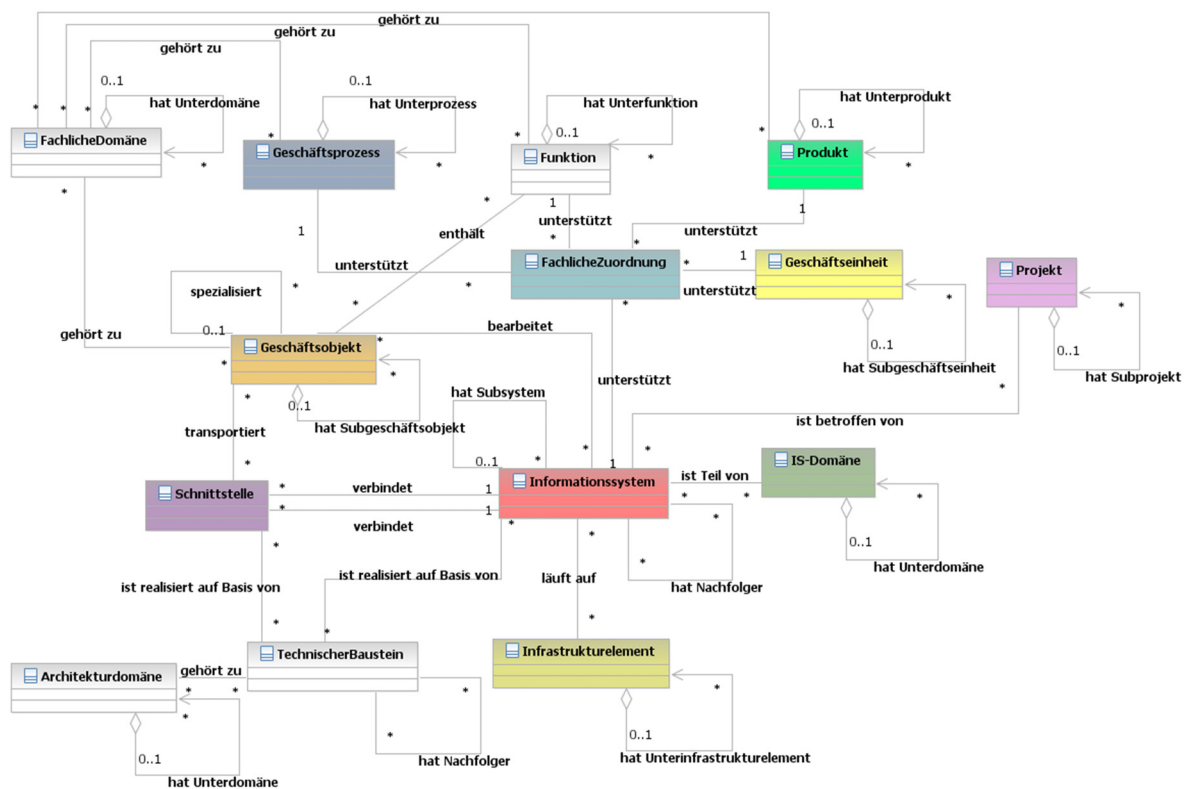


Figure 35: Target Model 2

As the coloured classes in Figure 35 imply, several matching classes were identified and are listed in Table 7. Classes of the source model that have no equivalent in the target model are: `ContainedProduct`, `Lifecyclephase`, `ProjectEffect`, `Demanded-Solution`, `ArchitecturalSolution`, `Technology`, `Infrastructure-SoftwareDeployment`, `ApplicationComponent`. Classes of target model 2, which have no matching class in the source model are: `FachlicheDomäne`, `Funktion`, `Architekturdomäne`, `TechnischerBaustein`.

Source Model	Target Model 2
BusinessApplication	Informationssystem
Domain	IS-Domäne
HardwareSystem	Infrastrukturelement
Interface	Schnittstelle
BusinessObject	Geschäftsobjekt
BusinessProcess	Geschäftsprozess
SupportRelationship	FachlicheZuordnung
Product	Produkt
OrganizationalUnit	Geschäftseinheit
Project	Projekt

Table 7: Matching Classes of Source Model and Target Model 2

Besides the differing classes there are also many associations which have no equivalent in the other model. In this thesis only two main differences in associations will be described. Firstly, the `isPartOf` relationship between `ApplicationComponent` and `BusinessApplication` and `ApplicationComponent`'s hierarchy relationship can be mapped on the `hatSubsystem` association of `Informationssystem` in the target model. The target model's solution makes no clear distinction between application components and business application and consequently the application components inevitably have the same attributes as the business applications. Secondly the `istTeilVon` association between `Informationssystem` and `IS-Domäne` has a different cardinality than the `partOf` association between `BusinessApplication` and `Domain`. In the target model the cardinality is many to many whereas the source model defines domains to be disjunctive sets.

5 Evaluation of the EAM Pattern Catalog Approach

The starting point for developing a new EA management approach for the insurance company was, in accordance with the usage scenario *establishing an organization-specific EA management through EAM Pattern Integration*, the compilation of all the pain points of the insurance company, the so called Concerns. The third Chapter of the EAM Pattern Catalog [Bu08a] contains a list of all Concerns which served as a guideline for the meeting in which the Concerns were rated as relevant or not. This list saved a lot of time with regard to the preparation of the meeting and furthermore it was a good basis for discussions, which led to the discovery of additional Concerns.

Another benefit of the EAM Pattern Catalog is that the sole usage of this document was sufficient for the development of an EA management approach and no further assistance was required. This is partly due to the detailed introduction chapter of the EAM Pattern Catalog, in which the pattern-based approach is described extensively. The definitions provided for the different EAM Pattern types and the characterizations of their dependencies facilitate understanding the overall structure of the EAM Pattern Catalog. Moreover, the usage scenarios described in Chapter 2 of the EAM Pattern Catalog contributed largely to the comprehension of how the EAM Patterns are applied in order to obtain a new or enhanced EA management approach.

In this thesis over 60% of all EAM Patterns of the catalog were selected. Despite this large number of EAM Patterns no difficulties concerning the scope of the selected EAM Patterns occurred. The reason for this is that the EAM Pattern overview graph constitutes a powerful tool which gives an overview of all EAM Patterns and enables the identification of all dependencies of a certain EAM Pattern at a glance.

A further benefit of the EAM Pattern Catalog is its consistent and meaningful terminology. A principle is that different concepts ought to be named clearly different in order to avoid misconceptions. Especially when it comes to integrating I-Patterns it is essential to obviate any misinterpreting and therefore for every class contained in an information model fragment a detailed definition is provided.

When integrating I-Patterns by simply identifying and merging similar named classes, as described in the first step of the integration process (see Section 3.4), an almost cohesive information model was created. The only exception was I-55, which can be easily integrated by replacing the generic class `Component` by `BusinessApplication`. I-Pattern I-55 only constitutes an exception due to the fact that it is kept very abstract in order to facilitate its usage in several contexts. The resulting cohesive information model indicates that the I-Patterns of the EAM Pattern Catalog are well matched.

Practice orientation is a further advantage of the EAM Pattern Catalog. This is proved by calculating the EAM Pattern Catalog share in the entirety of selected and additionally developed EAM Patterns. For Concerns this proportion is 84% and thus 84% of the pain points of the insurance company coincide with Concerns of the EAM Pattern Catalog. The shares calculated for V-Patterns and I-Patterns are 88% and 87% respectively.

A drawback of the EAM Pattern Catalog Approach lies in the strictly linear structure of the EAM Pattern Catalog, i.e. Concerns are addressed by M-Patterns, which use V-Patterns. V-Patterns in turn visualize the information of I-Patterns. In this thesis M-Patterns were neglected completely and consequently problems arose when the additionally developed V- and I-Patterns were to be related to Concerns. This problem however was solved by the new relationships between Problem, M-Patterns, V-Patterns and I-Patterns suggested in [Er08].

A further disadvantage was revealed while integrating I-Patterns. It was detected that many I-Patterns constitute modelling alternatives of other I-Patterns. In the course of the thesis those modelling alternatives were identified and are outlined in Section 3.2.3. Applying the

adjusted EAM Pattern structure of [Er08] this problem might be solved by determining one modelling alternative as original of an I-Pattern and describing the others in its variant section.

Another drawback is that Section 2.4.3 on integrating I-Patterns (see page 28 in [Bu08a]) is too compact considering the complexity of the integration task. Therefore it would be beneficial to complete Section 2.4.3 in [Bu08a] with a description of the different ways of integrating information model fragments and supplement each way with an example. Furthermore, it would be useful to have integration guidelines or even an integration process in order to ensure a structured proceeding.

Due to the pros mentioned above, first results can be generated quickly and can be iteratively refined later on. Furthermore, the development of an EA management approach can be carried out by a single person in a reasonable period of time. All in all the EAM Pattern Catalog provides a convenient way of developing or enhancing an EA management approach.

6 Recapitulation and Prospects of the Thesis

This last chapter of the thesis provides a concise summary of the findings and concludes with an outlook on future research possibilities.

6.1 Recapitulation

The objective of this thesis was the application and evaluation of the EAM Pattern in practice. Initially the former and the current EA management approach of the insurance company were investigated in Sections 2.1 and 2.2. In particular the former approach conforms to some patterns of the EAM Pattern Catalog. At the beginning of Chapter 3 the existing concerns were identified. Thereby, it was detected that some EA management topics were of no or little interest whereas others like, for instance, infrastructure management was of major interest. In Section 3.2 the respective EAM Patterns were deducted from the Concerns. The following Section 3.3 deals with the extensions of the EAM Pattern Catalog. In the first subsection the additional concerns are listed for which in Sections 3.3.2 and 3.3.3 respective V- and I-Patterns were developed. Subsequently those additional EAM Patterns were related to the additional concerns and also to each other. Thereby, the adjusted structure of the additional EAM Patterns led to adjustments in the relationship types of the overview graph. The subsequent Section 3.4 describes an exemplary integration process for I-Patterns and resulted in the information model, which served as an input for the mappings of Chapter 4. The first subsection describes the mapping of the developed information model to the information model of the current approach. This mapping revealed a minor match due the tiny size of the current approach information model. The second subsection is concerned with the mapping to the information model of the iteraplan EA management tool, which has several concepts in common with the developed information model. Chapter 5 evaluates the EAM Pattern approach and the concluding Chapter 6 identifies future research potential.

6.2 Prospects

The additional V- and I-Patterns developed in Sections 3.3.2 and 3.3.3 were so far only assessed by the advisor of this thesis. For reasons of quality assurance it might be reasonable to present the additionally developed EAM Patterns to the EAM pattern community and have those patterns revised by several experts and practitioners.

Furthermore, a survey might be conducted which is analogous to the survey that was carried out during the compilation of the EAM Pattern Catalog. Thus, the relevance of the additionally developed EAM Patterns would be assessed by a number of experienced practitioners and might serve as a basis of decision which of those EAM Patterns might become part of a later version of the EAM Pattern Catalog.

The additional usage scenario of the EAM Pattern Catalog that was discovered in the course of this thesis might be the subject of further research. An empirical study with the aim of identifying the relevance of the three usage scenarios given by the EAM Pattern Catalog and the usage scenario applied in this thesis, might be conducted.

Moreover, the I-Patterns which were identified as modelling alternatives (see Section 3.2.3) could be united in a single I-Pattern due to the adjusted EAM Pattern structure presented in [Er08]. Thereby, one of the modelling alternatives has to be determined as original and the others are specified in the variants section.

A further prospect is the development of a generic I-Pattern integration process, which can be customized according to the requirements of a specific project.

As Section 2.4.3 of the EAM Pattern Catalog (see page 28 in [Bu08a]), delineating the integration of I-Patterns, was evaluated as too compact, a prospect might be to enhance this section with descriptions of the different ways of integrating information model fragments.

Sticking to the pattern-based approach of the EAM Pattern Catalog, those descriptions could be provided in the form of patterns which might resemble the metamodel integration patterns in [Kü04].

A. List of Abbreviations

BaFin	Bundesanstalt für Finanzdienstleistungsaufsicht
BSI	Bundesamt für Sicherheit in der Informationstechnik
EAM	Enterprise Architecture Management
IT	Information Technology

B. Bibliography

- [BAF07] Bundesanstalt für Finanzdienstleistungsaufsicht: Rundschreiben 5/2007 (BA) – Mindestanforderungen an das Risikomanagement - MaRisk. In: http://www.bafin.de/cln_109/nn_722754/SharedDocs/Veroeffentlichungen/DE/Service/Rundschreiben/2007/rs_0705_ba.html#Start, accessed 13.11.08.
- [Bu07] Buckl, S.; Ernst, A.; Lankes, J.; Matthes, F.; Schweda, C.; Wittenburg, A.: A Pattern based Approach for constructing Enterprise Architecture Management Information Models. In: 8. Internationale Tagung Wirtschaftsinformatik, Karlsruhe, 2007.
- [Bu08a] Buckl, S.; Ernst, A.; Lankes, J.; Matthes, F.: Enterprise Architecture Management Pattern Catalog (Version 1.0, February 2008). Technical Report TB 0801, Chair for Informatics 19, Technische Universität München, 2008.
- [Bu08b] Buckl, S.; Ernst, A.; Lankes, J.; Matthes, F.; Schweda, C. M.: Enterprise Architecture Management Patterns - Exemplifying the Approach. In: The 12th IEEE International EDOC Conference (EDOC 2008), München, 2008.
- [BSI04] Bundesamt für Sicherheit in der Informationstechnik: IT Grundschutz Manual. In: <http://www.bsi.bund.de/english/gshb/manual/index.htm>, accessed 15.08.08.
- [CS03] Caplat, G.; Sourrouille, J. L.: Considerations about Model Mapping. Workshop in Software Model Engineering (WiSME 2003), San Francisco, 2003.
- [DFH03] Disterer, G.; Fels, F.; Hausotter A.: Taschenbuch der Wirtschaftsinformatik. Carl Hanser Verlag, München, 2003.
- [Er06] Ernst, A.; Lankes, J.; Schweda, C.; Wittenburg, A.: Tool Support for Enterprise Architecture Management - Strengths and Weaknesses. In: The 10th IEEE International EDOC Conference (EDOC 2006), Hong Kong, 2006.
- [Er08] Ernst, A.: Enterprise Architecture Management Patterns. Pattern Languages of Programs Conference 2008 (PLoP08), Nashville, 2008. (in publication)
- [FAW07] Fischer, R.; Aier, S.; Winter, R. (2007): A Federated Approach to Enterprise Architecture Model Maintenance. In: 2nd International Workshop on Enterprise Modeling and Information Systems Architectures - Concepts and Applications, St. Goar/Rhine, Germany, 2007.
- [Jo05] Jonkers, H.; Groenewegen, L.; Bonsangue, M.; van Buuren, R.: A language for Enterprise Modelling. In: Lankhorst, M.: Enterprise Architecture at Work. Springer, Berlin, Heidelberg, New York, 2005.

- [KB05] Kotler, P.; Bliemel, F.: Marketing-Management: Analyse, Planung und Verwirklichung. 10th edition, Pearson Studium, München 2005.
- [Kr05] Krcmar, H.: Informationsmanagement. 4th edition, Springer, Berlin 2005.
- [Kü03] Kühn, H.; Bayer, F.; Junginger, S.; Karagiannis, D.: Enterprise Model Integration. In: Bauknecht, K.; Tjoa, A M.; Quirchmayr, G. (Eds.): Proceedings of the 4th International Conference EC-Web 2003 . Dexa 2003, Prague, Czech Republic, 2003.
- [Kü04] Kühn, H.: Methodenintegration im Business Engineering. Dissertation, Fakultät für Wirtschaftswissenschaften und Informatik, Universität Wien, 2004.
- [PS05] Pereira, C. M.; Sousa, P.: Enterprise Architecture: Business and IT Alignment. In: Proceedings of the 2005 ACM symposium on Applied computing, 2005.
- [Wi07] Wittenburg, A.: Softwarekartographie – Modelle und Methoden zur systematischen Visualisierung von Anwendungslandschaften. Dissertation, Fakultät für Informatik, Technische Universität München, 2007.