Enterprise Architecture Frameworks

Prof. Dr. Knut Hinkelmann
Learning Objective of Chapter 2

- **Topic: Enterprise Architecture Framework**
  - Content and structure of enterprise architecture descriptions

- **This is necessary because**
  - Enterprises are complex systems and thus also enterprise architecture descriptions are complex
  - Frameworks provide a structure for EA descriptions

- **Learning Objective**
  - Understand the content of an enterprise architecture description and how it can be organised
  - know the two main enterprise architecture frameworks TOGAF and Zachman
ISO/IEC/IEEE 42010 Systems and Software Engineering — Architecture Description

- International standard for architecture descriptions of systems and software.
- The original IEEE 1471 specified requirements on the contents of architecture descriptions of systems.
  - An architecture description (AD) expresses the architecture of a system of interest
- ISO/IEC/IEEE 42010 adds definitions and requirements on architecture frameworks and architecture description languages (ADLs)

http://www.iso-architecture.org/ieee-1471/faq.html
Architecture Description and Architecture Models

- An *Architecture Description* consists of one or several *Architecture Models*.

- A Model is a reproduction of a *relevant* part of reality which contains the essential aspects to be investigated.

- Relevance depends on stakeholders and their concerns.
Stakeholder and Concerns

- **Stakeholders** are individuals, groups or organizations holding concerns for the System, i.e.
  - Examples of Stakeholders: client, owner, user, operator, maintainer, developers, suppliers, regulator, auditor, architect.

- A **Concern** is any interest in the system, i.e. the objective for which a model is used
  - Examples of Concerns: optimisation, efficiency, quality of service, automation, agility, behavior, business goals, customer experience, flexibility, maintainability, regulatory compliance, security.
Architecture Views and Viewpoints

Not everyone is interested in everything.

Views and Viewpoints are a means to specify which part of an Architecture Description is of relevance

♦ **View**: Part of an architecture description, which is relevant for
  ● one or more *Stakeholders*
  ● to address specific *Concerns*

♦ **Viewpoint** specifies a view
  ● a characterisation of stakeholders and their concerns
  ● the concepts, relationships, models, and visualizations that are provided by the view

A view is what you see and a viewpoint is where you are looking from

Source: ArchiMate 2.0 Specification, chapter 8, http://pubs.opengroup.org/architecture/archimate2-doc/chap08.html
Architecture Models and Model Kinds

■ An **Architecture View** consists of one or more **Architecture Models**
  ♦ Examples of Models:
    ● The model of the order process of the company,
    ● the organisation structure of a specific company
    ● the model of the customer data,

■ A **Model Kind** defines the concepts and relations needed to model Architecture Views governing **Architecture Viewpoint**.
  ♦ Examples of Model Kinds:
    ● process models
    ● organisation models
    ● data models
An **Architecture Framework** establishes a common practice for creating, interpreting, analyzing and using architecture descriptions (Views and Viewpoints) within a particular domain of application or stakeholder community.

http://www.iso-architecture.org/ieee-1471/cm/
Timeline of Enterprise Architecture Frameworks

- 1996: C4ISR Architecture Framework (CAF) v2.0
- 1998: C4ISR Architecture Framework (CAF) v1.0
- 1999: DoD retires TAFIM
- 2000: The Open Group takes over TAFIM
- 2002: DoD Architecture Framework, 1.0 (DoDAF)
- 2003: Zachman 2003
- 2004: Gartner buys Meta
- 2005: TOGAF Certification Program
- 2006: TOGAF 8.1
- 2007: TOGAF 8.0 enterprise edition
- 2008: TOGAF 8.1.1
- 2009: DoDAF 1.5
- 2010: DoDAF 2.0

*(Bespoke Systems 2012)*

Prof. Dr. Knut Hinkelmann
Enterprise Architecture Frameworks

- There are a number of Enterprise Architecture Frameworks
- We can distinguish two main types of structures:
  - **Matrix** of aspects and perspectives, e.g.
    - **Zachmann Enterprise Architecture Framework**
      - An enterprise ontology
  - **Three layer architecture** with business, applications and technology, e.g.
    - **TOGAF - The Open Group Architecture Framework**
      - A methodology for architecture development
    - **ArchiMate – A modeling language for EA**
    - **Best Practice Enterprise Architecture**
The Zachman Framework
Zachman Framework

- Regarded the origin of enterprise architecture frameworks (originally called "Framework for Information Systems Architecture")
- First version published in 1987 by John Zachman
- It is still further developed by Zachman International (http://www.zachman.com)
- Often referenced as a standard approach for expressing the basic elements of enterprise architecture
- The framework is a logical structure for classifying and organising the descriptive representations of an enterprise

Rationale of the Zachman Architecture

- There is not a single descriptive representation for a complex object ... there is a SET of descriptive representations.

- Descriptive representations (of anything) typically include:
  - Perspectives
  - Abstractions
Dimension 1 – Perspectives

Zachman originally used the analogy of classical architecture
For the different stakeholders different aspects of a building are relevant - models of the building from different perspectives

- **Bubble charts**: conceptual representation delivered by the architect
- **Architect's drawing**: transcription of the owner's perceptual requirements – *owner's perspective*
- **Architect's plans**: translation of the owner's requirements into a product – *designer's perspective*
- **Contractor's plans**: phases of operation, architect's plans conained by nature and technology – *builder's perspective*
- **Shop plans**: parts/sections/components of building details (out-of-context specification) – *subcontractor's perspective*
- **The building**: physical building itself

(Zachman 1987)
Dimension 1: Architectural Representations with analogies in Building and Information Systems

<table>
<thead>
<tr>
<th>Generic</th>
<th>Buildings</th>
<th>Information Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ballpark</td>
<td>Bubble charts</td>
<td>Scope/objectives</td>
</tr>
<tr>
<td>Owner's representation</td>
<td>Architect's drawings</td>
<td>Model of the business (or business description)</td>
</tr>
<tr>
<td>Designer's representation</td>
<td>Architect's plans</td>
<td>Model of the information system (or information system description)</td>
</tr>
<tr>
<td>Builder's representation</td>
<td>Contractor's plans</td>
<td>Technology model (or technology-constrained description)</td>
</tr>
<tr>
<td>Out-of-context representation</td>
<td>Shop plans</td>
<td>Detailed description</td>
</tr>
<tr>
<td>Machine language representation</td>
<td>—</td>
<td>Machine language description (or object code)</td>
</tr>
<tr>
<td>Product</td>
<td>Building</td>
<td>Information system</td>
</tr>
</tbody>
</table>

(Zachman 1987)
### Perspectives

<table>
<thead>
<tr>
<th>Each row is different in nature, in content, in semantics from the others – representing different perspectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Representations do not correspond to different levels of details – level of detail is an independent variable, varying within one representation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scope (Boundaries)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements (Concepts)</td>
</tr>
<tr>
<td>Design (Logic)</td>
</tr>
<tr>
<td>Plan (Physics)</td>
</tr>
<tr>
<td>Part (Configurations)</td>
</tr>
<tr>
<td>Product (Instances)</td>
</tr>
</tbody>
</table>

- The content of these cells defines the scope of the enterprise, identifying what should possibly be modeled.
- These cell models comprise the Business Model - the Owner's expectations from a business perspective for the operating enterprise.
- These cell models comprise the technology neutral System Model - the Designer's plan for enabling the Business Model.
- These cell models comprise the Technology Model - the Builder's plan for applying technology to the System Model.
- These cells are listings, identifying the actual solutions that have been implemented.
- The functioning enterprise.
Dimension 2: Aspects of an Architecture

- There exist different types of descriptions oriented to different aspects
- Zachman associates each aspect with a question word
  - WHAT     inventory models
  - HOW      functional/process models
  - WHERE    location/distribution models
  - WHO      organisation models
  - WHEN     timing models
  - WHY      motivation models

(Zachman 1987)
Abstractions for Manufacturing

(Zachman 2012)
The Zachman Framework for Enterprise Architecture – Enterprise Ontology

Abstractions/Aspects

Each cell contains models

Prof. Dr. Knut Hinkelmann
Strategic Alignment Model and Zachman Framework

Organizational Infrastructure and Processes

IT Strategy

Business Strategy

I/S Infrastructure and Processes

Functional Integration

Strategic Fit

Business Information Technology

Enterprise Architecture Frameworks

Prof. Dr. Knut Hinkelmann
Model Types in Zachmann

- There are different model kinds for each viewpoint (one model kind per cell)
- There can be different modeling languages to represent a kind of model
- The Architecture Description language consists of the different model kinds used

Data model
- UML class
- ERM

Organisation model
- org chart

Motivation Model
- BMM

Process model
- Flow diagram
- BPMN
- Petri Net

Workflow model
- BPEL

IT model
- IT systems
Relations between Models and Model Elements

- There are relations between (elements of) the models

**Horizontal Relations**: In same perspective, e.g.
- Data used in a process
- Application implementing a process activity

**Vertical relations**: Between different perspectives
- Implementation of an application
- Database model for an entity relationship model
Enterprise Architecture Modeling – Examples of Models Kinds

Process Model

Organisation Model

Data/Documents

Fact Type Model

Business Motivation

UML class diagram

UML activity diagram

UML sequence diagram

UML component diagram
TOGAF – The Open Group Architecture Framework
TOGAF – The Open Group Architecture Framework

■ Developed and continuously evolved since the mid-90’s by The Open Group’s Architecture Forum

■ While Zachman is more an ontology, TOGAF is a methodology

■ At the heart of the framework is the Architecture Development Method (ADM)

■ http://www.opengroup.org/togaf/
Structure of the TOGAF Document

(The Open Group 2009, p. 4)
The TOGAF enterprise architecture model is organised in four partial sub-architectures:

♦ **Business Architecture**
  ● Strategies, governance, organisation and business processes of the enterprise

♦ **Information Systems Architecture** – consists of
  ● **Data Architecture**
    – data and their relations as well as principles for the organisation and the management of resources
  ● **Application Architecture**
    – information systems and their relations to business processes

♦ **Technology Architecture**
  ● current technical realisation and future enterprise-specific standards like operating system, middleware, infrastructure
TOGAF Architecture Development Method (ADM)

- TOGAF addresses the whole enterprise architecture lifecycle
- The TOGAF Architecture Development Method (ADM) is a generic method for developing an enterprise architecture
- The goals, approaches, required input, activities and deliverables are documented for each phase separately
- The ADM method is enriched by specific ADM guidelines and techniques.

(The Open Group 2009)
TOGAF Architecture Views

Business Architecture
Data Architecture
Application Architecture
Technology Architecture
TOGAF Architecture Development Method (ADM)

- Although originally represented as a sequential method, chapter 19.2 of TOGAF describes also iteration cycles.
TOGAG Content Metamodel

Architecture Principles, Vision, and Requirements

<table>
<thead>
<tr>
<th>Preliminary</th>
<th>Architecture Vision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture Principles</td>
<td>Business Strategy</td>
</tr>
<tr>
<td></td>
<td>Technology Strategy</td>
</tr>
<tr>
<td></td>
<td>Business Principles, Objectives, and Drivers</td>
</tr>
<tr>
<td></td>
<td>Architecture Vision</td>
</tr>
<tr>
<td></td>
<td>Stakeholders</td>
</tr>
</tbody>
</table>

Architecture Requirements

| Requirements | Constraints | Assumptions | Gaps |

Business Architecture

Motivation

Drivers | Goals | Objectives | Measures

Organization

Organization Units | Locations | Actors, Roles

Function

Business Services, Contracts, Service Qualities | Processes, Events, Controls, Products | Functions

Information Systems Architecture

Data

Data Entities

Logical Data Components

Physical Data Components

Application

Information System Services

Logical Application Components

Physical Application Components

Technology Architecture

Platform Services

Logical Technology Components

Physical Technology Components

Architecture Realization

Opportunities, Solutions, and Migration Planning

Capabilities | Work Packages | Architecture Contracts

Implementation Governance

Standards | Guidelines | Specifications

TOGAF Content Metamodel

The content metamodel provides a definition of all the types of building blocks that may exist within an architecture.

The content metamodel

- identifies all of these building block (i.e., application, data entity, technology, actor, and business service),
- shows the relationships that are possible between them, e.g.
  - actors consume business services
  - data entities are held within applications
  - technologies implement applications
  - applications support business users or actors
- identifies artifacts that can be used to represent them.

The architecture content framework “provides a structural model for architectural content” and may also be substituted with other frameworks, such as the Zachman Framework (The Open Group, 2009, p. 361).
TOGAF: Architecture Content Framework

• The content framework is intended to allow TOGAF to be used as a stand-alone framework for architecture.

• However, some enterprises may opt to use an external framework (such as the Zachman Framework or ArchiMate) in conjunction with TOGAF.

• In these cases, the content framework provides a useful reference and starting point for TOGAF content to be mapped to other frameworks.

(The Open Group 2011, Part IV)
ArchiMate
Archimate

- Archimate is a modeling language that supports the TOGAF content metamodel and the TOGAF ADM

- Three architecture layers:
  - Business
  - Application
  - Technology

- Three main types of elements:
  - *Active structure* element: an entity that is capable of performing behavior.
  - *Behavior* element: a unit of activity performed by one or more active structure elements.
  - *Passive structure* element: an object on which behavior is performed.
Archimate and TOGAF

TOGAF ADM
Cross-Layer Dependencies: Business-IT Alignment

Extensions of ArchiMate to cover the whole TOGAF ADM
Best Practice Enterprise Architecture
Best Practice Enterprise Architecture

- The Best Practice Architecture from Inge Hanschke (2010) is another example of a three-layer enterprise architecture framework.

- In contrast to TOGAF
  - it is quite simple
  - it differentiates between the technical architecture and the infrastructure architecture
  - it does not have a separate data or information architecture

from (Hanschke, 2010)
Partial Architectures of the Best Practice Architecture

- **Business Architecture**
  - Describing main entities that determine the business: business processes, functions, products, business units and business objects.

- **Application Architecture**
  - Documentation of the information systems landscape, i.e. information systems, their data and interfaces and the information flow
  - Bridge between business architecture and the architectures of technology and infrastructure

- **Technology Architecture**
  - Determination of enterprise-specific technical standards for information systems, interfaces and infrastructure

- **Infrastructure Architecture**
  - Entities of the infrastructure, on which the information systems are running
Enterprise Architecture Modeling

Overview: ArchiMate

Detailed Models

Overview: Zachman

- Business Motivation
- Organisation Model
- Business Rules
- Fact Type Model

- UML class diagram
- UML sequence diagram
- UML component diagram
- UML activity diagram