Enterprise Architecture Frameworks

Prof. Dr. Knut Hinkelmann
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 and Architecture Description

- An architecture is a *conception of a system* – i.e., it is in the human mind. An architecture may exist without ever being written down.

- An *architecture description* (AD) is an artifact that expresses an Architecture to share with others.
  - An AD is what is written down as a concrete work product. It could be a document, a repository or a collection of artifacts used to define and document an architecture.
  - Architects and other system stakeholders use Architecture Descriptions to understand, analyze and compare Architectures, and often as "blueprints" for planning and construction.

http://www.iso-architecture.org/ieee-1471/cm/
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.
Architecture Models and Model Kinds

- An Architecture Model represents a specific part of an Architecture, i.e. a View on the Architecture
  - 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.
  - Examples of Model Kinds:
    - process models
    - organisation models
    - data models

http://www.iso-architecture.org/ieee-1471/cm/
Architecture Views and Viewpoints

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 viewpoint determines the concepts, relationships, models, and visualizations that should be 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
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.

It is a structure for the content of an Enterprise Architecture Description.

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

1987
- A framework for architecture, John Zachman article in IBM Systems Journal, Vol 26, No 3

1992
- TOGAF 1.0

1993
- The Open Group takes over TAFIM
- DoD retires TAFIM

1994
- Capp Gemini Integrated Architecture Framework (IAF)
- C4ISR Architecture Framework (CAF) v1.0

1995
- Clinger-Cohen Act, Information Technology Management Reform Act of 1996

1996
- Federal CIO Council introduces Federal Enterprise Architecture Framework (FEAF)

1997
- A Practical Guide to Federal Enterprise Architecture (Version 1.0), CIO Council

1998
- GAO report to congress 'Enterprise Architecture Use across the Federal Government can be Improved'

2000
- OMB takes over stewardship of FEAF, renames it FEA

2001
- FEA mostly complete

2002
- Zachman 2003

2003
- Gartner buys Meta

2004
- TOGAF Certification Program

2005
- TOGAF 8.1

2006
- TOGAF 8.0 technical edition

2007
- TOGAF 8.0 enterprise edition

2008
- TOGAF 8.1.1

2009
- TOGAF 9

2010
- DoDAF 1.5

(Bespoke Systems 2012)
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

(Zachman 2012)
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 constrained 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

■ Each row is different in nature, in content, in semantics from the others – representing different perspectives

■ Representations do not correspond to different levels of details – level of detail is an independent variable, varying within one representation
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

<table>
<thead>
<tr>
<th>Perspectives</th>
<th>Information Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business</td>
<td>Technology</td>
</tr>
<tr>
<td>Executive</td>
<td>Engineer</td>
</tr>
<tr>
<td>Business Mgmt</td>
<td>Technician</td>
</tr>
<tr>
<td>Architector</td>
<td>Tool Components</td>
</tr>
<tr>
<td>Enterprise</td>
<td>Enterprise</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classification Names</th>
<th>Abstractions/Aspects</th>
</tr>
</thead>
<tbody>
<tr>
<td>What</td>
<td>How</td>
</tr>
<tr>
<td>Where</td>
<td>Who</td>
</tr>
<tr>
<td>When</td>
<td>Why</td>
</tr>
</tbody>
</table>

Each cell contains models.
The Zachman Framework is not a Methodology

**Ontology**

The Zachman Framework™ schema technically is an ontology -

a theory of the existence of a structured set

of essential components of an object

(the object being an Enterprise, a department, a value chain,

a "sliver," a solution, a project,

an airplane, a building, a bathtub or whatever or whatever).

A Framework is a STRUCTURE.
(A Structure DEFINES something.)

**Methodology**

A Methodology is a PROCESS.
(A Process TRANSFORMS something.)

A Structure IS NOT A Process
A Process IS NOT a Structure.

© 1990-2011 John A. Zachman, Zachman International®
**Ontology vs Methodology**

An Ontology is the classification of the total set of “**Primitive**” (elemental) components that exist and that are relevant to the existence of an object.

A Methodology produces “**Composite**” (compound) implementations of the Primitives.
Ontology

The Zachman Framework for Enterprise Architecture

The Enterprise Ontology

“Primitives” are Timeless.

This is NOT a Process.

© 2012 John A. Zachman
Analogy: Chemistry

**ONTOLOGY**

**PERIODIC TABLE OF THE ELEMENTS**

*Elements are Timeless*

*Until an ontology exists, nothing is repeatable, nothing is predictable.*

*There is no DISCIPLINE.*
Analogy: Chemistry

PROCESS
(METHODOLOGY)

Add Bleach to an Alkali and it is transformed into Saltwater.

HCl + NaOH $\rightarrow$ NaCl + H₂O

COMPONDS

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Salt</td>
<td>NaCl</td>
</tr>
<tr>
<td>Aspirin</td>
<td>C₉H₈O₄</td>
</tr>
<tr>
<td>Vicodin</td>
<td>C₁₈H₂₁NO₅</td>
</tr>
<tr>
<td>Naproxen</td>
<td>C₁₄H₁₄O₅</td>
</tr>
<tr>
<td>Ibuprophen</td>
<td>C₁₃H₁₈O₂</td>
</tr>
<tr>
<td>Viagra</td>
<td>C₂₂H₃₀N₆O₄S</td>
</tr>
<tr>
<td>Sulphuric Acid</td>
<td>H₂SO₄</td>
</tr>
<tr>
<td>Water</td>
<td>H₂O</td>
</tr>
</tbody>
</table>

e tc., etc., etc.
Model and the Zachman Framework

- Concepts for modelling are related to cells.
- Models are composites, they can roughly be assigned to cells, if they are composed of elements (concepts) of this cell.
- The elements of models can (roughly) be assigned to cells, but often cover

![Diagram of the Zachman Framework]

- Logical data model
- Organisational model
- Process model
- Motivation model
- Workflow model
Viewpoints in Zachmann

- A viewpoint is characterized by the row, the column and the domain, e.g. the «How» and «Who» of the Sales Processing from the Architect Perspective.
- There can be different model kinds for a viewpoint
- There can be different modeling languages to represent a kind of model

**Data model**
- UML class
- ERM

**Organisation model**
- org chart
- Interaction

**Motivation Model**
- BMM
- ArchiMate Ext.

**Process model**
- Flow diagram
- BPMN
- Petri Net

**Workflow model**
- BPEL

**IT model**
- IT systems
Strategic Alignment Model and Zachman Framework
Relations between Models and Model Elements (Composites)

- 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

UML class diagram

UML activity diagram

UML sequence diagram

UML component diagram

Business Motivation

Enterprise Architecture Frameworks
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/](http://www.opengroup.org/togaf/)
Structure of the TOGAF Document

(The Open Group 2009, p. 4)
TOGAF Architecture Views

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 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
Although originally represented as a sequential method, chapter 19.2 of TOGAF describes also iteration cycles.
TOGAG 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 Open Group 2011, Part IV)
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).

(The Open Group 2009, p. 379)
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

Business

Application

Technology

Passive structure
Behavior
Active structure

Product domain
Information domain
Process domain
Organization domain

Data domain
Application domain

Technical infrastructure domain
Cross-Layer Dependencies: Business-IT Alignment

Extensions of ArchiMate to cover the whole TOGAF ADM

Enterprise Architecture Frameworks

© The Open Group
Best Practice Enterprise Architecture
Best Practice Enterprise Architecture

- The Bast 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
- Data/Documents
- Business Rules
- Fact Type Model
- UML class diagram
- UML sequence diagram
- UML component diagram
- UML activity diagram

Enterprise Architecture Frameworks