Patterns in Enterprise Architecture Management

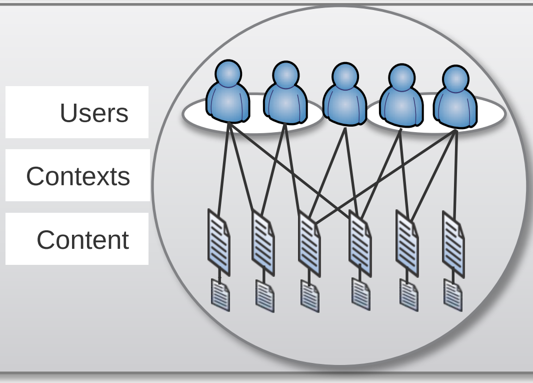
SI-SE Fachtagung, Zürich, 30.1.2009

Prof. Florian Matthes, Alexander Ernst
Software Engineering for Business Information Systems (sebis)
wwwmatthes.in.tum.de

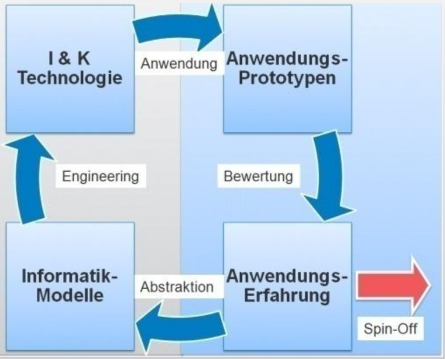
Areas of research

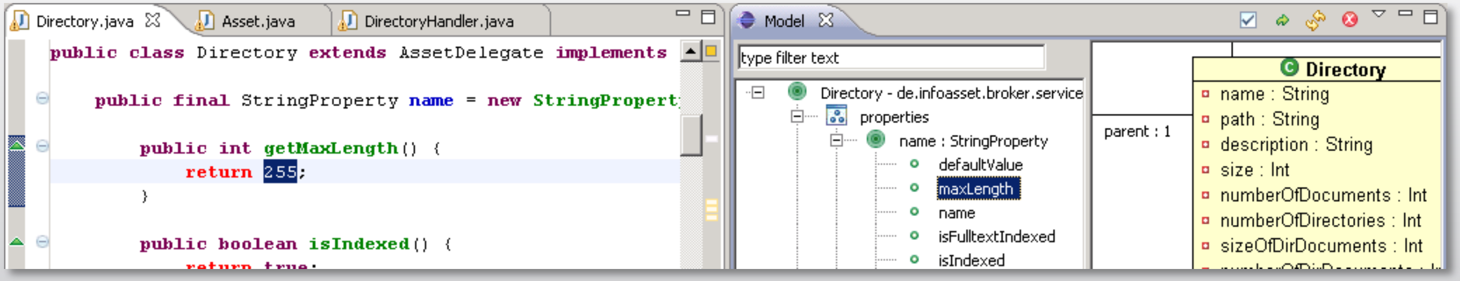
Managed Evolution of Socio-Technical Systems of Systems (EAM)

Social Software



Transfer Projects





```
public class Directory extends AssetDelegate implements

    public final StringProperty name = new StringPropert

        public int getMaxLength() {
            return 255;
        }

        public boolean isIndexed() {
            return true;
```

```
Directory - de.infoasset.broker.service
    properties
        name : StringProperty
            defaultValue
            maxLength
            name
            isFulltextIndexed
            isIndexed
```

parent : 1

Directory
- name : String
- path : String
- description : String
- size : Int
- numberOfDocuments : Int
- numberOfDirectories : Int
- sizeOfDirDocuments : Int

Domain-Driven Software Architectures

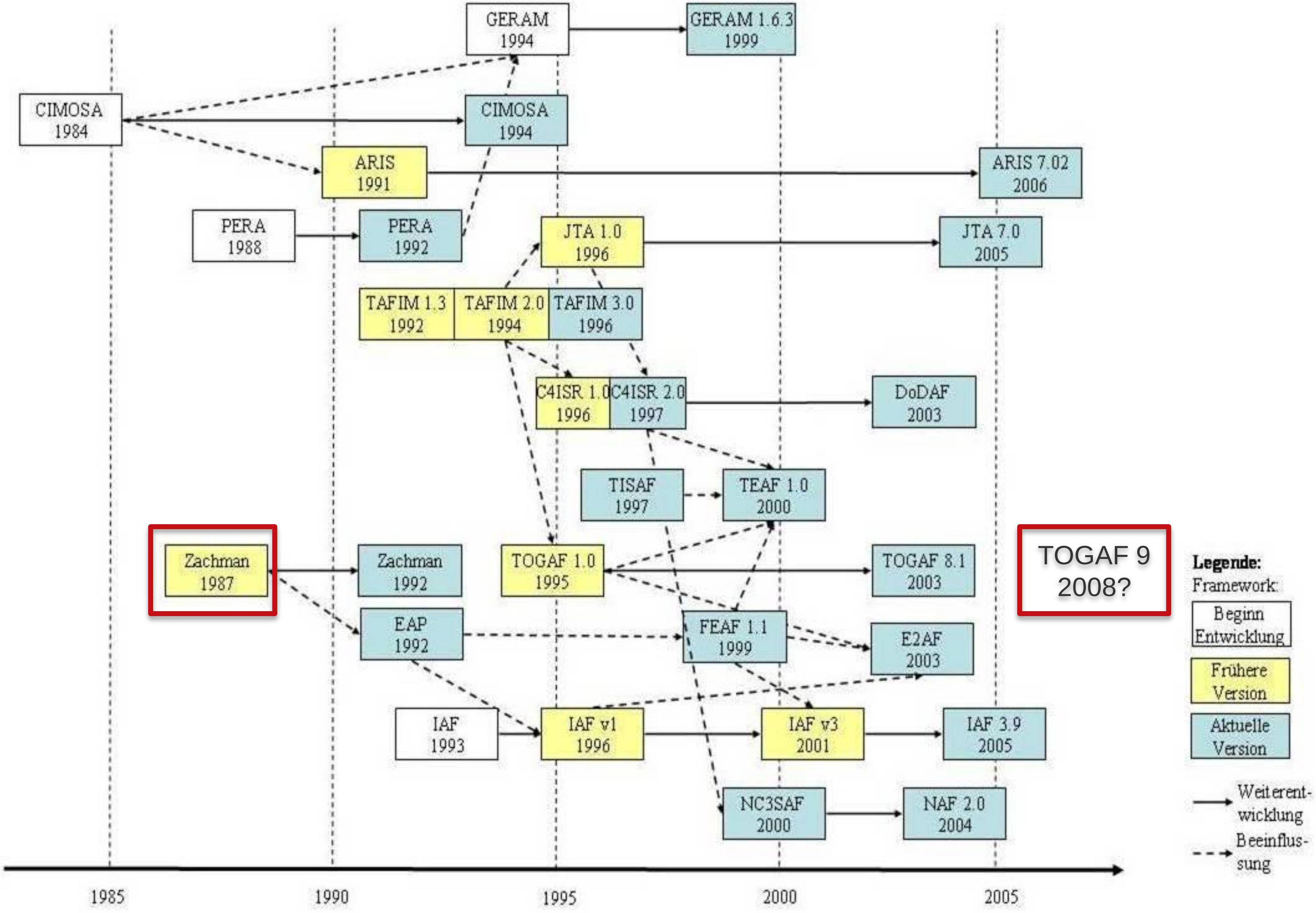
Outline

1. Introducing EAM in an enterprise is a challenge
   - models, viewpoints, management processes

2. The EAM pattern catalog 1.0
   - rationale, contents, contributors

3. Towards an EAM pattern community

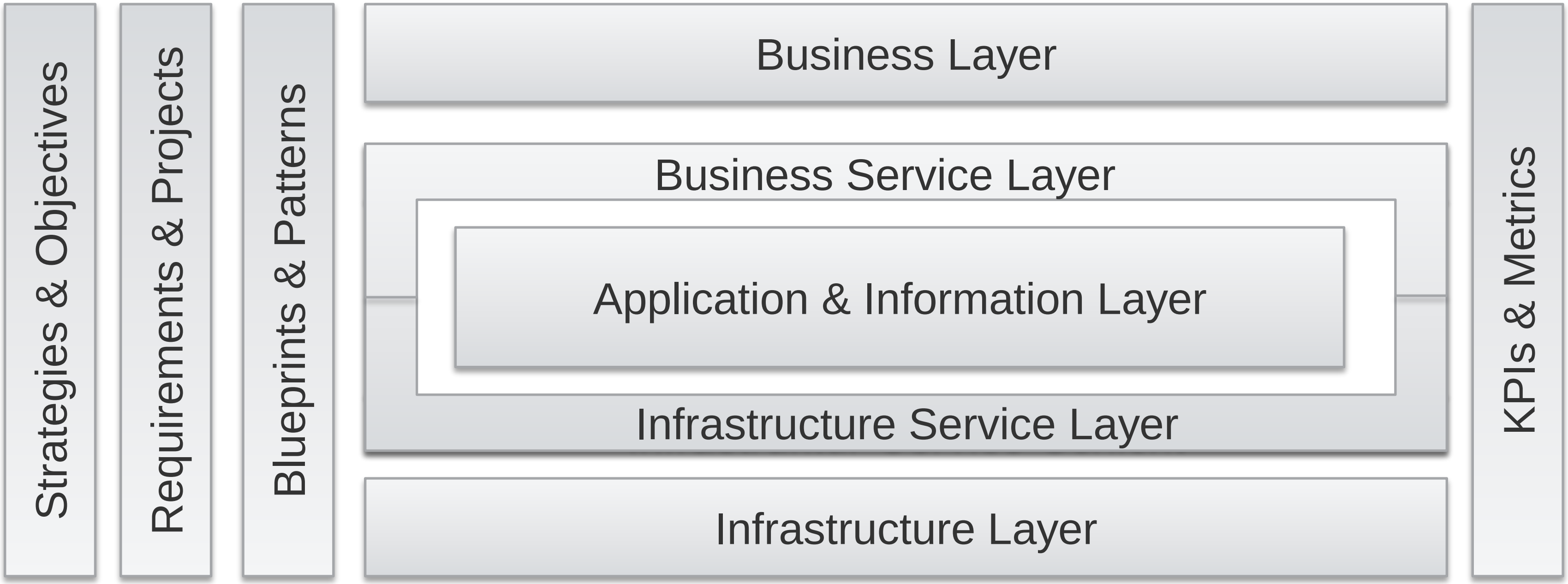
Introducing EAM in an enterprise is a challenge: EA frameworks provide only limited support



Application landscape management requires a holistic view

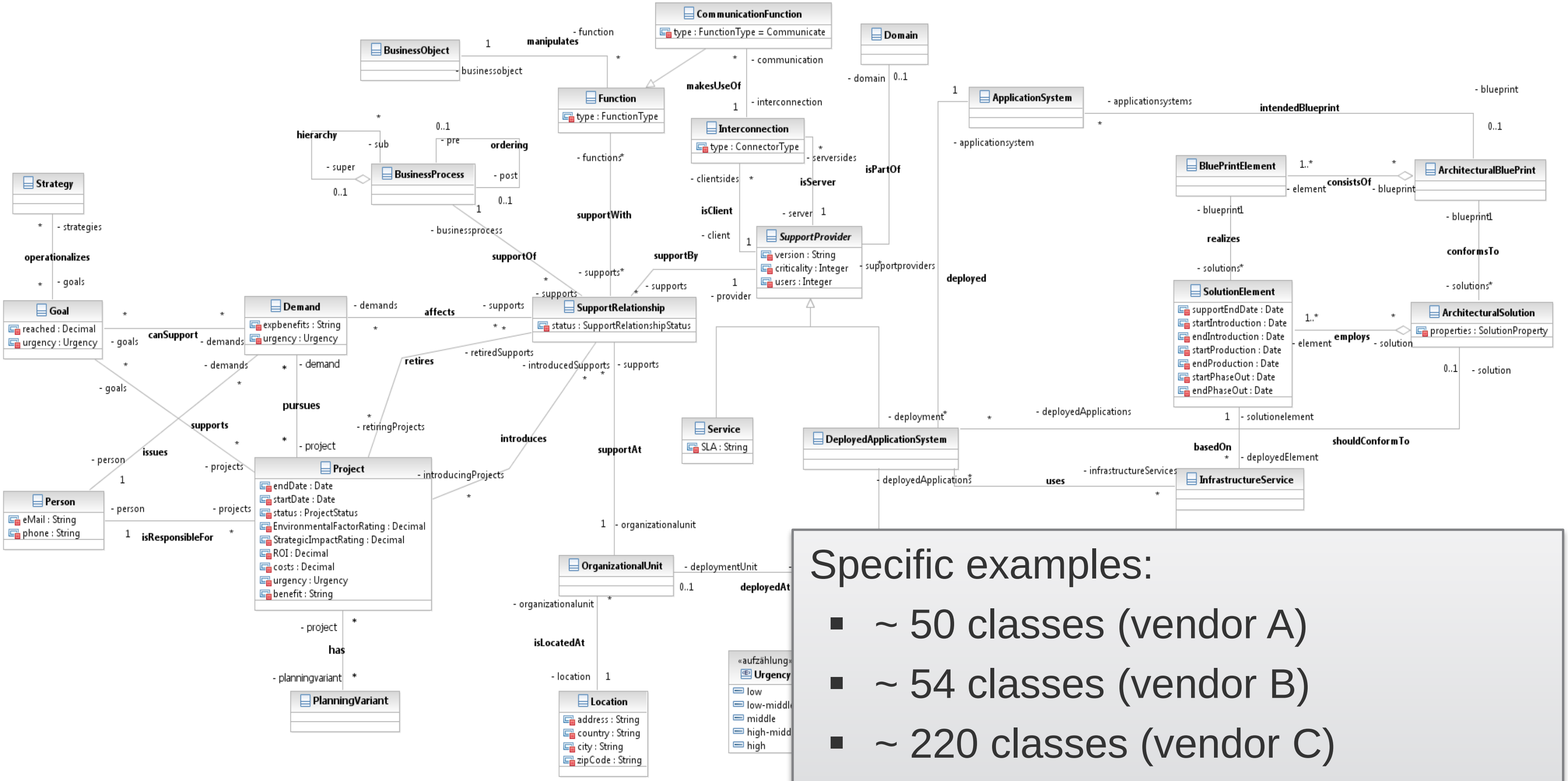
- Technical, social and economic aspects
- Layers and crosscutting concerns
- Relationships are more important than element details
  has, consists of, depends on, uses, controls, owns, produces, consumes,…

➔ Enterprise Architecture



Where to start? Which level of detail? Best practices?

Introducing EAM in an enterprise is a challenge: Information models are too complex



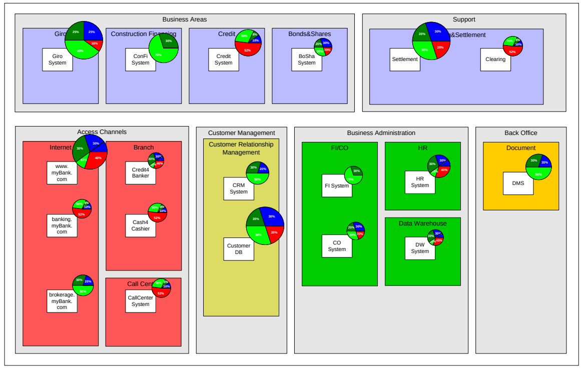
Specific examples:

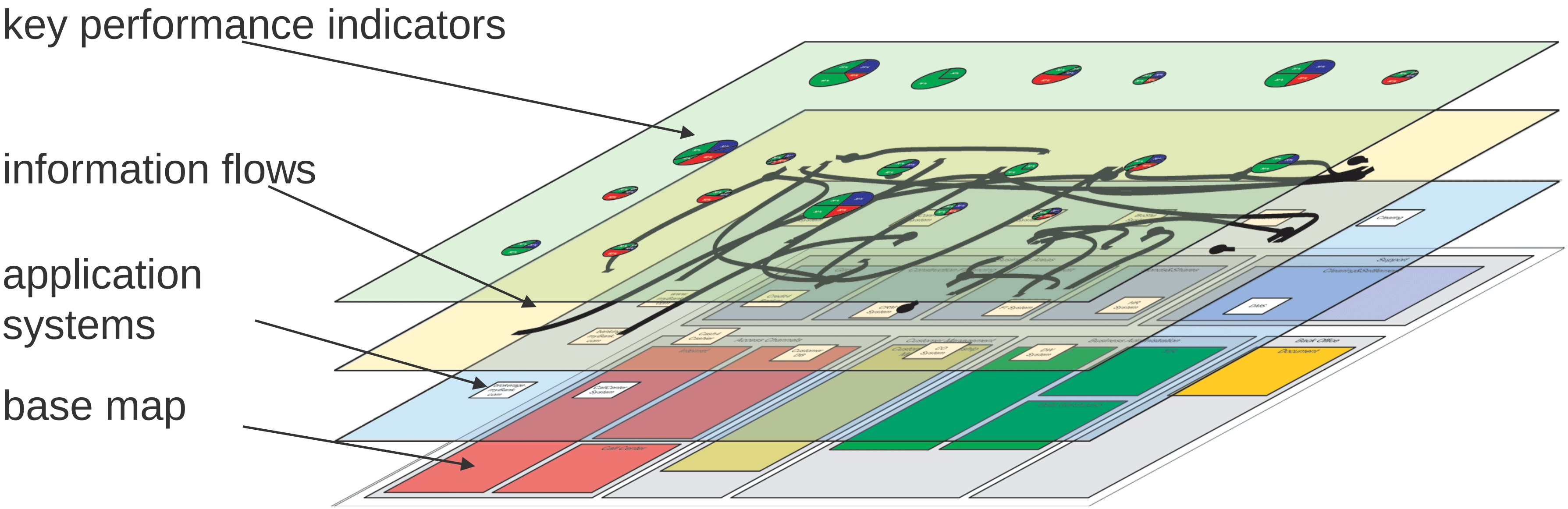
- ~ 50 classes (vendor A)
- ~ 54 classes (vendor B)
- ~ 220 classes (vendor C)
- ~ 470 classes (vendor D)
- At least twice as many associations
- Numerous attributes per instance

Software Cartography provides a visual language to communicate an enterprise architecture

Multiple viewpoints

- Shared problem-specific map types (base maps)
- Rule-based layout of visual elements
- Hide / show details based on layers





key performance indicators

information flows

application systems

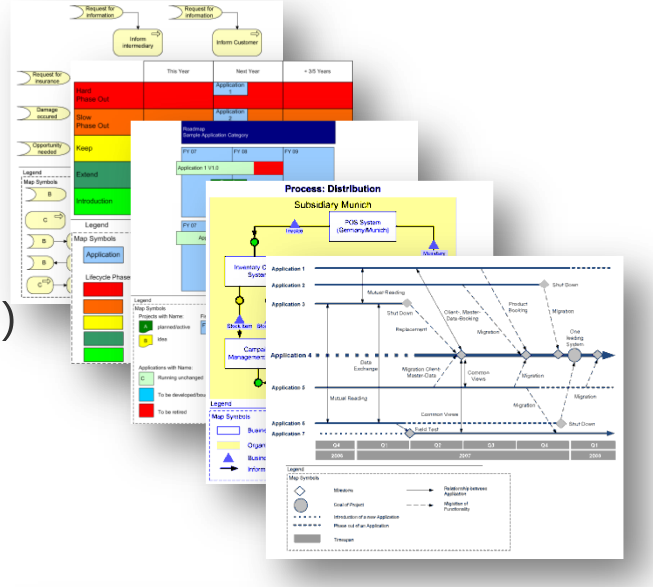
base map

Which viewpoints for which concerns & stakeholders?

Introducing EAM in an enterprise is a challenge: Lack of standardized EAM viewpoints

Software Engineering: Established viewpoints for recurring and known problems

- modularity, deployment, interaction, …

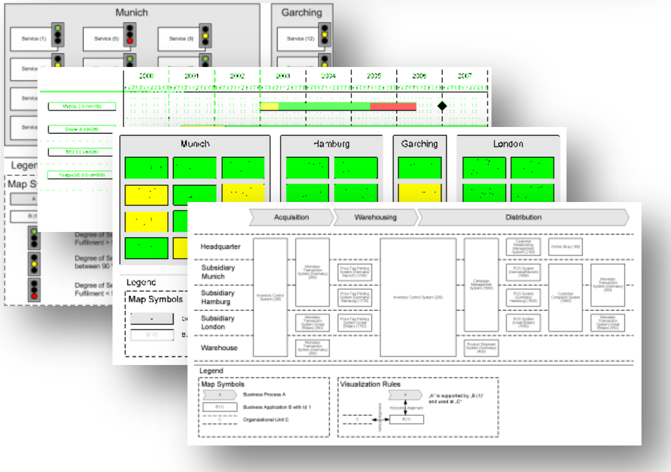


Enterprise Architectures: Emerging modeling languages and viewpoints, e.g.

- ArchiMate (http://www.archimate.com)
- Software Cartography (http://www.systemcartography.info)

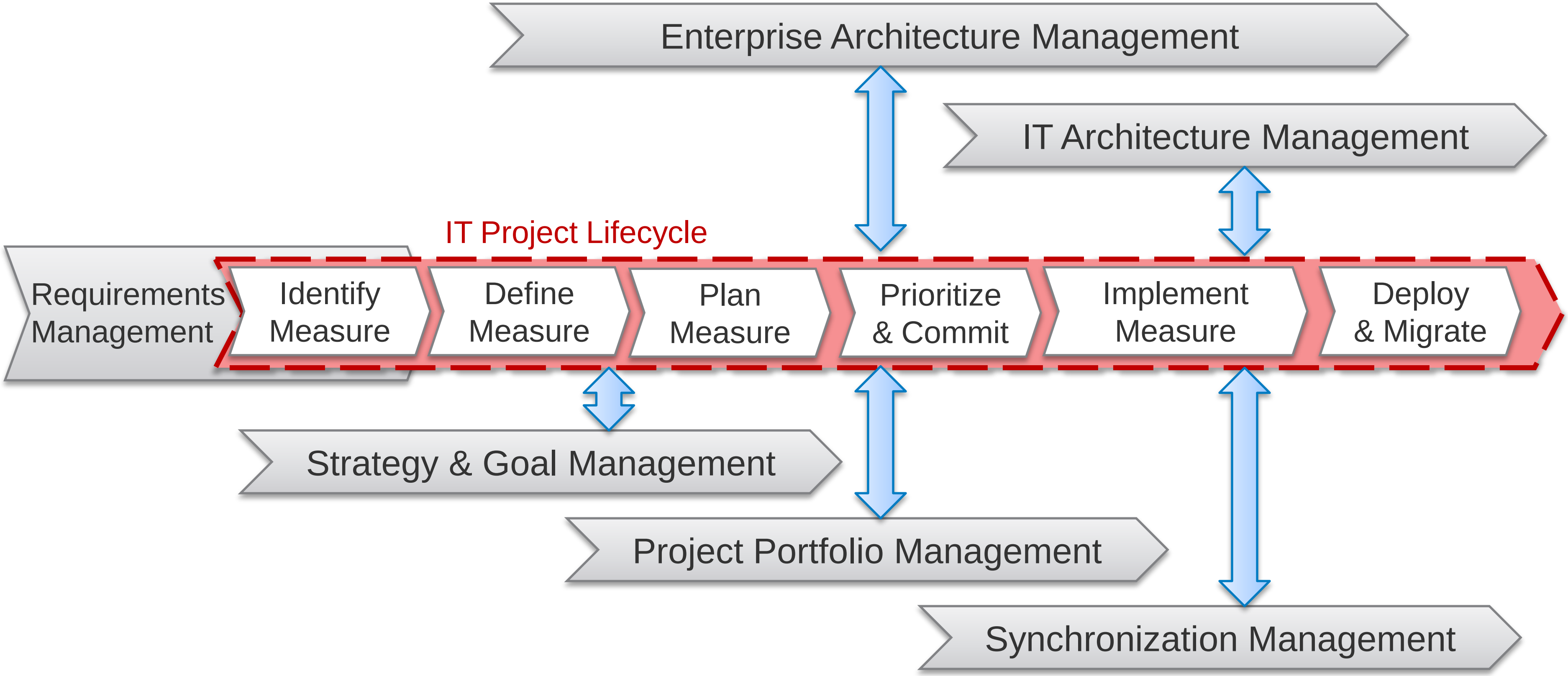
Many organization-specific viewpoints:

- rarely documented
- visibility limited to a single organization



The evolution of an application landscape can be improved by supporting management processes

IT-Governance Processes


Enterprise Architecture Management
IT Architecture Management
IT Project Lifecycle
Requirements Management
Identify Measure
Define Measure
Plan Measure
Prioritize & Commit
Implement Measure
Deploy & Migrate
Strategy & Goal Management
Project Portfolio Management
Synchronization Management

What are successful governance structures & management practices?

(2008)

Peer knowledge exchange advances EAM

- EUROFORUM, IIR conferences and seminars
- EAM Tage, act consulting
- SOA Innovation Lab, Deutsche Post
- CEISAR, Paris
- Systemkartographie Stammtisch, sebis
- IT Management Days, iteratec
- Cap Gemini sd&m EAM events
- EAM Think Tank, Syracom
- …

How to capture, disseminate and apply this empirical knowledge?

Outline

1. Introducing EAM in an enterprise is a challenge
   - models, viewpoints, management processes

2. The EAM pattern catalog 1.0
   - rationale, contents, contributors

3. Towards an EAM pattern community

A Patterns is a general, reusable solution to a common problem in a given context

Analogy to other disciplines: Address recurring problems with patterns.

Alexander et al. [Al77] (**Architecture**)

- Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.
- Each pattern is a three-part rule, which expresses a relation between a certain context, a problem and a solution

Buschmann et al. [Bu96] (**Software Architecture**)

- A pattern for software architecture describes a particular recurring design problem that arises in specific design contexts, and presents a well-proven generic scheme for its solution. The solution scheme is specified by describing its constituent components, their responsibilities and relationships, and the ways in which they collaborate

Gamma et al. [Ga94] (**Software Engineering**)

- Descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context.

Enterprise architecture management patterns

An enterprise architecture management pattern (**EAM pattern**) is

- a general, reusable solution to a common problem
- in a given context
- identifies driving forces,
- known usages and
- consequences.

An EAM pattern takes a **holistic perspective**:

- It address concerns at the enterprise (systems of systems) level.
- It considers social, technical and economic forces in a balanced manner.

- It is discovered in working solutions rather than being invented or hoped for.
- It uses a clear, accessible and informal language that allows practitioners to describe their knowledge and experience.

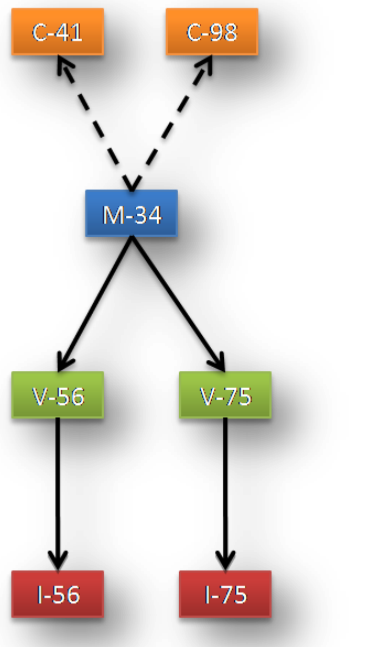
Pattern languages are a proven way to capture best practices and expert knowledge and to socialize it inside a group, department, entire company, or entire design discipline.

The idea behind the EAM pattern catalog 1.0

Tailor the EAM to the specific situation (*pains*) of the enterprise and follow an incremental strategy based on **EAM patterns** representing proven practices.

Systematically document the dependencies between

- Individual management concerns,
  Which concern is relevant for which stakeholder?
- Methodology patterns (M-Pattern),
  Which activities are required to address a concern?
- Viewpoint patterns (V-Pattern) and
  Which viewpoints help stakeholders to collaboratively perform the activities?
- Information model patterns (I-Pattern)
  Which information has to be available to generate a view?

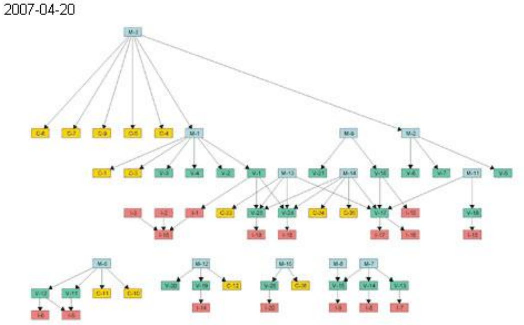


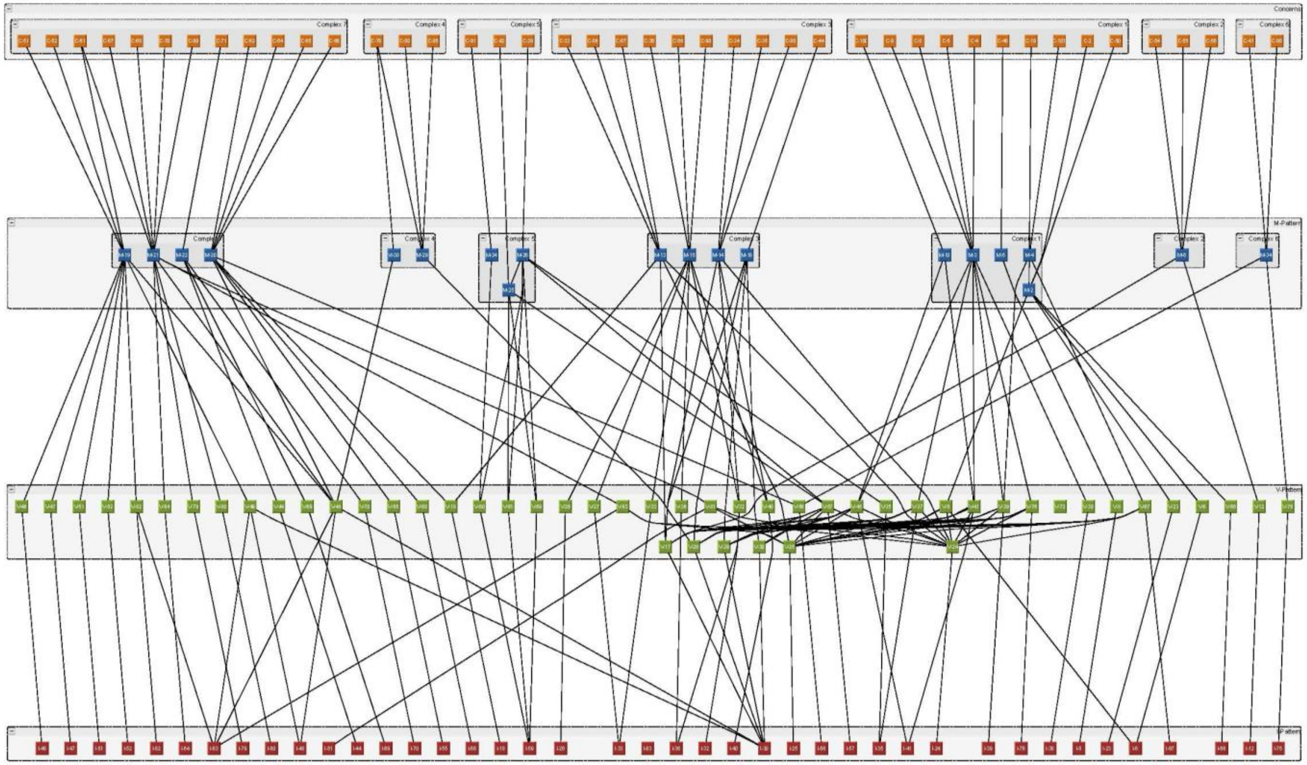
Draw attention to the consequences implied by a pattern (labor, required information, *political* resistance, …)

Overview of the pattern catalog version 1.0

- Basis: literature, experience from *sebis* research projects, structured interviews of 25 enterprise architects
- Selection based on relevance and adoption by an extensive online questionnaire

➔ 43 concerns, 20 M-Patterns, 53 V-Patterns, and 47 I-Patterns

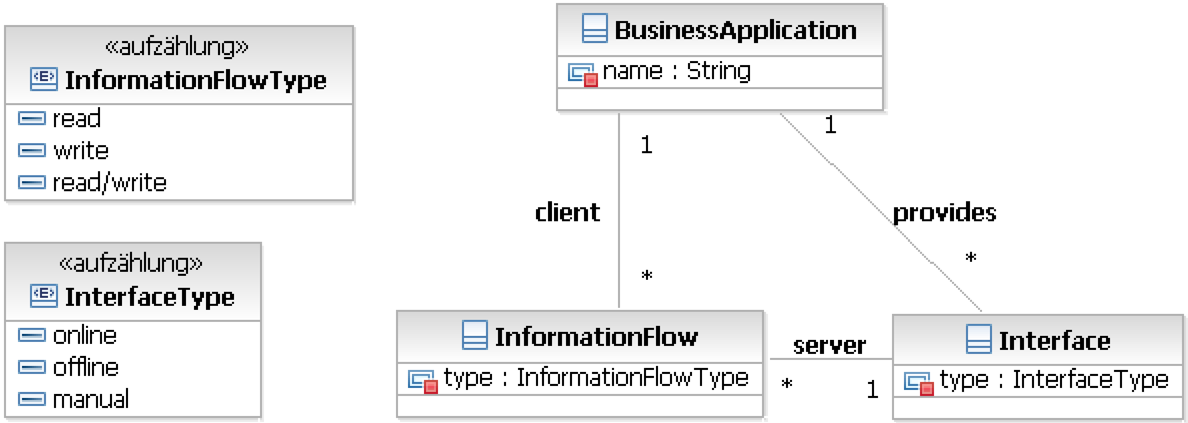

2007-04-20



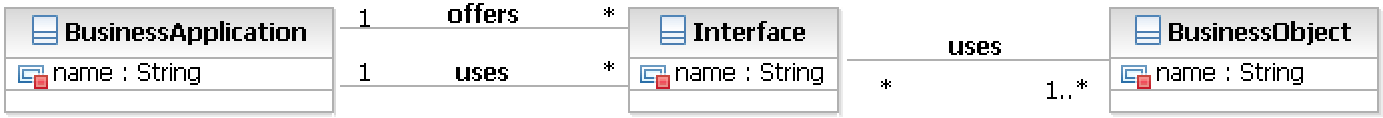
Using patterns: Constructing EAM information models based on I-Patterns

I-Pattern as manageable, concern-specific units, which can be composed to an integrated information model.

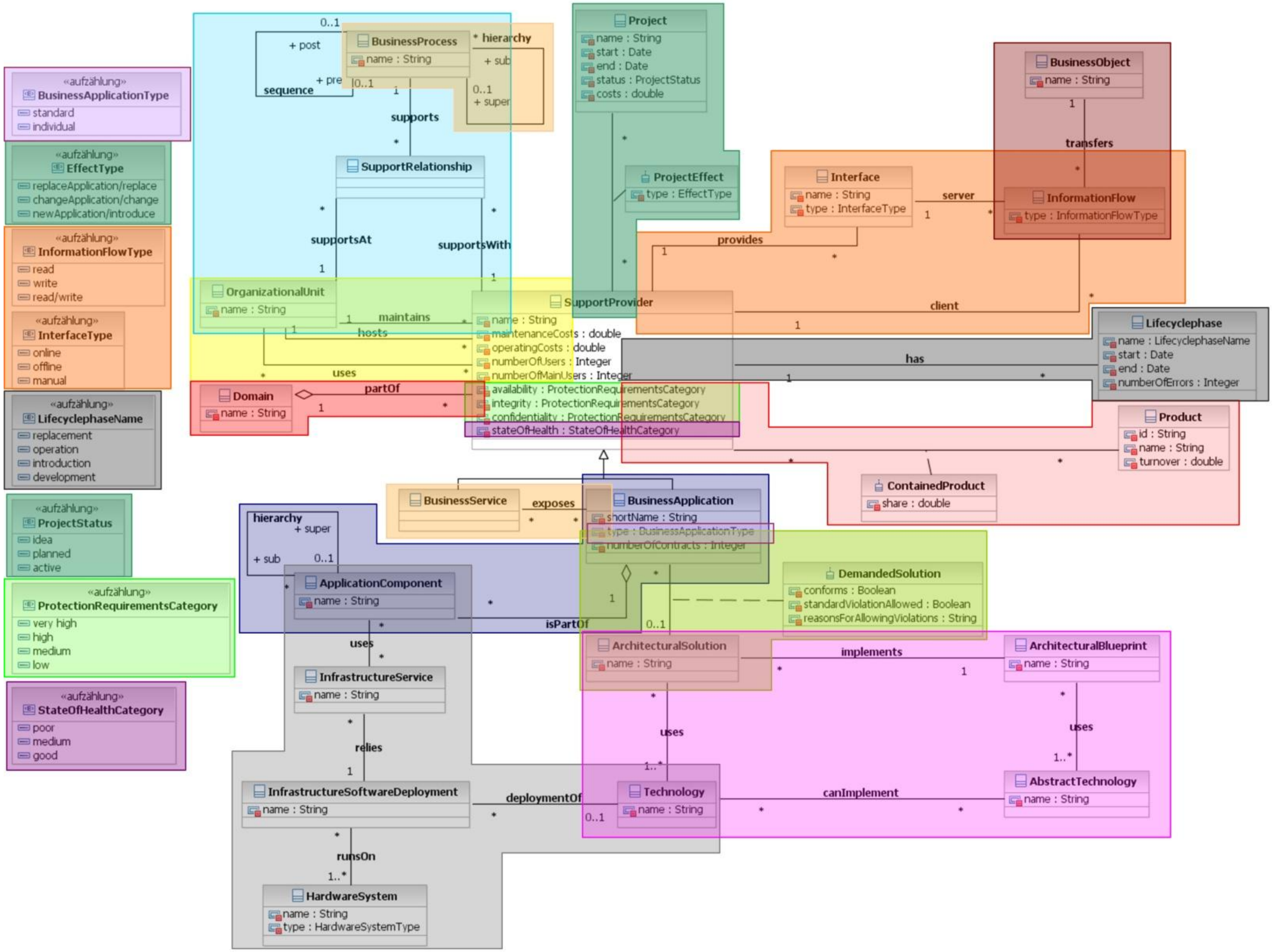
I-63



I-82



Information model with highlighted I-Patterns



Legend:

🟧	I-18 parts	🟪	I-66 modified
🟨	I-24* modified	🟧	I-63 modified
⬜	I-26 modified	🟩	I-67 complete
🟦	I-30 modified	🟦	I-84 complete
🟩	I-36 modified	🟩	I-86 modified
🟪	I-41 complete	🟪	I-87 modified
🟫	I-48 parts	🟥	I-88 modified
🟥	I-55 modified		

Contributors to the EAM pattern catalog 1.0

Allianz

AXA

BMW

MINI

B/S/H/

BTC

Credit Suisse

Deutsche Börse Systems

Die Bahn DB

Deutsche Post

EWE

FIDUCIA
Ihr IT-Partner

FJH

HVB Information Services
Member of UniCredit Group

KRONES

KUEHNE+NAGEL

Münchener Rück
Munich Re Group

Nokia Siemens Networks

O_2

RWE

SIEMENS

cirquent
credible consulting
softlab group

T

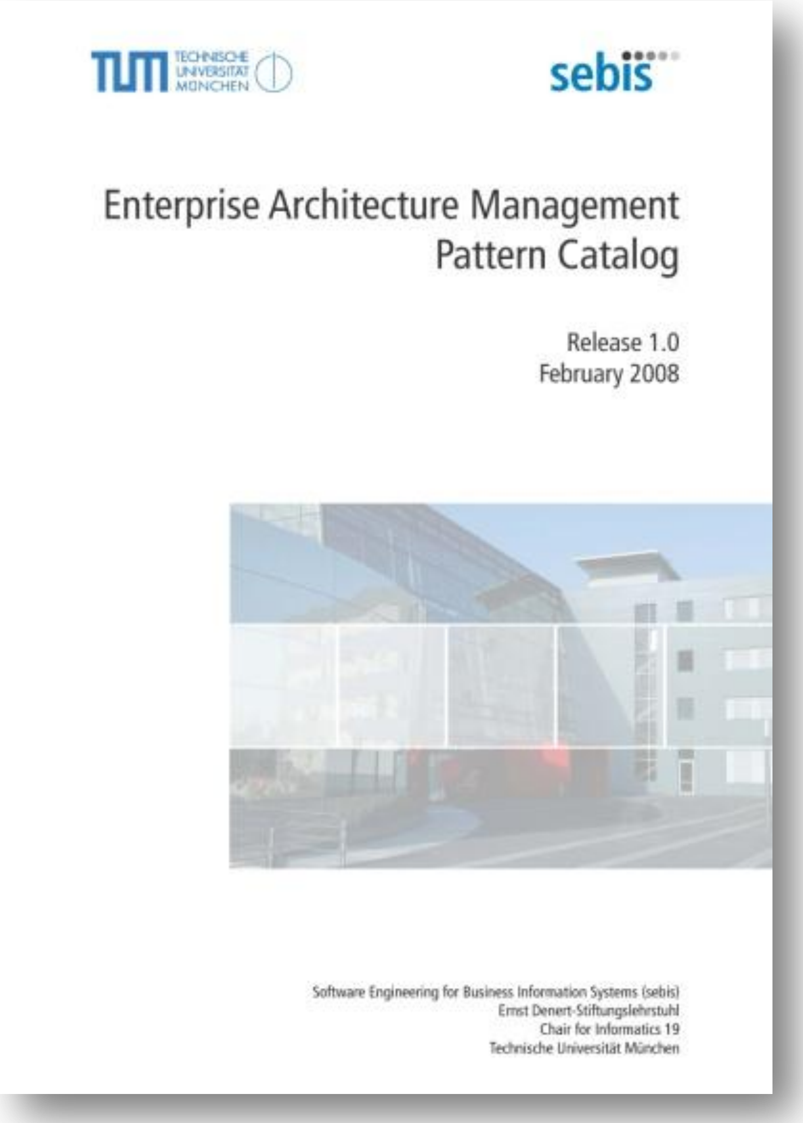
WACKER

zollner

Using the EAM pattern catalog

1. Develop enterprise-specific EA management processes, governance structures, and information models
2. Evolve and assess existing EA management approaches in the enterprise
3. Specify EA requirements to
   - select an EAM tool
   - clarify goals of an EAM approach

4. Conduct scientific research
   - Evolve and validate individual patterns
   - Develop domain-specific patterns (financial sector, health care, …)
   - Analyze relationships between management patterns, maturity models, …



TUM TECHNISCHE UNIVERSITÄT MÜNCHEN

sebis

Enterprise Architecture Management
Pattern Catalog

Release 1.0
February 2008

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

Outline

1. Introducing EAM in an enterprise is a challenge
   - models, viewpoints, management processes

2. The EAM pattern catalog 1.0
   - rationale, contents, contributors

3. Towards an EAM pattern community

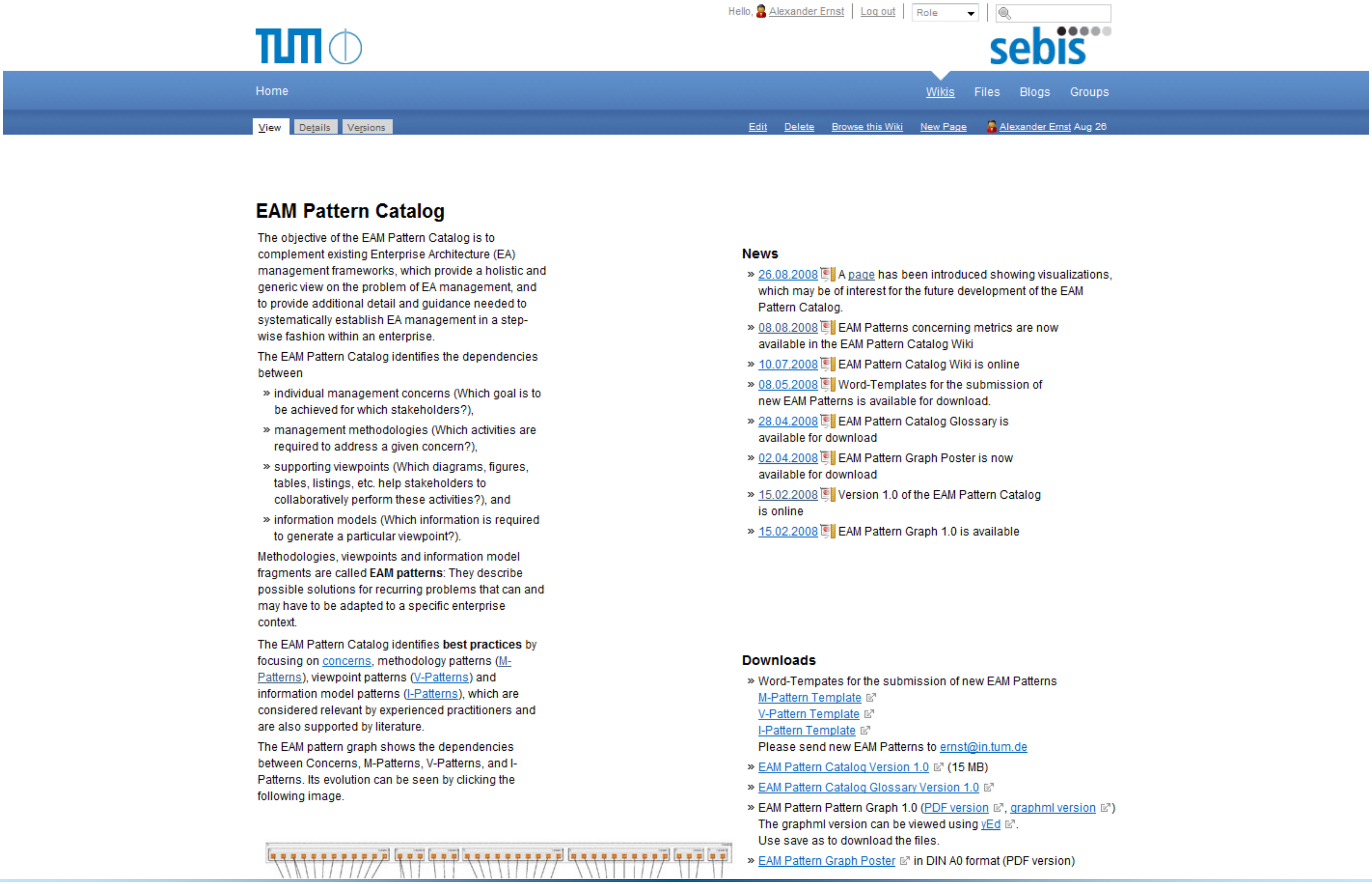
From a pattern catalog towards a pattern language

- Make EAM patterns more self-contained
  - Each EAM pattern (M-Pattern, V-Pattern, and I-Pattern) addresses a specific problem affected by its forces
- Restructure EAM patterns
  - Pattern description similar to Buschmann et al. [Bu96]
- Enrich existing EAM patterns
  - Example, implementation, usage, extended solution, …
- Develop new EAM patterns in cooperation with practitioners and academia

See [Er08] for these changes and further development.

[Bu96] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. *Pattern-oriented software architecture: a system of patterns*. John Wiley & Sons, Inc., New York, NY, USA, 1996.
[Er08] Ernst, A.: *Enterprise Architecture Management Patterns*. Pattern Languages of Programs Conference 2008 (PLoP08), Nashville, 2008. (in publication)

EAM pattern catalog wiki
http://www.systemcartography.info/eampc-wiki


Hello, Alexander Ernst | Log out | Role
Home
Wikis Files Blogs Groups
View Details Versions
Edit Delete Browse this Wiki New Page Alexander Ernst Aug 26

EAM Pattern Catalog

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

The EAM Pattern Catalog identifies the dependencies between

- individual management concerns (Which goal is to be achieved for which stakeholders?),
- management methodologies (Which activities are required to address a given concern?),
- supporting viewpoints (Which diagrams, figures, tables, listings, etc. help stakeholders to collaboratively perform these activities?), and
- information models (Which information is required to generate a particular viewpoint?).

Methodologies, viewpoints and information model fragments are called **EAM patterns**: They describe possible solutions for recurring problems that can and may have to be adapted to a specific enterprise context.

The EAM Pattern Catalog identifies **best practices** by focusing on concerns, methodology patterns (M-Patterns), viewpoint patterns (V-Patterns) and information model patterns (I-Patterns), which are considered relevant by experienced practitioners and are also supported by literature.

The EAM pattern graph shows the dependencies between Concerns, M-Patterns, V-Patterns, and I-Patterns. Its evolution can be seen by clicking the following image.

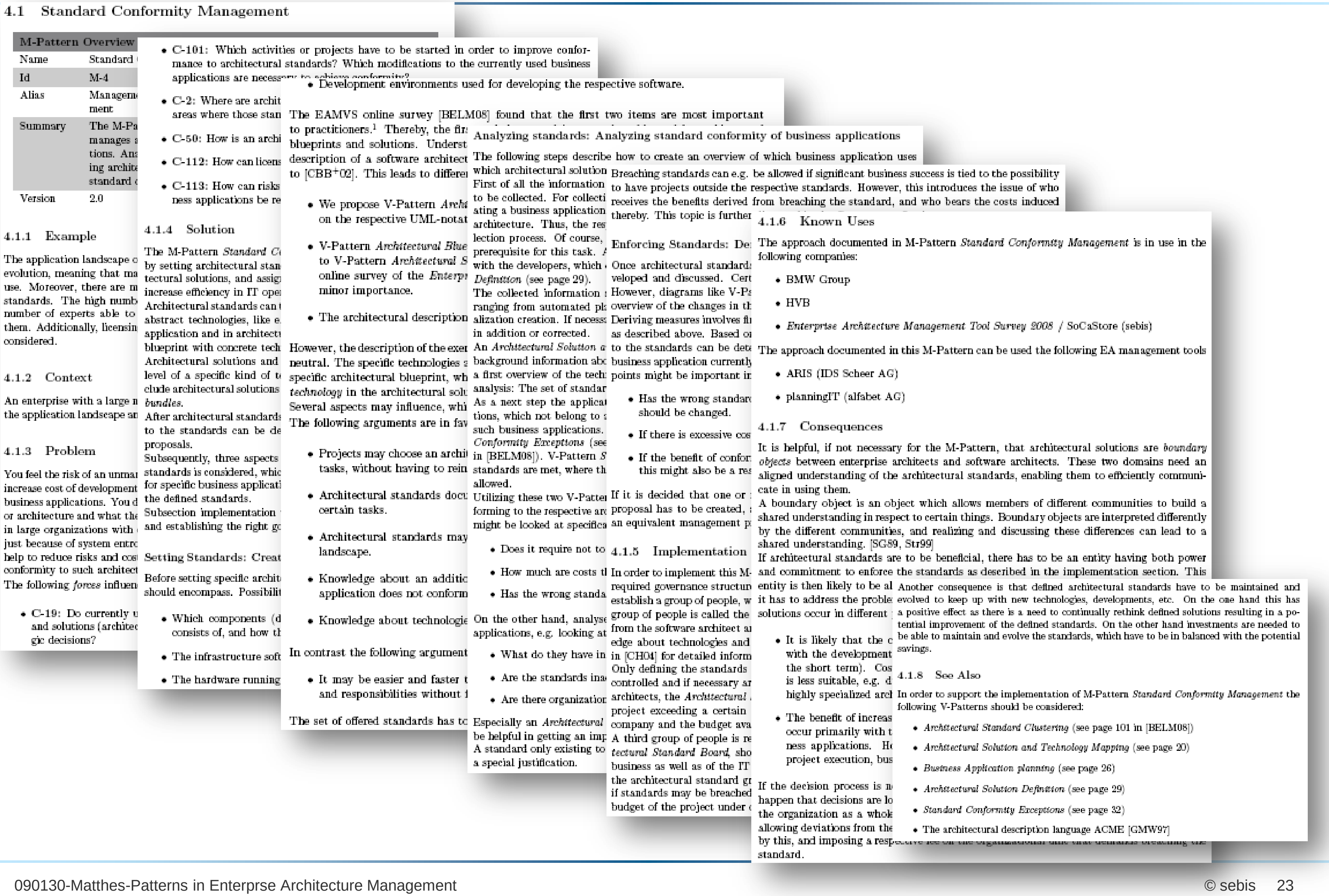
News

- 26.08.2008 A page has been introduced showing visualizations, which may be of interest for the future development of the EAM Pattern Catalog.
- 08.08.2008 EAM Patterns concerning metrics are now available in the EAM Pattern Catalog Wiki
- 10.07.2008 EAM Pattern Catalog Wiki is online
- 08.05.2008 Word-Templates for the submission of new EAM Patterns is available for download.
- 28.04.2008 EAM Pattern Catalog Glossary is available for download
- 02.04.2008 EAM Pattern Graph Poster is now available for download
- 15.02.2008 Version 1.0 of the EAM Pattern Catalog is online
- 15.02.2008 EAM Pattern Graph 1.0 is available

Downloads

- Word-Tempates for the submission of new EAM Patterns
  M-Pattern Template
  V-Pattern Template
  I-Pattern Template
  Please send new EAM Patterns to ernst@in.tum.de
- EAM Pattern Catalog Version 1.0 (15 MB)
- EAM Pattern Catalog Glossary Version 1.0
- EAM Pattern Pattern Graph 1.0 (PDF version, graphml version)
  The graphml version can be viewed using yEd.
  Use save as to download the files.
- EAM Pattern Graph Poster in DIN A0 format (PDF version)

Example M-Pattern – Standard Conformity Management



4.1 Standard Conformity Management

M-Pattern Overview	
Name	Standard
Id	M-4
Alias	Manageme ment
Summary	The M-Pa manages tions. Ana ing archite standard
Version	2.0

4.1.1 Example

The application landscape o
evolution, meaning that ma
use. Moreover, there are m
standards. The high numb
number of experts able to
them. Additionally, licensin
considered.

4.1.2 Context

An enterprise with a large n
the application landscape an

4.1.3 Problem

You feel the risk of an unmar
increase cost of development
business applications. You d
or architecture and what the
in large organizations with
just because of system entro
help to reduce risks and cost
conformity to such architect
The following *forces* influen

- C-19: Do currently u
and solutions (architec
gic decisions?

- C-101: Which activities or projects have to be started in order to improve conformance to architectural standards? Which modifications to the currently used business applications are necessary to achieve conformity?
- C-2: Where are archit
areas where those stan
- C-50: How is an archi
- C-112: How can licens
- C-113: How can risks
ness applications be re

4.1.4 Solution

The M-Pattern *Standard C*
by setting architectural stan
tectural solutions, and assig
increase efficiency in IT oper
Architectural standards can
abstract technologies, like e.
application and in architect
blueprint with concrete tech
Architectural solutions and
level of a specific kind of t
clude architectural solutions
bundles.
After architectural standards
to the standards can be de
proposals.
Subsequently, three aspects
standards is considered, whic
for specific business applicati
the defined standards.
Subsection implementation
and establishing the right g

Setting Standards: Creat

Before setting specific archit
should encompass. Possibilit

- Which components (d
consists of, and how th
- The infrastructure soft
- The hardware running

- Development environments used for developing the respective software.

The EAMVS online survey [BELM08] found that the first two items are most important
to practitioners.[1] Thereby, the fir
blueprints and solutions. Underst
description of a software architect
to [CBB+02]. This leads to differe

- We propose V-Pattern *Archi*
on the respective UML-notat
- V-Pattern *Architectural Blue*
to V-Pattern *Architectural S*
online survey of the *Enterpr*
minor importance.
- The architectural description

However, the description of the exe
neutral. The specific technologies
specific architectural blueprint, wh
technology in the architectural solu
Several aspects may influence, whi
The following arguments are in fav

- Projects may choose an archit
tasks, without having to rein
- Architectural standards docu
certain tasks.
- Architectural standards may
landscape.
- Knowledge about an additio
application does not conform
- Knowledge about technologie

In contrast the following argument

- It may be easier and faster t
and responsibilities without

The set of offered standards has to

Analyzing standards: Analyzing standard conformity of business applications

The following steps describe how to create an overview of which business application uses
which architectural solution
First of all the information
to be collected. For collecti
ating a business application
architecture. Thus, the res
lection process. Of course,
prerequisite for this task.
with the developers, which
Definition (see page 29).
The collected information
ranging from automated pl
alization creation. If necess
in addition or corrected.
An *Architectural Solution*
background information abo
a first overview of the tech
analysis: The set of standar
As a next step the applica
tions, which not belong to
such business applications.
Conformity Exceptions (see
in [BELM08]). V-Pattern *S*
standards are met, where th
allowed.
Utilizing these two V-Patte
forming to the respective ar
might be looked at specifica

- Does it require not to
- How much are costs t
- Has the wrong standa

On the other hand, analyse
applications, e.g. looking at

- What do they have in
- Are the standards ina
- Are there organization

Especially an *Architectural*
be helpful in getting an imp
A standard only existing to
a special justification.

Breaching standards can e.g. be allowed if significant business success is tied to the possibility
to have projects outside the respective standards. However, this introduces the issue of who
receives the benefits derived from breaching the standard, and who bears the costs induced
thereby. This topic is further

Enforcing Standards: De

Once architectural standard
veloped and discussed. Cert
However, diagrams like V-P
overview of the changes in th
Deriving measures involves fi
as described above. Based o
to the standards can be dete
business application currently
points might be important in

- Has the wrong standard
should be changed.
- If there is excessive cos
- If the benefit of confor
this might also be a res

If it is decided that one or
proposal has to be created,
an equivalent management p

4.1.5 Implementation

In order to implement this M-
required governance structur
establish a group of people, w
group of people is called the
from the software architect a
edge about technologies and
in [CH04] for detailed inform
Only defining the standards
controlled and if necessary a
architects, the *Architectural*
project exceeding a certain
company and the budget ava
A third group of people is re
tectural Standard Board, sho
business as well as of the IT
the architectural standard g
if standards may be breached
budget of the project under

4.1.6 Known Uses

The approach documented in M-Pattern *Standard Conformity Management* is in use in the following companies:

- BMW Group
- HVB
- *Enterprise Architecture Management Tool Survey 2008* / SoCaStore (sebis)

The approach documented in this M-Pattern can be used the following EA management tools

- ARIS (IDS Scheer AG)
- planningIT (alfabet AG)

4.1.7 Consequences

It is helpful, if not necessary for the M-Pattern, that architectural solutions are *boundary objects* between enterprise architects and software architects. These two domains need an aligned understanding of the architectural standards, enabling them to efficiently communicate in using them.
A boundary object is an object which allows members of different communities to build a shared understanding in respect to certain things. Boundary objects are interpreted differently by the different communities, and realizing and discussing these differences can lead to a shared understanding. [SG89, Str99]
If architectural standards are to be beneficial, there has to be an entity having both power and commitment to enforce the standards as described in the implementation section. This
entity is then likely to be al
it has to address the proble
solutions occur in different

- It is likely that the c
with the development
the short term). Cos
is less suitable, e.g. d
highly specialized arc
- The benefit of increas
occur primarily with t
ness applications. H
project execution, bus

If the decision process is n
happen that decisions are lo
the organization as a whol
allowing deviations from the
by this, and imposing a resp
standard.

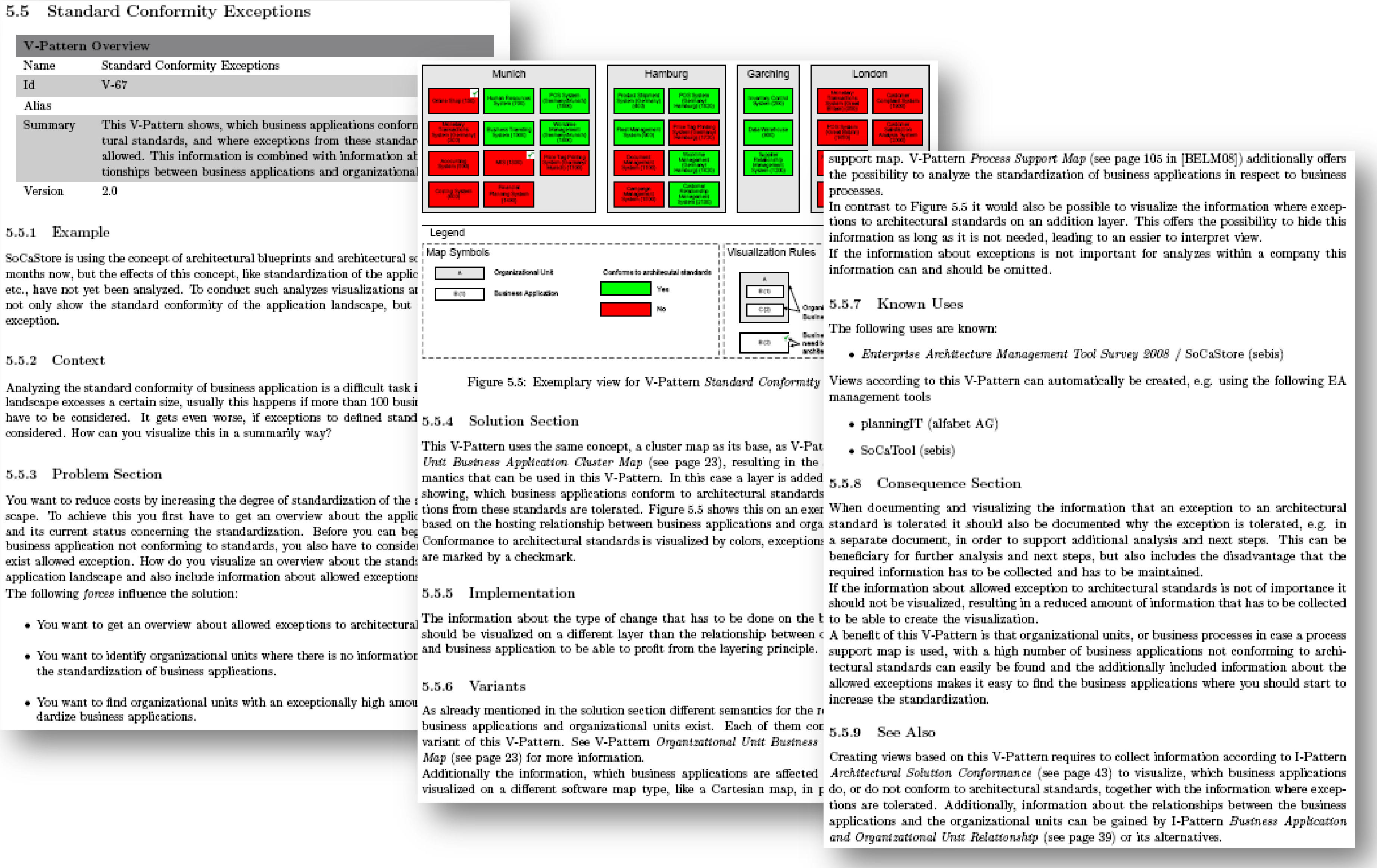
Another consequence is that defined architectural standards have to be maintained and evolved to keep up with new technologies, developments, etc. On the one hand this has a positive effect as there is a need to continually rethink defined solutions resulting in a potential improvement of the defined standards. On the other hand investments are needed to be able to maintain and evolve the standards, which have to be in balanced with the potential savings.

4.1.8 See Also

In order to support the implementation of M-Pattern *Standard Conformity Management* the following V-Patterns should be considered:

- *Architectural Standard Clustering* (see page 101 in [BELM08])
- *Architectural Solution and Technology Mapping* (see page 20)
- *Business Application planning* (see page 26)
- *Architectural Solution Definition* (see page 29)
- *Standard Conformity Exceptions* (see page 32)
- The architectural description language ACME [GMW97]

Example V-Pattern – Standard Conformity Exceptions


Munich
Online Shop (100)
Human Resources System (700)
POS System (Germany/Munich) (1900)
Monetary Transactions System (Germany) (300)
Business Traveling System (1000)
Worktime Management (Germany/Munich) (1900)
Accounting System (500)
MIS (1500)
Price Tag Printing System (Germany/Munich) (1100)
Costing System (600)
Financial Planning System (1400)
Hamburg
Product Shipment System (Germany) (400)
POS System (Germany/Hamburg) (1800)
Fleet Management System (900)
Price Tag Printing System (Germany/Hamburg) (1700)
Document Management System (1100)
Worktime Management (Germany/Hamburg) (1600)
Campaign Management System (1200)
Customer Relationship Management System (2100)
Garching
Inventory Control System (200)
Data Warehouse (800)
Supplier Relationship Management System (1300)
London
Monetary Transactions System (Great Britain) (350)
Customer Complaint System (1000)
POS System (Great Britain) (1650)
Customer Satisfaction Analysis System (2000)
Legend
Map Symbols
A
Organizational Unit
B(1)
Business Application
Conforms to architectural standards
Yes
No
Visualization Rules
A
B(1)
C(2)
Organi
Busine
B(3)
Busine
need b
archite

5.5 Standard Conformity Exceptions

V-Pattern Overview	
Name	Standard Conformity Exceptions
Id	V-67
Alias	
Summary	This V-Pattern shows, which business applications conforn tural standards, and where exceptions from these standar allowed. This information is combined with information ab tionships between business applications and organizational
Version	2.0

5.5.1 Example

SoCaStore is using the concept of architectural blueprints and architectural sc
months now, but the effects of this concept, like standardization of the applic
etc., have not yet been analyzed. To conduct such analyzes visualizations ar
not only show the standard conformity of the application landscape, but
exception.

5.5.2 Context

Analyzing the standard conformity of business application is a difficult task i
landscape excesses a certain size, usually this happens if more than 100 busin
have to be considered. It gets even worse, if exceptions to defined stand
considered. How can you visualize this in a summarily way?

5.5.3 Problem Section

You want to reduce costs by increasing the degree of standardization of the a
scape. To achieve this you first have to get an overview about the applic
and its current status concerning the standardization. Before you can beg
business application not conforming to standards, you also have to conside
exist allowed exception. How do you visualize an overview about the stand
application landscape and also include information about allowed exceptions
The following *forces* influence the solution:

- You want to get an overview about allowed exceptions to architectural
- You want to identify organizational units where there is no informatior
  the standardization of business applications.
- You want to find organizational units with an exceptionally high amou
  dardize business applications.

Figure 5.5: Exemplary view for V-Pattern *Standard Conformity*

5.5.4 Solution Section

This V-Pattern uses the same concept, a cluster map as its base, as V-Pat
Unit Business Application Cluster Map (see page 23), resulting in the
mantics that can be used in this V-Pattern. In this case a layer is added
showing, which business applications conform to architectural standards
tions from these standards are tolerated. Figure 5.5 shows this on an exer
based on the hosting relationship between business applications and orga
Conformance to architectural standards is visualized by colors, exceptions
are marked by a checkmark.

5.5.5 Implementation

The information about the type of change that has to be done on the b
should be visualized on a different layer than the relationship between c
and business application to be able to profit from the layering principle.

5.5.6 Variants

As already mentioned in the solution section different semantics for the r
business applications and organizational units exist. Each of them cor
variant of this V-Pattern. See V-Pattern *Organizational Unit Business*
Map (see page 23) for more information.
Additionally the information, which business applications are affected
visualized on a different software map type, like a Cartesian map, in p

support map. V-Pattern *Process Support Map* (see page 105 in [BELM08]) additionally offers the possibility to analyze the standardization of business applications in respect to business processes.
In contrast to Figure 5.5 it would also be possible to visualize the information where exceptions to architectural standards on an addition layer. This offers the possibility to hide this information as long as it is not needed, leading to an easier to interpret view.
If the information about exceptions is not important for analyzes within a company this information can and should be omitted.

5.5.7 Known Uses

The following uses are known:

- *Enterprise Architecture Management Tool Survey 2008* / SoCaStore (sebis)

Views according to this V-Pattern can automatically be created, e.g. using the following EA management tools

- planningIT (alfabet AG)
- SoCaTool (sebis)

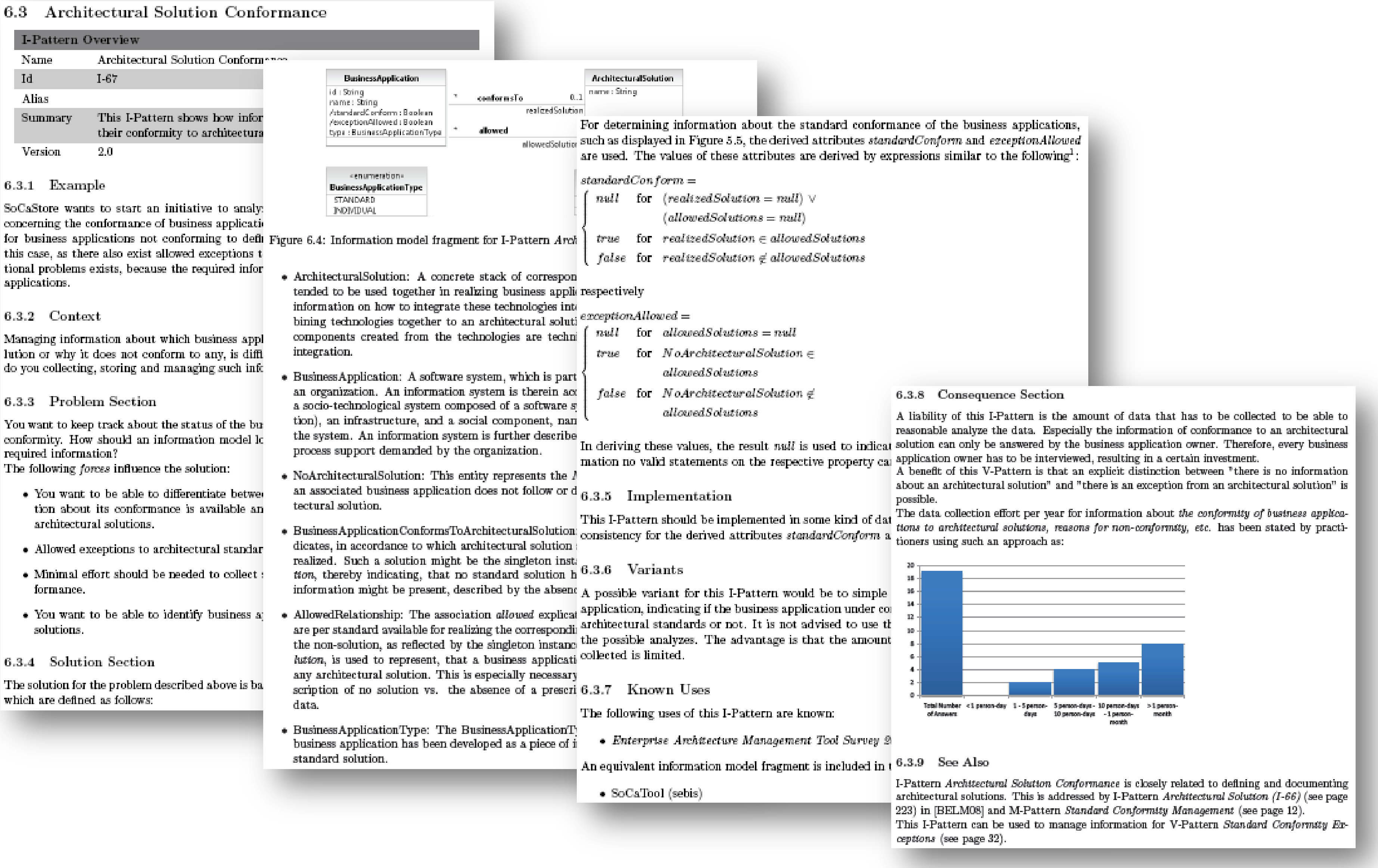
5.5.8 Consequence Section

When documenting and visualizing the information that an exception to an architectural standard is tolerated it should also be documented why the exception is tolerated, e.g. in a separate document, in order to support additional analysis and next steps. This can be beneficiary for further analysis and next steps, but also includes the disadvantage that the required information has to be collected and has to be maintained.
If the information about allowed exception to architectural standards is not of importance it should not be visualized, resulting in a reduced amount of information that has to be collected to be able to create the visualization.
A benefit of this V-Pattern is that organizational units, or business processes in case a process support map is used, with a high number of business applications not conforming to architectural standards can easily be found and the additionally included information about the allowed exceptions makes it easy to find the business applications where you should start to increase the standardization.

5.5.9 See Also

Creating views based on this V-Pattern requires to collect information according to I-Pattern *Architectural Solution Conformance* (see page 43) to visualize, which business applications do, or do not conform to architectural standards, together with the information where exceptions are tolerated. Additionally, information about the relationships between the business applications and the organizational units can be gained by I-Pattern *Business Application and Organizational Unit Relationship* (see page 39) or its alternatives.

Example I-Pattern – Architectural Solution Conformance



6.3 Architectural Solution Conformance

I-Pattern Overview	
Name	Architectural Solution Conform
Id	I-67
Alias	
Summary	This I-Pattern shows how infor their conformity to architectura
Version	2.0

6.3.1 Example

SoCaStore wants to start an initiative to analy
concerning the conformance of business applicati
for business applications not conforming to defi
this case, as there also exist allowed exceptions t
tional problems exists, because the required infor
applications.

6.3.2 Context

Managing information about which business appl
lution or why it does not conform to any, is diffi
do you collecting, storing and managing such info

6.3.3 Problem Section

You want to keep track about the status of the bu
conformity. How should an information model lo
required information?
The following *forces* influence the solution:

- You want to be able to differentiate betwe
tion about its conformance is available an
architectural solutions.
- Allowed exceptions to architectural standar
- Minimal effort should be needed to collect
formance.
- You want to be able to identify business a
solutions.

6.3.4 Solution Section

The solution for the problem described above is ba
which are defined as follows:

Figure 6.4: Information model fragment for I-Pattern *Arch*

- ArchitecturalSolution: A concrete stack of correspon
tended to be used together in realizing business appli
information on how to integrate these technologies int
bining technologies together to an architectural soluti
components created from the technologies are techni
integration.
- BusinessApplication: A software system, which is part
an organization. An information system is therein acc
a socio-technological system composed of a software s
tion), an infrastructure, and a social component, nan
the system. An information system is further describe
process support demanded by the organization.
- NoArchitecturalSolution: This entity represents the *N*
an associated business application does not follow or d
tectural solution.
- BusinessApplicationConformsToArchitecturalSolution
dicates, in accordance to which architectural solution
realized. Such a solution might be the singleton inst
tion, thereby indicating, that no standard solution h
information might be present, described by the absenc
- AllowedRelationship: The association *allowed* explicat
are per standard available for realizing the correspondi
the non-solution, as reflected by the singleton instanc
lution, is used to represent, that a business applicati
any architectural solution. This is especially necessary
scription of no solution vs. the absence of a prescri
data.
- BusinessApplicationType: The BusinessApplicationT
business application has been developed as a piece of i
standard solution.

For determining information about the standard conformance of the business applications, such as displayed in Figure 5.5, the derived attributes *standardConform* and *exceptionAllowed* are used. The values of these attributes are derived by expressions similar to the following[1]:

$$standardConform = \begin{cases} null & \text{for } (realizedSolution = null) \vee \\ & \quad (allowedSolutions = null) \\ true & \text{for } realizedSolution \in allowedSolutions \\ false & \text{for } realizedSolution \notin allowedSolutions \end{cases}$$

respectively

$$exceptionAllowed = \begin{cases} null & \text{for } allowedSolutions = null \\ true & \text{for } NoArchitecturalSolution \in \\ & \quad allowedSolutions \\ false & \text{for } NoArchitecturalSolution \notin \\ & \quad allowedSolutions \end{cases}$$

In deriving these values, the result *null* is used to indicat
mation no valid statements on the respective property ca

6.3.5 Implementation

This I-Pattern should be implemented in some kind of dat
consistency for the derived attributes *standardConform* a

6.3.6 Variants

A possible variant for this I-Pattern would be to simple
application, indicating if the business application under co
architectural standards or not. It is not advised to use th
the possible analyzes. The advantage is that the amount
collected is limited.

6.3.7 Known Uses

The following uses of this I-Pattern are known:

- *Enterprise Architecture Management Tool Survey 2*

An equivalent information model fragment is included in t

- SoCaTool (sebis)

6.3.8 Consequence Section

A liability of this I-Pattern is the amount of data that has to be collected to be able to reasonable analyze the data. Especially the information of conformance to an architectural solution can only be answered by the business application owner. Therefore, every business application owner has to be interviewed, resulting in a certain investment.
A benefit of this V-Pattern is that an explicit distinction between "there is no information about an architectural solution" and "there is an exception from an architectural solution" is possible.
The data collection effort per year for information about *the conformity of business applications to architectural solutions, reasons for non-conformity, etc.* has been stated by practitioners using such an approach as:

6.3.9 See Also

I-Pattern *Architectural Solution Conformance* is closely related to defining and documenting architectural solutions. This is addressed by I-Pattern *Architectural Solution (I-66)* (see page 223) in [BELM08] and M-Pattern *Standard Conformity Management* (see page 12).
This I-Pattern can be used to manage information for V-Pattern *Standard Conformity Exceptions* (see page 32).

Workshop as part of Software Engineering 2009
Patterns in Enterprise Architecture Management

Kaiserslautern, March 2-6 2009,
http://www.se2009.de/

The workshop addresses
- researchers in software engineering and information system
- IT managers, enterprise architects, software architects

We seek contributions in the following areas (non-exclusive)
- specific EAM patterns derived from case studies and research projects
- EAM patterns on a metamodel level and model level
- organization of pattern catalogs
- usage of EAM patterns in industry or in education
- empirical studies about pattern adoption

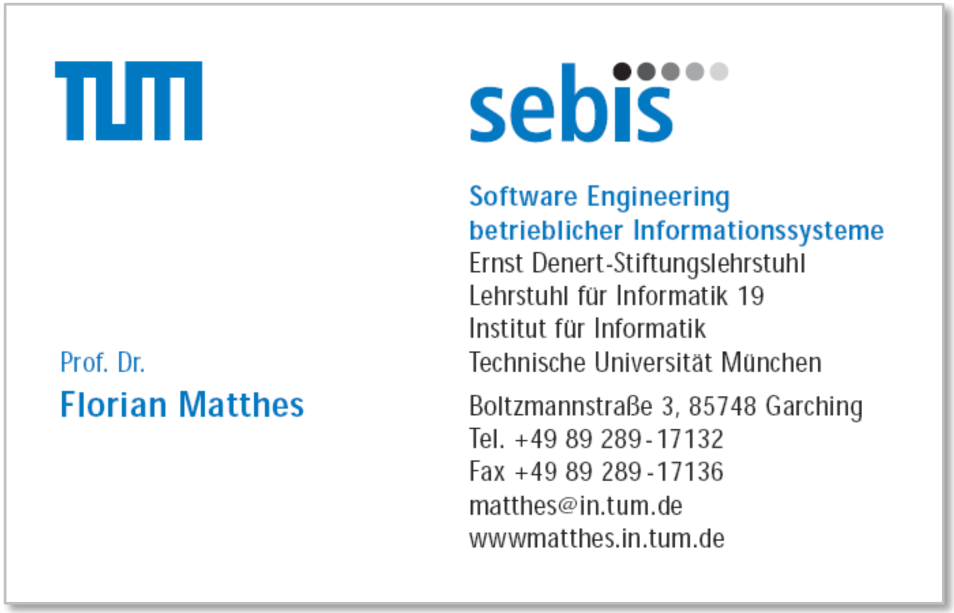
Program Committee Members
- Prof. Dr. Dr. h.c. Hans-Jürgen Appelrath (Universität Oldenburg, Germany)
- Prof. Dr. Gregor Engels (Universität Paderborn, Germany)
- Prof. Dr. Ulrich Frank (Universität Duisburg Essen, Germany)
- Dr. Pontus Johnson (KTH Stockholm, Sweden)
- Prof. Dr. Dimitris Karagiannis (Universität Wien, Austria)
- Wolfgang Keller (objectarchitects, Germany)
- Dr. Marc Lankhorst (Telematica Instituut, Netherlands)
- Prof. Dr. Florian Matthes (Technische Universität München, Germany)
- Prof. Dr. Ralph Reussner (Universität Karlsruhe, Germany)
- Dr. Michael Rohloff (Universität Potsdam, Germany)
- Prof. Peter Sommerlad (Hochschule für Technik Rapperswil, Switzerland)
- Dr. Ulrike Steffens (OFFIS, Oldenburg)
- Johannes Willkomm (Capgemini sd&m Research, Germany)
- Prof. Dr. Robert Winter (Universität St. Gallen, Switzerland)

http://wwwmatthes.in.tum.de/wikis/sebis/peam-2009

Summary

- EAM is advanced by experienced practitioners in large organizations.
- EAM patterns are a promising approach to capture, disseminate and apply EAM knowledge.
- An EAM pattern language might help to improve the communication
  - within an enterprise
  - within an industry
  - between industry and academia
  - between academic disciplines (engineering & management sciences)

Thank you for your attention!



Prof. Dr.

Florian Matthes

Software Engineering
betrieblicher Informationssysteme
Ernst Denert-Stiftungslehrstuhl
Lehrstuhl für Informatik 19
Institut für Informatik
Technische Universität München

Boltzmannstraße 3, 85748 Garching
Tel. +49 89 289-17132
Fax +49 89 289-17136
matthes@in.tum.de
wwwmatthes.in.tum.de

More information: www.systemcartography.info