Zachman: The Rationale for Enterprise Architecture and the Zachman Framework
Based on the Lecture of John Zachman, October 2015
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
We need Enterprise Architecture because ...
In short, the reasons you need Architecture: COMPLEXITY AND CHANGE
Architecture: Dealing with Complexity and Change

- If the object you want to create or change is simple, and it is not likely to change, then you can do it directly.
- On the other hand, if the object is complex, you can't see it in its entirety at one time and it is likely to change considerably over time, you need a description or model.
- You need a description of the Architecture.
Why Enterprise Architecture Description?

- Manage change of complex system
  - Baseline for complex, interdependent enterprise decisions
  - Communication of decisions to organization stakeholders.
  - *If architecture is not explicit, there is a high risk that the implementation is not what is intended*

- Continuous, coordinated organisation change
  - Continuously update Enterprise Architecture to reflect changes
  - Coordinate change between different projects
Change the Model before you Change the System!
What is reification?

Start presenting to display the poll results on this slide.
Reification

- Reification is the process by which an abstract idea is turned into an explicit object or thing
The Zachman Framework

Abstractions

Version 3.0
To which dimension of the Zachman Framework does Reification correspond?
Dimension 1: Reification
Turning an abstract Idea into something Concrete
Dimension 1 – Perspectives

Zachman originally used the analogy of classical architecture

From the idea to the building

On each level 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
Reification: Turning an Abstract Idea (Strategy) into Something Concrete (Functioning Enterprise)

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.

Representations do not correspond to different levels of details:
- Each row is different in nature, in content, in semantics from the others – representing different perspectives
- Level of detail is an independent variable, varying within one representation
## Dimension 1: Architectural Representations with analogies in Building and Information Systems

<table>
<thead>
<tr>
<th>Generic</th>
<th>Buildings</th>
<th>Airplanes</th>
<th>Information Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ballpark</td>
<td>Bubble charts</td>
<td>Concepts</td>
<td>Scope/objectives</td>
</tr>
<tr>
<td>Owner’s representation</td>
<td>Architect’s drawings</td>
<td>Work breakdown structure</td>
<td>Model of the business (or business description)</td>
</tr>
<tr>
<td>Designer’s representation</td>
<td>Architect’s plans</td>
<td>Engineering design/bill-of-materials</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>Manufacturing engineering design/bill-of-materials</td>
<td>Technology model (or technology-constrained description)</td>
</tr>
<tr>
<td>Out-of-context representation</td>
<td>Shop plans</td>
<td>Assembly/fabrication drawings</td>
<td>Detailed description</td>
</tr>
<tr>
<td>Machine language representation</td>
<td>—</td>
<td>Numerical code programs</td>
<td>Machine language description (or object code)</td>
</tr>
<tr>
<td>Product</td>
<td>Building</td>
<td>Airplane</td>
<td>Information system</td>
</tr>
</tbody>
</table>
What are the six abstractions of the Zachman Framework?
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

- What
- How
- Where
- Who
- When
- Why

Executive Perspective (Business Context Planners)
Business Mgmt Perspective (Business Concept Creators)
Architect Perspective (Business Logic Generators)
Engineer Perspective (Business Physics Builders)
Technician Perspective (Business Component Implementers)
Enterprise Perspective (Users) The Enterprise

Inventory Models
Bills of Material
Process Models
Functional Specs
Distribution Models
Drawings
Responsibility Models
Operating Instructions
Timing Models
Timing Diagrams
Motivation Models
Design Objectives

Version 3.0

Scope Contexts (Scope Identification Lists)
Business Concepts (Business Distinction Models)
System Logic (System Representation Models)
Technology Physics (Technology Specification Models)
Tool Components (Tool Configuration Models)
Operations Instances (Implementations)
What does quality mean in the context of the enterprise (architecture)?
QUALITY
“Producing end results (the product) that meet the requirements as defined by the customer.”

QUALITY IN THE CONTEXT OF THE ENTERPRISE
Producing Implementations (manual and/or automated)
i.e. the ENTERPRISE (Row 6) that are “aligned” with the intentions of Management (Row 2).
QUALITY PROBLEMS

Either:
A. The Requirements at Row 2 were incorrectly transcribed

Or:
B. In the transformation from Row 2 to Row 6, integrity was lost.

Or:
C. Whoever entered the data at Row 6 created errors.
(This is a Management problem, not an Architecture problem.)
Fixing Quality Problems

A. Fix the Process of transcribing the Requirements (Row 2)

And/Or:

B. Fix the Process of transforming the Requirements (Row 2) into Implementation (Row 6)

and Iterate
until Row 6 is aligned with Row 2.

Continuous Process Improvement
Why does an enterprise need information systems people?
Enteprises need Information Systems People

- Information Systems people …
  … bring to the table drafting skills
  … can describe things very precisely
  … can build models, which are unambiguous

- Is this important?
  ♦ Yes! If several people are involved in a change process, they better should be able to look at the models and know precisely without ambiguity what is described
Modeling Skills

- Modeling skills consist of two aspects
  - Learning modeling languages
    - Syntax/Notation
    - Semantics
  - Being able to express reality appropriately
    - Pragmatics
Models and the Zachman Framework

- Concepts for modelling are related to cells.
- Cells shall contain models with concepts from a single abstraction perspective.
What statements about Ontology and Methodology are true?
Ontology

The Zachman Framework™ schema technically is an ontology - a theory of the existence of a structured set of essential components of an object.

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.
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.
Analogy: Chemistry

This is NOT a Process.

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.

\[ \text{HCl} + \text{NaOH} \rightarrow \text{NaCl} + \text{H}_2\text{O} \]

**COMPONENTS**

Salt \( \text{NaCl} \)
Aspirin \( \text{C}_9\text{H}_8\text{O}_4 \)
Vicodin \( \text{C}_{18}\text{H}_{21}\text{NO}_3 \)
Naproxen \( \text{C}_{14}\text{H}_{14}\text{O}_3 \)
Ibuprofen \( \text{C}_{13}\text{H}_{18}\text{O}_2 \)
Viagra \( \text{C}_{22}\text{H}_{30}\text{N}_6\text{O}_4\text{S} \)
Sulphuric Acid \( \text{H}_2\text{SO}_4 \)
Water \( \text{H}_2\text{O} \)

etc., etc., etc.

This is NOT an Ontology.
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
Is the Zachman Framework an Ontology or a Methodology or both?
The Zachman Framework

- The Zachman Framework is depicted as a 6 x 6 “matrix”.
  - Perspectives: Reification
  - Abstractions: Interrogatives

- The matrix constitutes the total set of descriptive representations that are relevant for describing anything, e.g. an enterprise → ONTOLOGY

- The Zachman Framework does not specify a methodology. Even for the Reification no process described, how to do it.
What is more important - Ontology or Methodology?
Ontology and Methodology

It is NOT either Ontology OR Methodology

It IS Ontology AND Methodology

Ontology and Methodologies do not COMPETE they COMPLETE

© 2015 John A. Zachman, Zachman International®
Alchemy - A Practice

This is a Methodology WITHOUT an Ontology

A Process with no ontological structure is ad hoc, fixed and dependent on practitioner skills.

This is NOT a science.

It is ALCHEMY,

a "practice."
THE PERIODIC TABLE METAPHOR

Before Mendeleev published the Periodic table, Alchemist (practitioners) could create compounds based on their experience ... whatever worked. After Mendeleev figured out the Periodic Table, Chemistry became a science. Creating compounds became predictable and repeatable based on the natural laws (Physics) expressed in the Periodic Table. Within 50 years, the Chemists and Physicists (practitioners) were splitting atoms.
Ontology and Methodology in Business Process Management

Ontology

- Primitives for describing a business process
  - Activity
  - Event
  - Flow
  - Role
  - KPI
  - ...  

- Corresponds to elements and attributes of process modeling and their relations

Methodology

- The procedure for managing business processes: design, monitor, improve

- Corresponds to the BPM life cycle
Does a BPMN Model belong to one cell or is it a composite with elements of cells in different columns?
Composite

Application arrived

Application

Prepare eligibility check

Assessment sheet

Candidate eligible?

Reject candidate

Enrol candidate

Decide eligibility

Head of Program

Study Assistant
Primitives

■ How

Application arrived → Prepare eligibility check → Decide eligibility → Candidate eligible?

Candidate eligible? → Reject candidate → Enrol candidate

■ What

Application
Assessment sheet

Head of Program
- Lecturer
- Study Assistant
Examples of Composites (COMPOUNDS)

- Java Programs
- Objects
- BPMN Models
- Business Architecture
- Capabilities
- Applications Data Models
- Security Architecture
- Services
- Technology Architecture
- Missions/Visions
- Agile Code
- Business Processes
- DoDAF Models
- Balanced Scorecard Clouds
- Archimate Artifacts
According to Zachman's "Enterprise Physics": What is the case, if a cell of the Enterprise Ontology is not made explicit?
Why making Enterprise models

Every Cell of the Enterprise Ontology Exists

Any Cell that is not made explicit is *implicit*, which means that you are allowing anyone and everyone to make whatever *assumptions*

Incorrect assumptions are sources of *defects* … and the source of miscommunication and *misunderstanding*

To avoid misunderstanding and miscommunication about the Enterprise, there should be only a *single version of Cells in Rows 1, 2 and 3*
The First Law of Enterprise Ontological Holism

Every Cell of the Enterprise Ontology exists. Any Cell or portion of Cell that is not made explicit is implicit which means that you are allowing anyone and everyone to make whatever assumptions they want to make about the contents and structure of that Cell.

The Second Law of Enterprise Ontological Holism

Correct assumptions about implicit Cell contents and structure save time and money. Incorrect assumptions are sources of defects … and the source of miscommunication and misunderstanding - conflicts, escalating General and Administrative costs (entropy) in the implemented Enterprise of Row 6.
The Third Law of Enterprise, Ontological Holism. Every Cell or portion of Cell that is not explicit (i.e. is implicit) is guaranteed to be a source of inconsistent assumptions and therefore discontinuities, risking potential conflicts, escalating General and Administrative costs (entropy) and even Enterprise liabilities.

The Fourth Law of Enterprise, Ontological Holism. To avoid misunderstanding and miscommunication about the Enterprise, there should be only a single version of Cells in Rows 1, 2 and 3. However, the Row 3 System Logic can be transformed to more than one Technology and the Row 4 Technology Physics transformed with more than one Vendor Tool as long as content redundancy is controlled.
The **Fifth** Law of Enterprise, Ontological Holism. Any fact that is not classifiable according to the defined classification rules is either not relevant to the Enterprise or not a single-variable, “Primitive” fact. That fact (if it is a fact and if it is relevant to the Enterprise) is likely a “Composite” fact.
The probability of the implemented Enterprise (Row 6) having anything to do with the intentions of the “stakeholders” of Rows 1, 2 or 3 is low to zero, if ...
The probability of the implemented Enterprise (Row 6) having anything to do with the intentions of the “stakeholders” of Rows 1, 2 or 3 is low to zero, if ...

... cells in **Rows 1, 2 or 3 are not made explicit**

or

... cells in **Rows 4 or 5 are not made explicit** and **aligned with the Rows 1, 2 and 3**
The First Law of Reification Incontrovertibility. If Cells in Rows 1, 2 or 3 are not made explicit, whoever is formalizing Cells in Rows 4, 5 and 6 has to make assumptions about Rows 1, 2 and 3 and the probability of the implemented Enterprise of Row 6 having anything to do with the intentions of Rows 1, 2 or 3 is low to zero.

The Second Law of Reification Incontrovertibility. If Cells in Rows 4, 5 or 6 are not made explicit and aligned with the transformations of Rows 1, 2 and 3, whether the Cells in Rows 1, 2 and 3 are made explicit and aligned or not, the probability of the implemented Enterprise of Row 6 having anything to do with the intentions of the “stakeholders” of Rows 1, 2 or 3 is low to zero.
Is building models the purpose of Enterprise Architecture?
Challenge to Enterprise Architects

Reframe the concept of Enterprise Architecture ...

It is not about building models!

It is about solving Enterprise problems while iteratively and incrementally building out the inventory of complete, reusable, Primitive Models that constitute:

Enterprise Architecture.
What are the options, if you do not retain (or did not create) architecture descriptions?
If you don't retain the descriptive representations after you create them (or if you never created them in the first place) and you need to change the resultant implementation, you have only three options:

- **Directly change the system** and see what happens. (High risk!)
- **Recreate ("reverse engineer") the architecture** representations from the existing ("as is") implementation. (Typical for many projects - Takes time and costs money!)
- Scrap the whole thing and **start over again**.

Better:

**Retain description of your enterprise architecture**
Change the Model before you Change the System!

Retain description of your enterprise architecture
Enterprise Architecture Repository

Enterprise Architecture Description

- Strategy
- Organisation and processes
- Information systems
- Infrastructure

as-is

Agree on objectives
Understand as-is situation
Determine change
Define to-be situation
Implement changes

Change needed

to be (update)

Agree on objectives
Understand as-is situation
Determine change
Define to-be situation
Implement changes

Change needed

as-is

to be (update)

as-is

to be (update)