Useful EAM-Standards and Best-Practice Frameworks

29.06.2016, Prof. Dr. Florian Matthes
1. Useful EAM-Standards and Best-Practice Frameworks
Enterprise architecture frameworks have a long history.
Enterprise architecture frameworks have a long history.

- Several frameworks for the Enterprise Architecture (EA Frameworks) have been developed over time
- Their level of detail differs strongly
  - Zachmann - “1” page
  - TOGAF (Version 9 "Enterprise Edition") - “700+” pages
- Generalized Enterprise Reference Architecture and Methodology (GERAM)
  - ISO Norm 15704
  - Guidelines for creating frameworks
  - (As of today) no well-accepted reference
- DoDAF (Department of Defense) and NAF (Nato Architecture Framework) are binding for IT in the military domain
# The Zachman Framework for Enterprise Architecture

## Enterprise Architecture - A Framework

<table>
<thead>
<tr>
<th>LEVEL OF DETAIL</th>
<th>SCOPE (CONTEXTUAL)</th>
<th>FUNCTION</th>
<th>NETWORK</th>
<th>PEOPLE</th>
<th>TIME</th>
<th>MOTIVATION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System Model (Logical)</strong></td>
<td>Data: Logical Data Model, Role: Data Manager</td>
<td>Function: Logical Data Model</td>
<td>Network: Logical Data Model</td>
<td>People: Users</td>
<td>Time: System Deployment</td>
<td>Motivation: System Administration</td>
</tr>
</tbody>
</table>

Zachman Institute for Framework Advancement - (810) 231-0531

© sebis
### Zachman: Different models depending on the stakeholder

<table>
<thead>
<tr>
<th><strong>Bubble charts</strong></th>
<th><strong>Contractor‘s plans</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic concepts for building</td>
<td>Final building as seen by the builder</td>
</tr>
<tr>
<td>Gross sizing, shape, spatial relationships</td>
<td>Architect’s plans constrained by laws of nature and available technology</td>
</tr>
<tr>
<td>Architect/owner mutual understanding</td>
<td>„How to build it“ description</td>
</tr>
<tr>
<td>Initiate project</td>
<td>Directs construction activities</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Architect‘s drawings</strong></th>
<th><strong>Shop plans</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Final building as seen by the owner</td>
<td>Subcontractor’s design of a part/section</td>
</tr>
<tr>
<td>Floor plans, cutaways, pictures</td>
<td>Detailed stand-alone model</td>
</tr>
<tr>
<td>Architect/owner agreement on building</td>
<td>Specification of what is to be constructed</td>
</tr>
<tr>
<td>Establish contract</td>
<td>Pattern</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Architect‘s plans</strong></th>
<th><strong>Building</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Final building as seen by the designer</td>
<td>Physical building</td>
</tr>
<tr>
<td>Translation of owner‘s view into a product</td>
<td></td>
</tr>
<tr>
<td>Detailed drawings – 16 categories</td>
<td></td>
</tr>
<tr>
<td>Basis for negotiation with general contractor</td>
<td></td>
</tr>
</tbody>
</table>

---

Zachman: Framework 1987

- **5 Levels**
  - Scope description (ballpark view)
  - Model of the business (owner’s view)
  - Model of the information system (designer’s view)
  - Technology model (builder’s view)
  - Detailed description (out-of-context view)

- **3 perspectives**
  - Data description
  - Process description
  - Network description

- Zachman Framework started in 1987
  - as „A framework for information systems architecture“!
  - with 5 levels and 3 perspectives

- In 1992 Zachman and Sowa
  - extended the framework with 3 new perspectives
    - Persons (Who?)
    - Time (When?)
    - Motivation (Why?)
  - Added a meta-model for the owner’s, designer’s und builder’s level
  - Defined 7 rules for the concretion of the framework

The basic structure of CapGemini can be divided into two dimensions

- **Architecture aspects**: Different architectures of an enterprise
- **Architecture layers**: context, conceptual, logical and physical layer of each architecture aspect

Quasar Enterprise: Macro-structure of the Integrated Architecture Framework (IAF) (2)

- **Business architecture** – Structures the business processes and business services in order to match the business goals and to model the organization of the enterprise

- **Information architecture** – Structures the information required in the business architecture

- **Information systems architecture** – Structures the application landscape from a business perspective

- **Technology infrastructure architecture** – Structures the used technical platforms and system software components

Creation of a regulation framework (1)

- Creation of a regulating framework for questions, which should be addressed in the context of an enterprise architecture
- Everything starts with a clear separation between business and IT

Creation of a regulation framework (2)

- Afterward it is important to distinguish between requirements and implementation

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Business strategy</th>
<th>IT strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation</td>
<td>Business architecture (Business process, Business services, Business objects, organizations, etc)</td>
<td>Architecture of the application landscape</td>
</tr>
</tbody>
</table>

Creation of a regulation framework (3)

- Business strategy, quality criteria and business architecture are driving the design of the application landscape

- Creation of an unique view on the business architecture. On the part of the IT, the IAF architecture aspects and -layers are respected.

TOGAF 9: Scope & goals

Scope
TOGAF emphasizes business goals as architecture drivers, and provides a repository of best practices, including:

- TOGAF Architecture Development Method (ADM)
- ADM Guidelines & Techniques
- TOGAF Architecture Content Framework
- Enterprise Continuum
- TOGAF Reference Models
- TOGAF Capability Framework

Long-term goals
- An industry standard, generic enterprise architecture method….
- ….usable on its own or in conjunction with frameworks having products relevant/specific to particular sectors.
  - Several frameworks are directly referenced: Zachman, Spewak, DoD Framework, FEAF, TEAF, …
  - Almost complete focus on artefacts, not method
  - TOGAF and…. (not TOGAF or….)

© sebis
TOGAF as a Business Transformation Framework

- **Business strategy**
  - **Design options**
  - **Business model & capabilities**

**Capability**
- **Planning, Monitoring, Controlling**
- **Capability Increment**

**People Dimension**
- Individual Training
- Collective Training
- Professional Development

**Process Dimension**
- Concepts
- Business Processes
- Information Management

**Material Dimension**
- Infrastructure
- Information Technology
- Equipment
The current version 9.1 of TOGAF provides a basis for developing an organization-specific Business Transformation Framework.

Needs of the business shape non-architectural aspects of business operation

The Architecture Capability operates a method

The method produces content to be stored in the Repository, classified according to the Enterprise Continuum

The method delivers new business solutions

Operational changes update the Enterprise Continuum and Repository

The Enterprise Continuum and Repository inform the business of current state

Learning from business operation creates new business need

Sets targets, KPIs, plans and budgets for architecture roles

Business Capability drives the need for Architecture Capability Maturity

Effective operations of the Architecture Capability ensures realization of the Business Vision

Informs the size, structure and culture of the capability

Business need feed into the method, identifying problems to be addressed

The method refines understanding of business need

The method produces content to be stored in the Repository, classified according to the Enterprise Continuum

The Enterprise Continuum and Repository inform the business of current state

TOGAF is a trademark of The Open Group
The structure of TOGAF 9.1 is organized along the structure and content of an Enterprise Architecture Capability.

**Part I: Introduction**
Introduction of the **core concepts** of enterprise architecture and in particular the TOGAF approach. Includes **definitions** and release notes with essential changes to previous TOGAF versions.

**Part II: Architecture Development Method (ADM)**
The core of TOGAF describes the TOGAF Architecture Development Method (ADM) – a phase-oriented approach for **developing an enterprise architecture**.

**Part III: ADM Guidelines and Techniques**
This **collection** includes several **manuals and methods**, supporting the implementation of TOGAF and the TOGAF ADM.

**Part IV: Architecture Content Framework**
This part describes the **TOGAF Content Framework**. It includes a structured **meta-model** for architecture artefacts, re-usable architecture building blocks and typical deliverables of enterprise architecting.
The structure of TOGAF 9.1 is organized along the structure and content of an Enterprise Architecture Capability.

This part includes necessary taxonomy and tools usable for classifying and storing enterprise architecting results within organizations.

A selection of reference models, i.e., the TOGAF Technical Reference Model (TRM), and the Integrated Information Infrastructure Reference Model (III-RM).

For implementing and operating the architecture function of an enterprise necessary organization, processes, skills, roles and responsibilities.
TOGAF Architecture Development Method (ADM)

- An iterative method, over the whole process, between phases and within phases
- Each iteration = new decisions:
  - Enterprise coverage
  - Level of detail
  - Time horizon
  - Architecture asset re-use:
    - previous ADM iterations
    - other frameworks, system models, industry models,…
- Decisions based on:
  - Competence / resource availability
  - Value accruing to the enterprise.

Remark: Every phase is validated against and validates the current requirements of the business. Iteration is possible.
This phase prepares the organization for undertaking successful EA projects:
- Understand business environment
- High level management commitment
- Agreement on scope
- Establish principles
- Establish governance structure
- Agree on method to be adopted
Phase A – Architecture Vision

- Initiates one iteration of the architecture process
  - Sets scope, constraints, expectations
  - Required at the start of every architecture cycle
- Creates the Architecture Vision
- Validates business context
- Creates Statement of Architecture work
Phase B – Business Architecture

- Describe current business architecture
- Develop target business architecture
- Perform gap analysis
- Define roadmap for transforming the business architecture
- Select and adapt relevant architecture viewpoints
- Create architecture definition document
Phase C – Information Systems Architecture

- This phase is detailed in data architecture and application architecture
  - Describe current data/application architecture
  - Develop target data/application architecture
  - Perform gap analysis
  - Define roadmap for transforming the data/application architecture
  - Select and adapt relevant architecture viewpoints
- Create architecture definition document
Phase D – Technology Architectures

- Describe current technology architecture
- Develop target technology architecture
- Perform gap analysis
- Define roadmap for transforming the technology architecture
- Select and adapt relevant architecture viewpoints
- Create architecture definition document
Phase E – Opportunities and Solutions

- Analyze existing culture of the enterprise
- Consolidate gaps identified in phases B to D
- Perform initial implementation planning (including dependencies)
- Identify the major implementation projects
- Group projects into Transition Architectures
- Decide on approach
  - Make v Buy v Re-Use
  - Outsource
  - COTS
  - Open Source
- Assess priorities
Phase F – Migration Planning

- For projects identified in Phase E perform
  - Cost/benefit analysis
  - Risk assessment
- Develop a detailed Implementation and Migration Plan (roadmap)
Phase G – Implementation Governance

- Provide architectural oversight for the implementation.
- Defines architecture constraints on implementation projects
- Architecture contract
- Monitors implementation work for conformance
- Realize EA compliance reviews
- Produce a Business Value Realization.
Phase H – Architecture Change Management

- Provide a continual monitoring and a change management process
- Ensures that changes to the architecture are managed in a cohesive and architected way
- Establishes and supports the EA to provide flexibility to evolve rapidly in response to changes in the technology or business environment
- Monitors the business and capacity management.
- Management of the governance structures (quality gates)
The TOGAF Architecture Content Framework provides an overview of possible information model elements.
## TOGAF bottom line

<table>
<thead>
<tr>
<th>Pro</th>
<th>Contra</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ Pervasive in practice</td>
<td>▪ Inconsistencies between parts</td>
</tr>
<tr>
<td>▪ Trainings available</td>
<td>▪ Adaptation necessary</td>
</tr>
<tr>
<td>▪ Certificates available</td>
<td>▪ Learning TOGAF ≠ mastering TOGAF</td>
</tr>
<tr>
<td>▪ Compliant tools available</td>
<td>▪ Lack of concrete guidelines for introducing TOGAF</td>
</tr>
<tr>
<td>▪ Internationally accepted</td>
<td>▪ Lean EAM countermovement</td>
</tr>
<tr>
<td>▪ Open development</td>
<td></td>
</tr>
</tbody>
</table>
ArchiMate’s historical origin

- Research project between 2002 and 2004
- Funded by 4 million €

**Motivation**
- Increasing need for **precise documentation** on the enterprise architecture level
- **Communicating** about architecture with others
- **Analysis** of architectures before their implementation

- Since 2009, ArchiMate is an Open Group standard and complements TOGAF
ArchiMate’s layers and aspects

- Information
- Behavior
- Structure

Layers:
- Business
- Application
- Technology

Aspects:
- Objects
- Verbs
- Subjects

Enterprise Architecture

Domains
ArchiMate’s layers and aspects (in detail)
Akin concepts at all layers facilitate learning the language and make its use more consistent:
ArchiMate’s core concepts

[Diagram showing core concepts of ArchiMate: Business, Application, Technology, Information, Behaviour, Structure.]
ArchiMate’s layered architecture (example)

Business Layer

Application Layer

Technology Layer

[Diagram of layered architecture with components labeled for each layer: Business Layer includes Insurant, Client, Claim registration service, Customer information service, Claims payment service, Handle Claim business process (register, accept and validate, pay), Insurer, ArchSurance. Application Layer includes Customer administration service, Claims administration service, Payment service, CRM system, Customer information service, Policy administration, Claim information service, Financial application. Technology Layer includes Claim files service, Customer files service, zSeries mainframe, DB 2 database, Sun Blade, iPlanet app server, Financial application EJBs.]
ArchiMate’s integration with TOGAF

TOGAF

Process

Viewpoints

Language

Repository, Reference Models

ArchiMate

[9]
ArchiMate’s extensions mapped to TOGAF

ArchiMate’s core supports these phases:
- B, C, D

Implementation & migration extension:
- E, F, G

Motivation extension
- H, Preliminary, Requirements Management
ArchiMate’s motivation extension

Different kinds of relationships:
- Association relation
- Realization relation
- Aggregation relation
- Specialization relation
- Contribution relation (+/-)
- Conflict relation
ArchiMate’s migration and implementation extension

Different kinds of relationships:
- Specialization relation
- Composition relation
- Aggregation relation
- Assignment relation
- Triggering relation
- Access relation
- Association relation
Relationship between the ArchiMate core and its extensions
Iteraplan “best practice” enterprise architecture modeling concepts and their relationships
Recently introduced concepts

Using domain-specific modeling languages

<table>
<thead>
<tr>
<th>Pro</th>
<th>Contra</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Predefined meta-model</td>
<td>- Maybe not prevalent in a specific region (e.g. Germany)</td>
</tr>
<tr>
<td>- Specific notation</td>
<td>- Training necessary for every modeler</td>
</tr>
<tr>
<td>- Aligned with other frameworks (e.g. TOGAF)</td>
<td>- Limited extensibility</td>
</tr>
<tr>
<td>- Easier start of modeling due to existing concepts</td>
<td>- Terminology mapping required</td>
</tr>
<tr>
<td>- Comprehensive documentation</td>
<td></td>
</tr>
<tr>
<td>- Publicly available</td>
<td></td>
</tr>
</tbody>
</table>
Discussion

1. Which EA frameworks are used in your company?
2. What are expected and realized benefits?
Thank you for your attention. Questions?