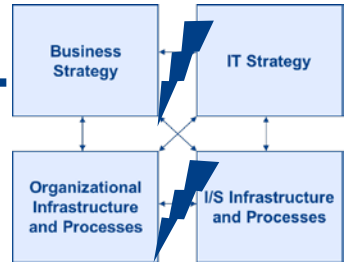
Enterprise Architecture – Dealing with Complexity and Change

Prof. Dr. Knut Hinkelmann
knut.hinkelmann@fhnw.ch

Changes and Alignment of Business and IT



- Change can affect the alignment of business and IT on both strategic and operational level
- On strategic level the alignment of business and IT has to deal with problems like the following:
  - What IT innovations are needed to react on market requirements?
  - How can we successfully integrate new firms after an acquisition?
- On the operational level questions can be:
  - Which business units and users will be affected by the migration of an application?
  - What information does the business process need and how can it be stored?
  - What applications and infrastructure technologies do we require to run new or redesigned business processes?
- Many organisations lack transparency due to the number and frequency of their organisational changes and have problems to answer these questions.

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.
- This description is what we call an "**Architecture**".

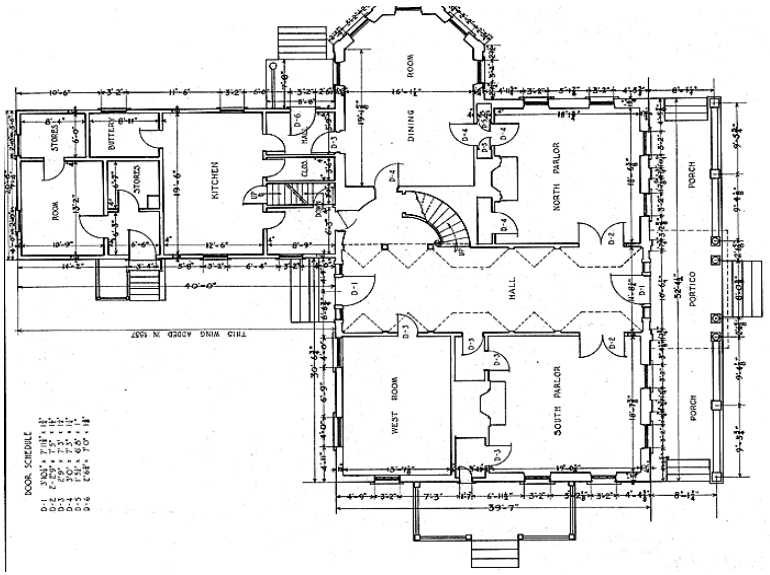
(John Zachmann, 2012)

Architecture – What is it?

- Is this an Architecture?

Architecture – What is it?

- Is this an Architecture?



Architecture – What is it?

"***Architecture***" names that which is fundamental about a system; the set of essential properties of a system which determine its form, function, value, cost, and risk. That which is **fundamental** to a system takes several forms:

- its **elements**: the constituents that make up the system;
- the **relationships**: both internal and external to the system; and
- the **principles of its design and evolution**

ISO/IEC/IEEE 42010 - http://www.iso-architecture.org/ieee-1471/cm

Enterprise Architecture: Overall View on the Enterprise

An **Enterprise Architecture** contains all *relevant*

- **Business structures** (e.g. organisation structure, business processes)
- **IT structures** (e.g. information systems, infrastructure)
- and their **relationships**

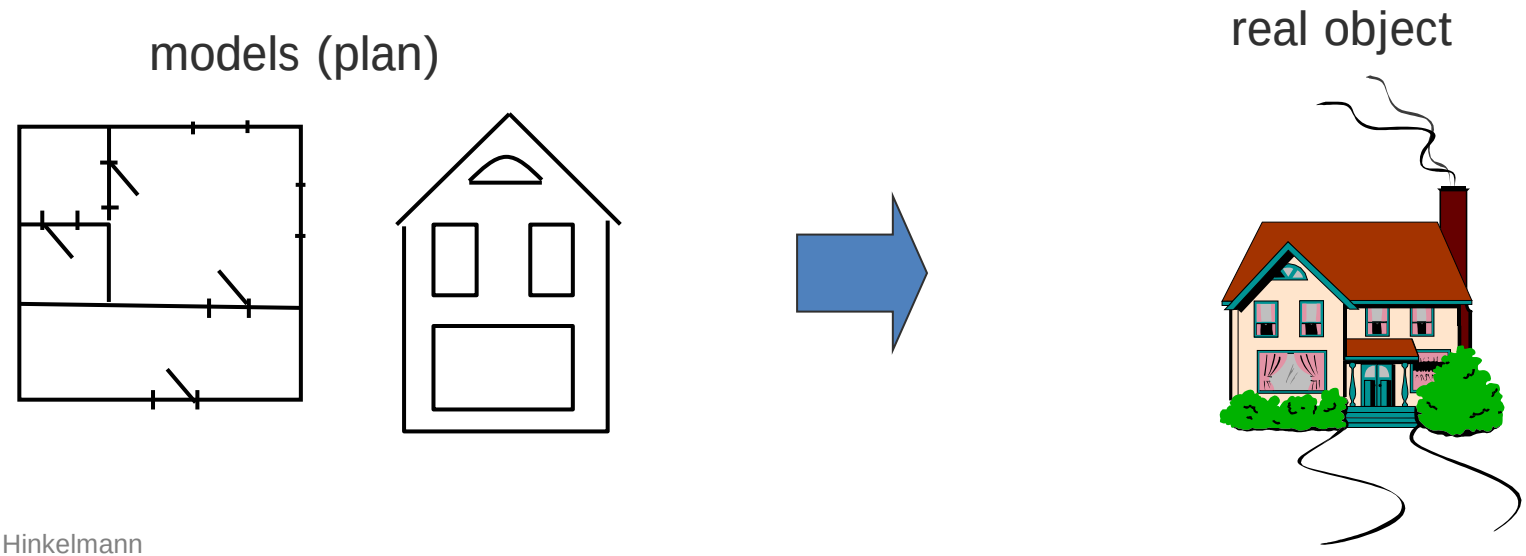
Architecture and Architecture Description

- 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/

Archtecture 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.

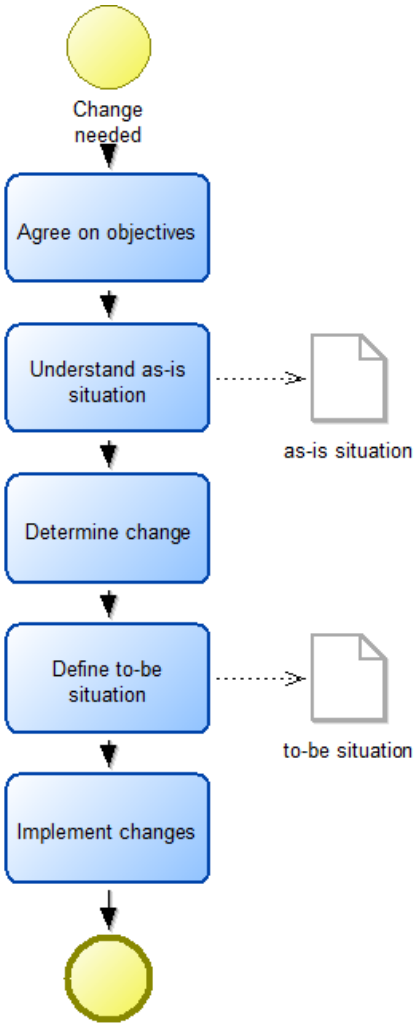


Enterprise Architecture (Description) – What is it?

- An "Architecture" (for anything) would be the total set of descriptive representations (models) relevant for describing a complex object such that it can be created and that constitute a baseline for changing the object after it has been instantiated.
- Therefore "**Enterprise Architecture**" would be the total set of models relevant for describing an Enterprise, that is, the descriptive representations required
  - ♦ to create a (coherent, optimal) Enterprise and
  - ♦ to serve as a baseline for changing the Enterprise once it is created.

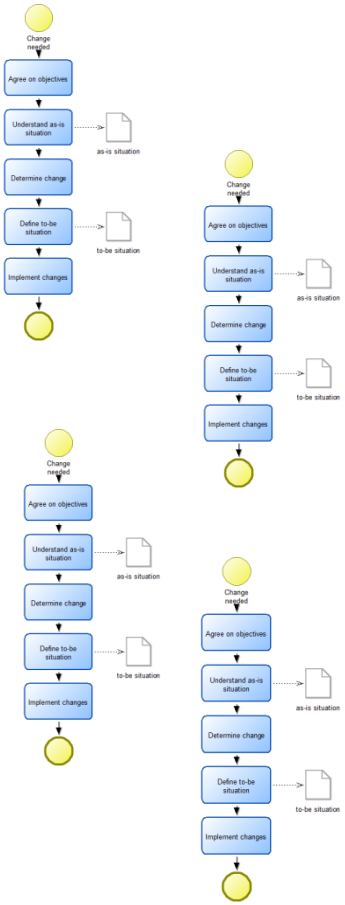
Adapted from Zachman (2012)

Typical (Change) Projects



- Typically organisations go through several stages in a change project:
  - recognizing the need to change
  - agreeing on the objectives of the change and a vision that describes a better future
  - understanding what the organisation is changing from (as-is model)
  - determine what needs to change
  - designing the new way of working and its support and management (→ to be model)
  - testing and implementing changes

Architecture Descriptions in an Enterprise



Typically …

… there are a large number of projects

- ♦ running concurrently or
- ♦ building on the result of previous projects

… projects have an extensive documentation of their (intended) result

… each project manages its own documentation which is not available for other projects

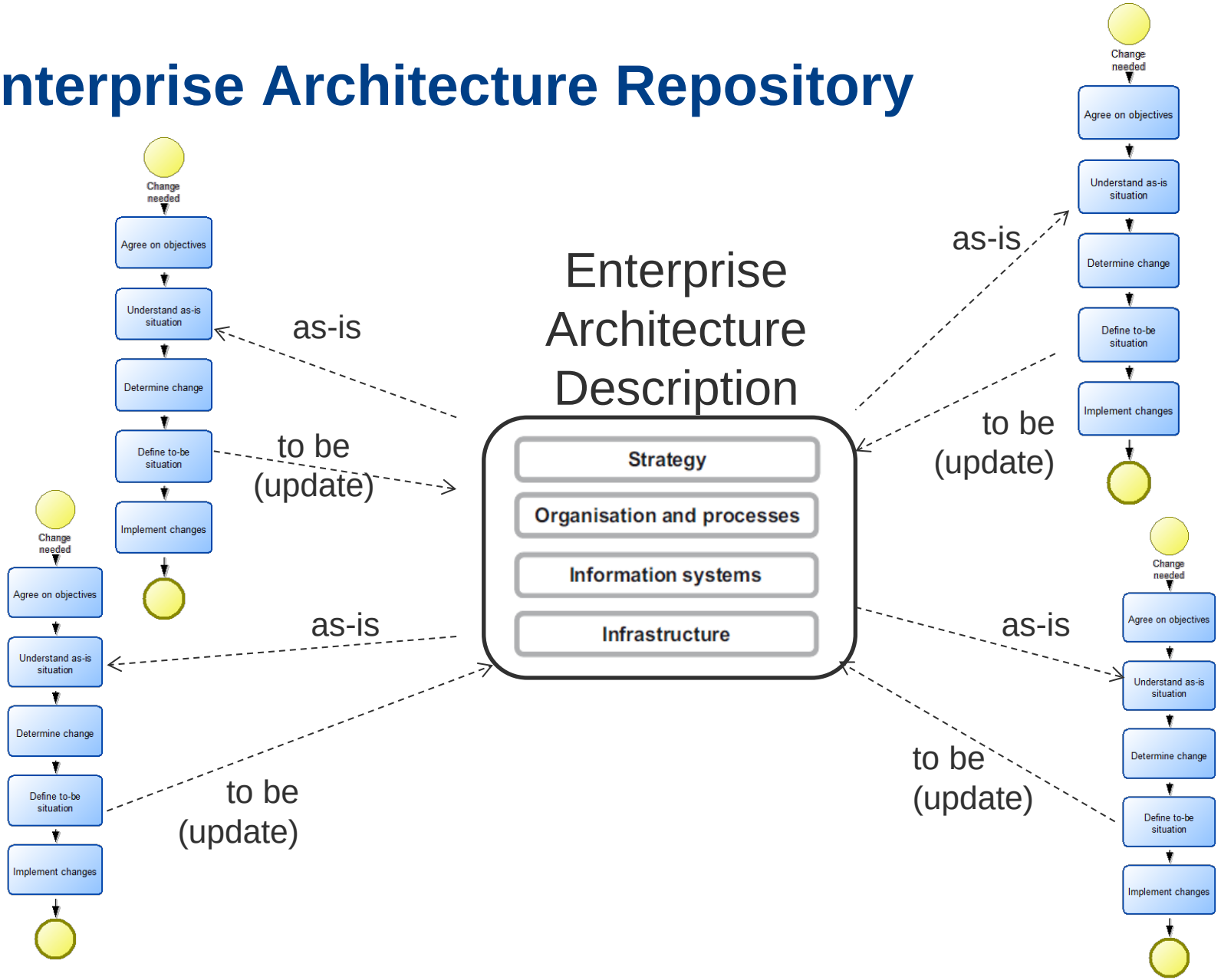
… there is a lack of coordination between projects

The Need for Architecture Description

- **Complexity:** If you can't describe it, you can't create it (whatever "it" is).
- **Change:** 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:
  - Change the instance and see what happens. (High risk!)
  - Recreate ("reverse engineer") the architectural 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**

(John Zachmann, 2012)

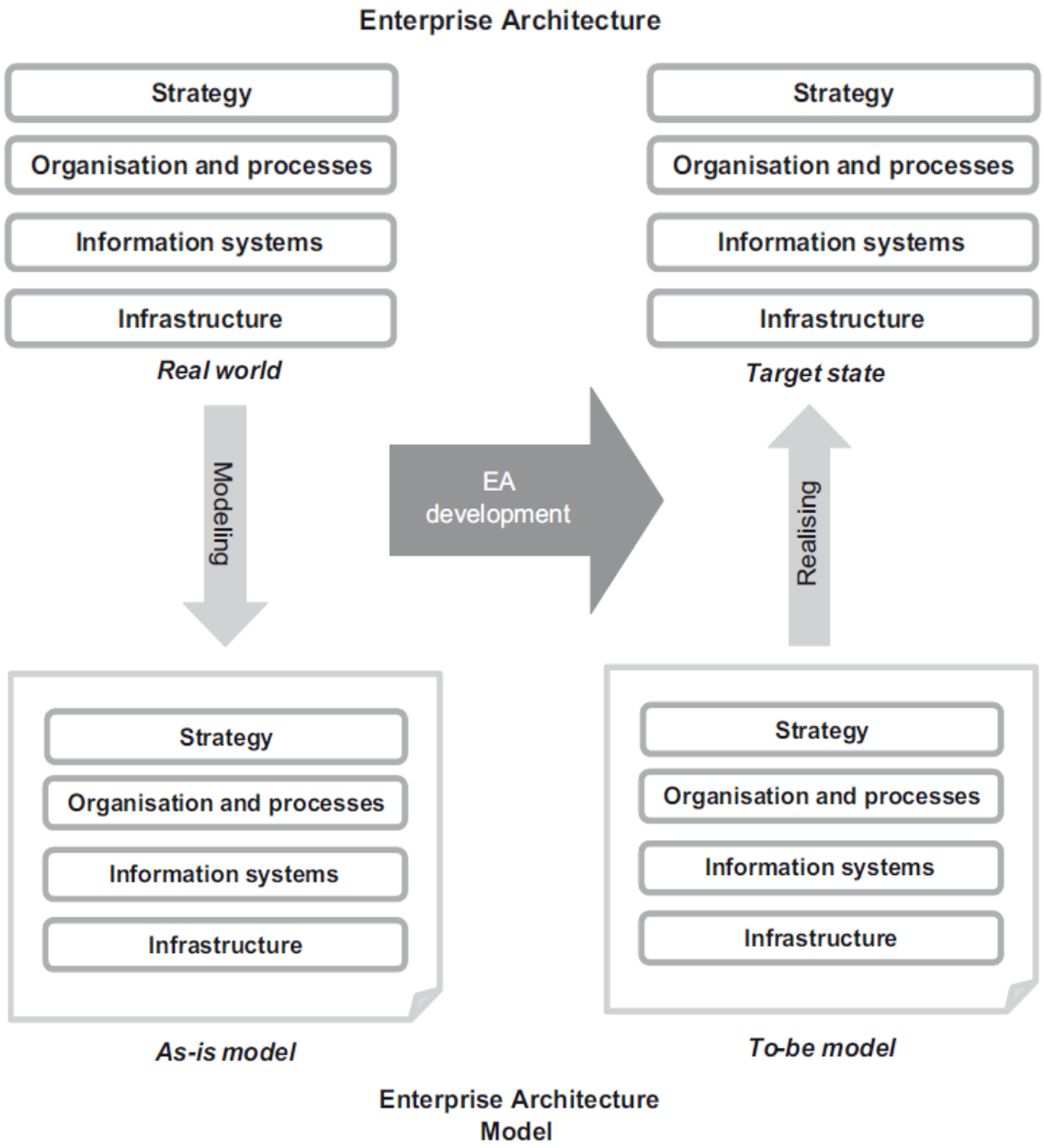
Enterprise Architecture Repository



Use of Enterprise Architecture: Managing Change and Decision Making

- Change the architecture before you change the object!
- The Enterprise Architecture is managed as a program that facilitates
  - systematic organization change
  - continuously aligns technology investments and projects with organisation mission needs.
- Enterprise Architecture is updated continuously to reflect changes
- It is a primary tool for baseline control of
  - complex, interdependent enterprise decisions and
  - communication of these decisions to organization stakeholders.

(Schekkermann 2008, p. 107)



(Ahlemann et al. 2012, p. 17)

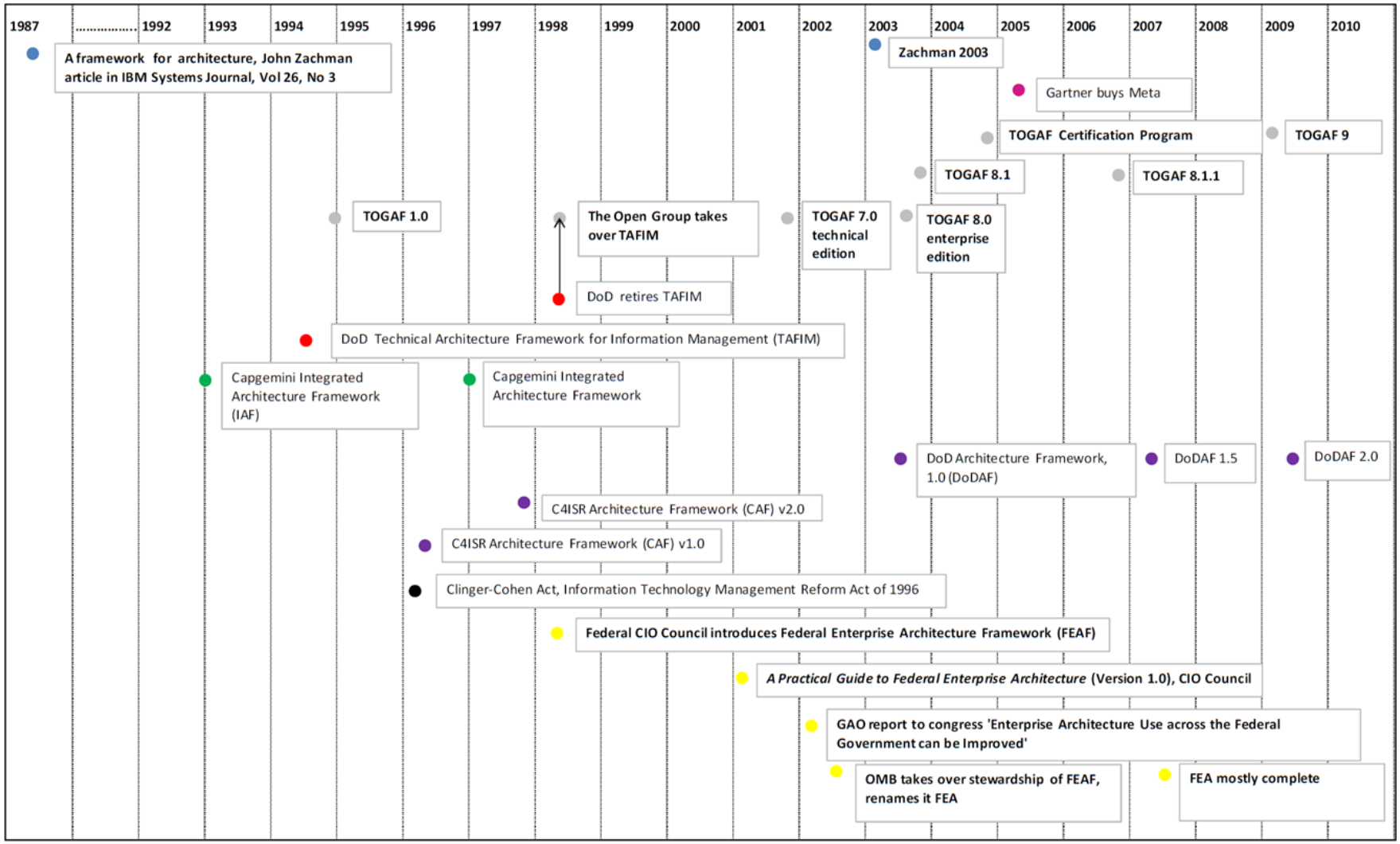
Enterprise Architecture Frameworks

Prof. Dr. Knut Hinkelmann

Architecture Framework

- An *Architecture Framework* establishes a common practice for creating, interpreting, analyzing and using architecture descriptions
- It is a logical structure for classifying and organising the descriptive representations of a system

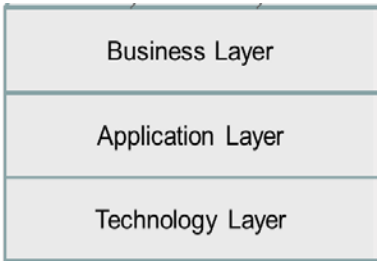
Timeline of Enterprise Architecture Frameworks



(Bespoke Systems 2012)

Enterprise Architecture Frameworks

- We can distinguish two main types of structures for Enterprise Architecture Frameworks:
  - ***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***



  - ***Matrix*** of aspects and perspectives, e.g.
    - ***Zachmann Enterprise Architecture Framework***
      - An enterprise ontology

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

Zachman, J.A., 1987. A framework for information systems architecture. *IBM Systems Journal*, 26(3).

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

Abstractions

Perspectives

(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 contrained 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

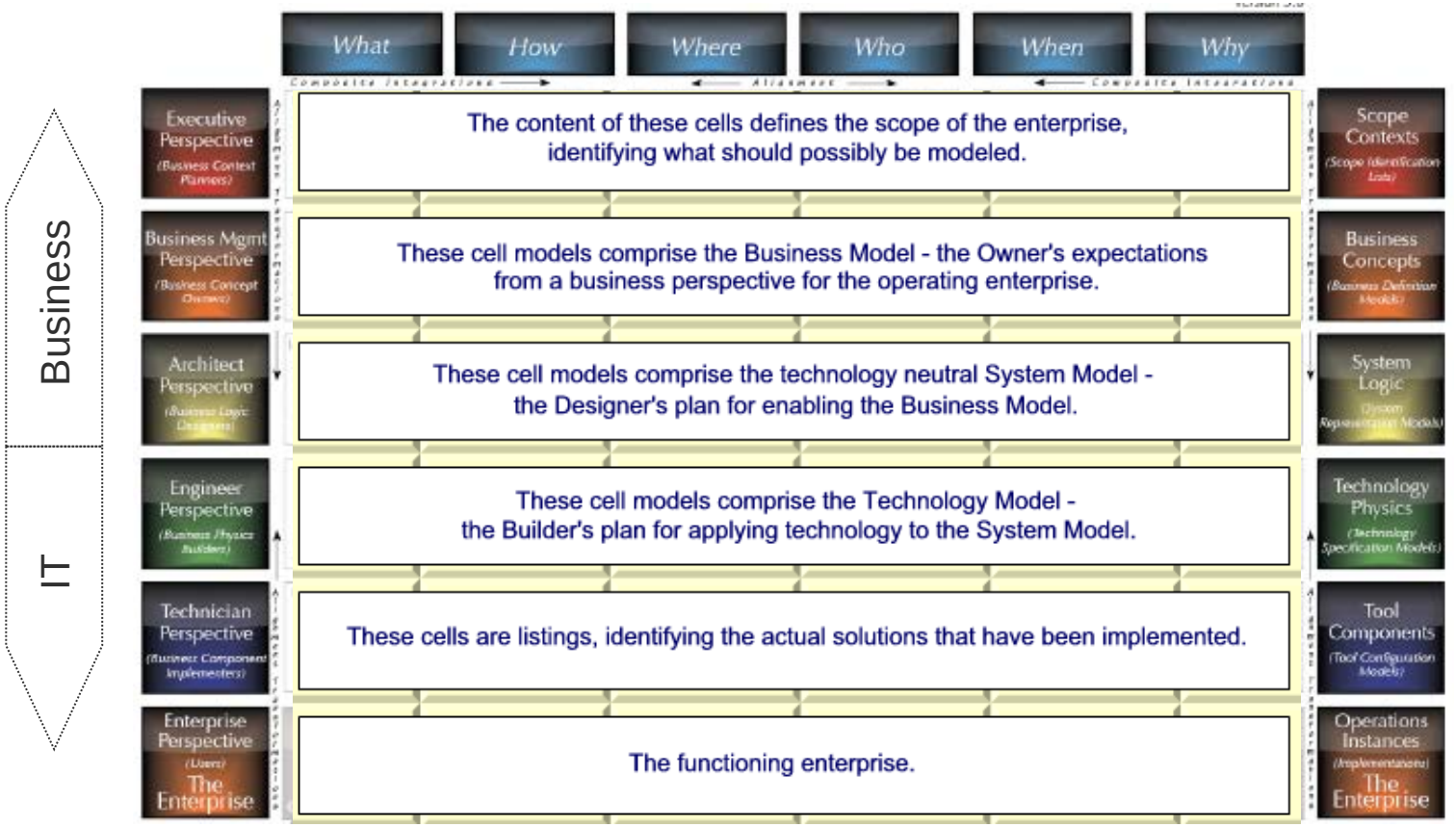
(Zachman 1987)

Dimension 1: Architectural Representations with analogies in Building and Information Systems

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

(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)

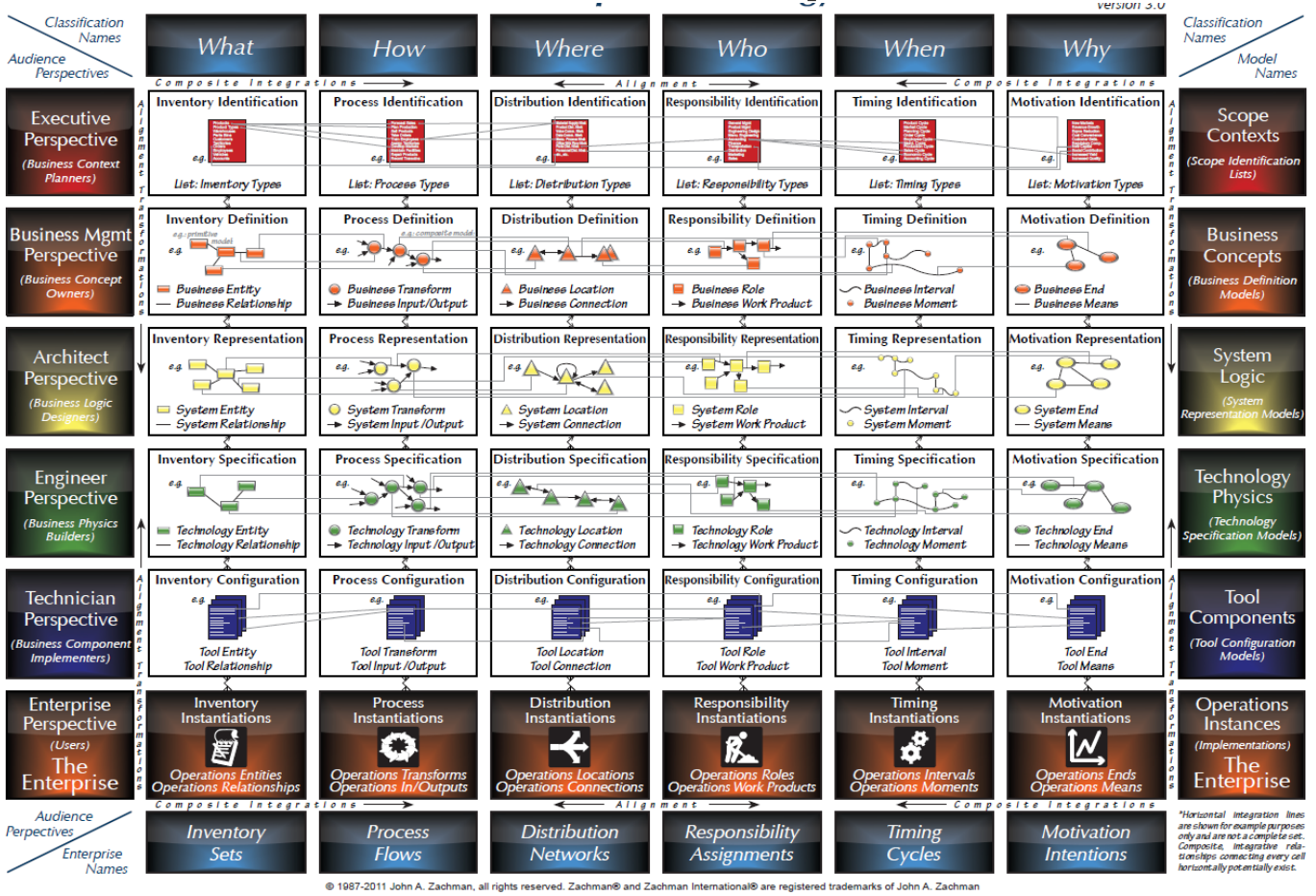
Abstractions for Manufacturing

(Zachman 2012)

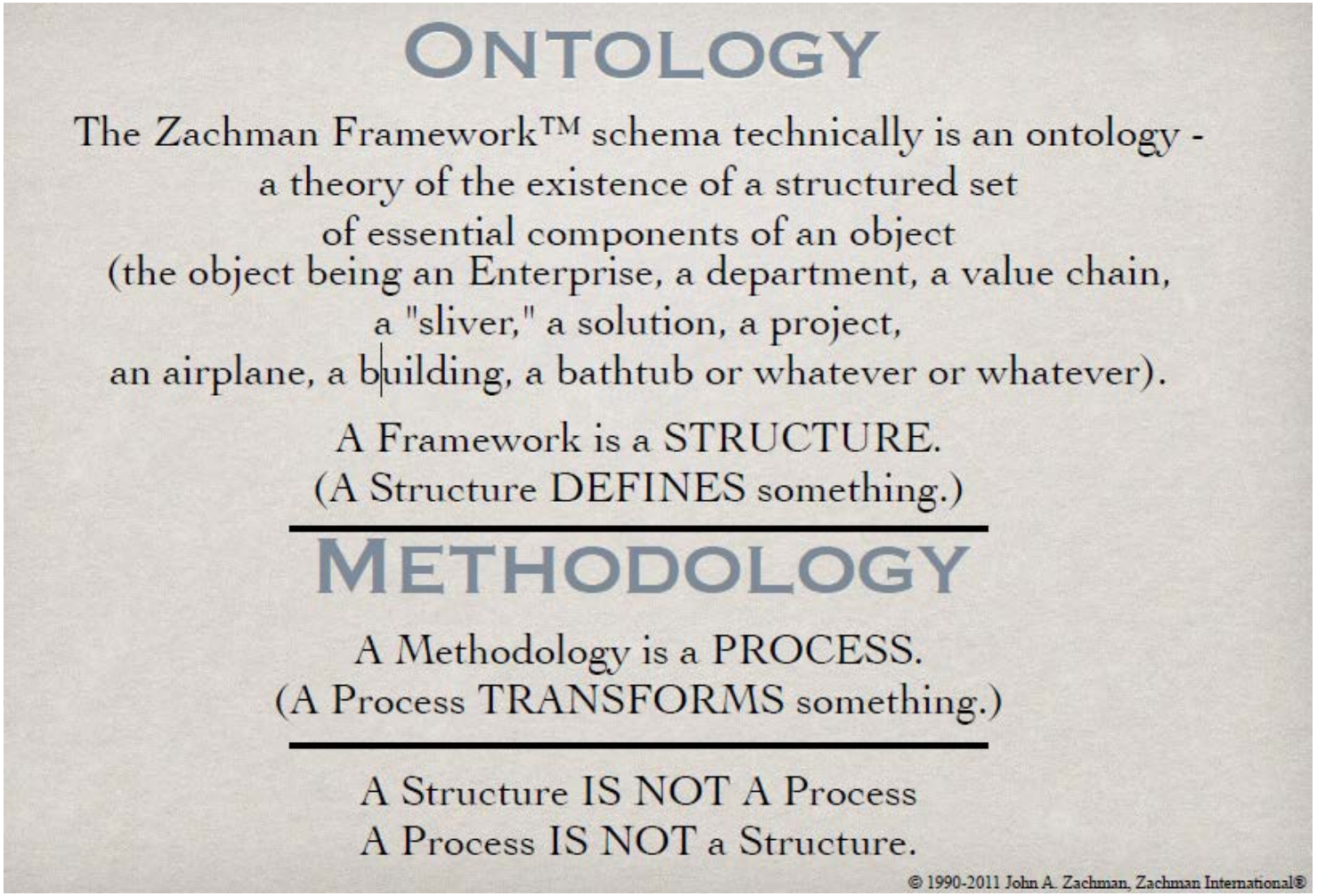
The Zachman Framework for Enterprise Architecture – Enterprise Ontology

Each cell contains models

Abstractions/Aspects



The Zachman Framework is not a Methodology

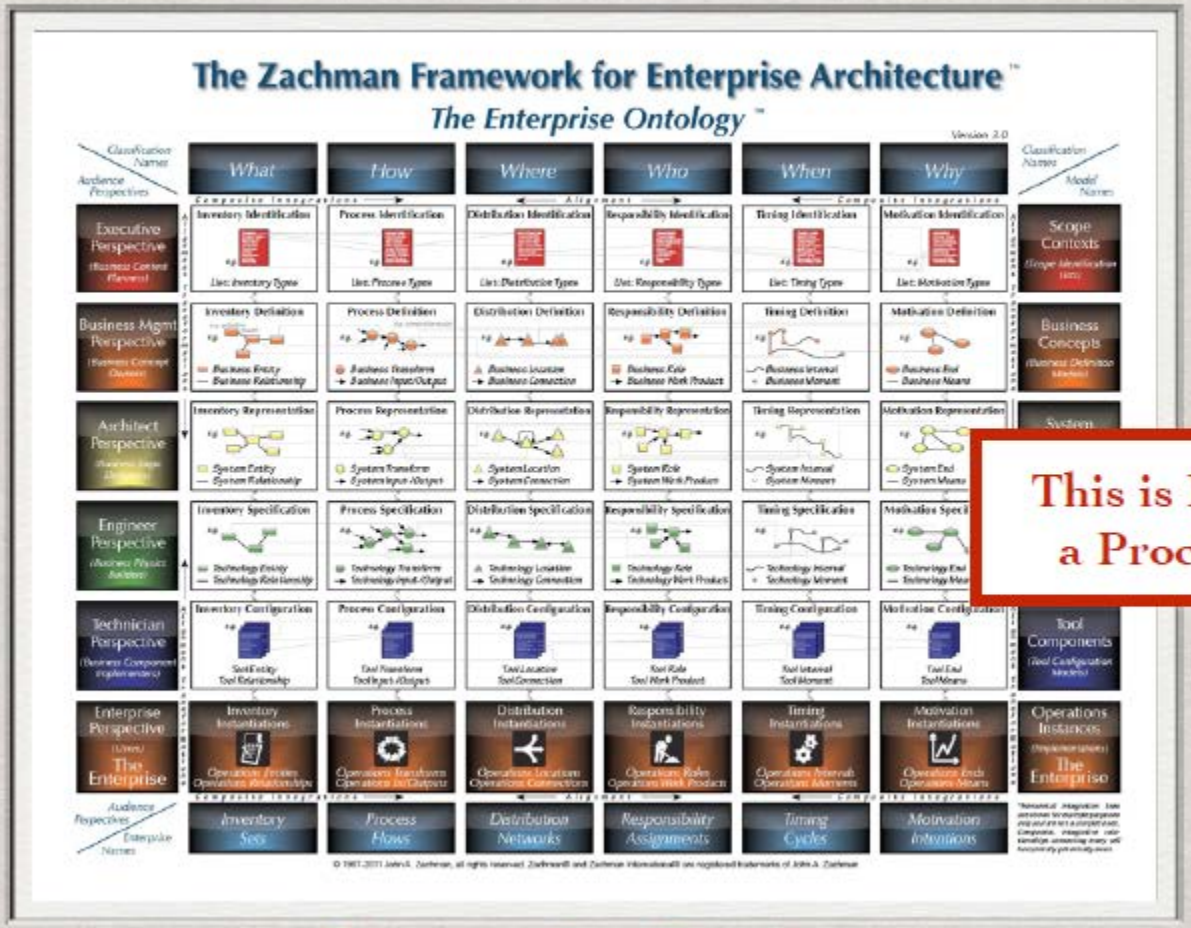


Ontology vs Methodology

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.

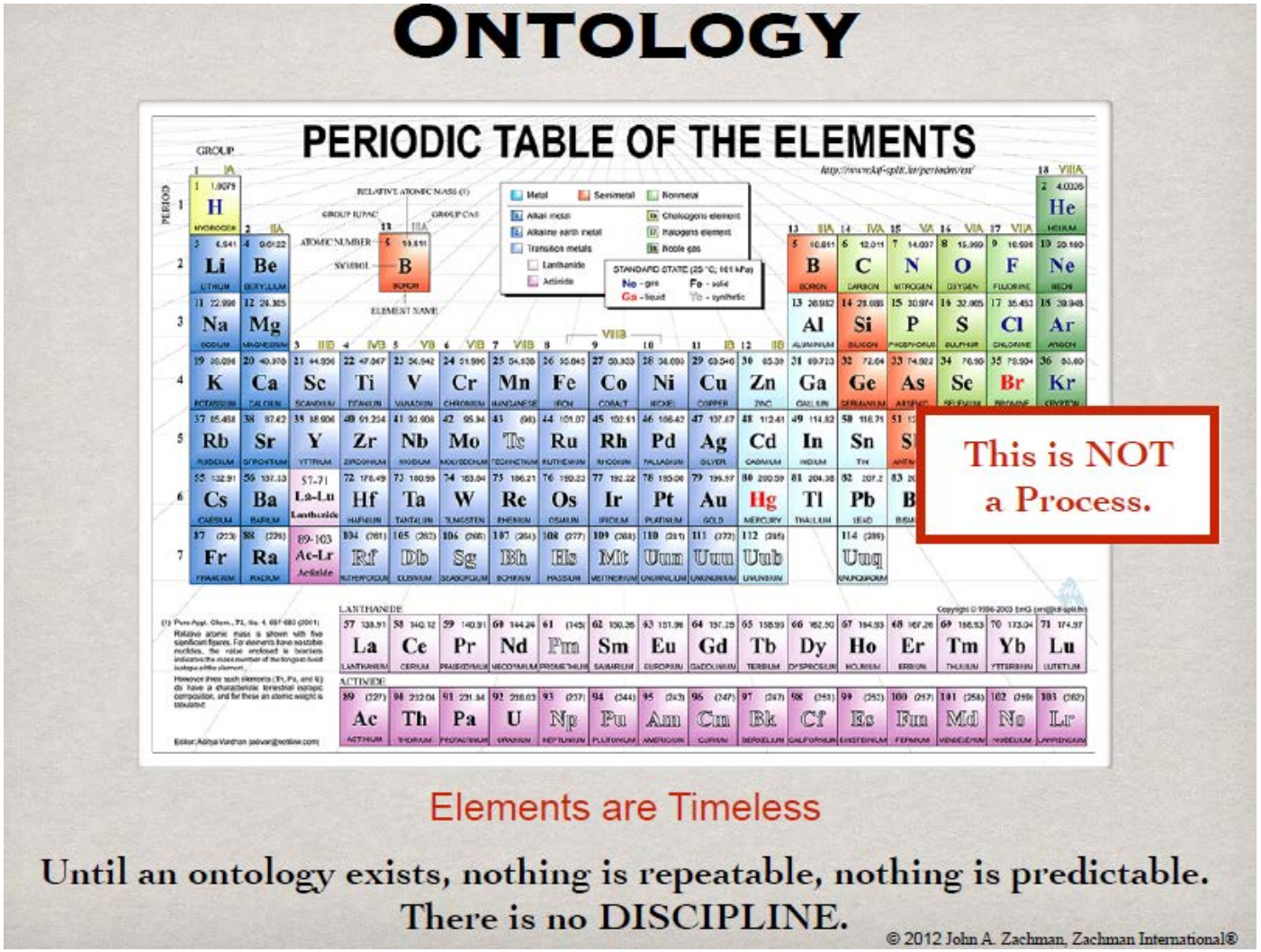
A Methodology produces "**Composite**" (compound) implementations of the Primitives.

Ontology



"Primitives" are Timeless.

Analogy: Chemistry



Analogy: Chemistry

PROCESS

(METHODOLOGY)

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

$$HCl + NaOH \rightarrow NaCl + H_2O$$

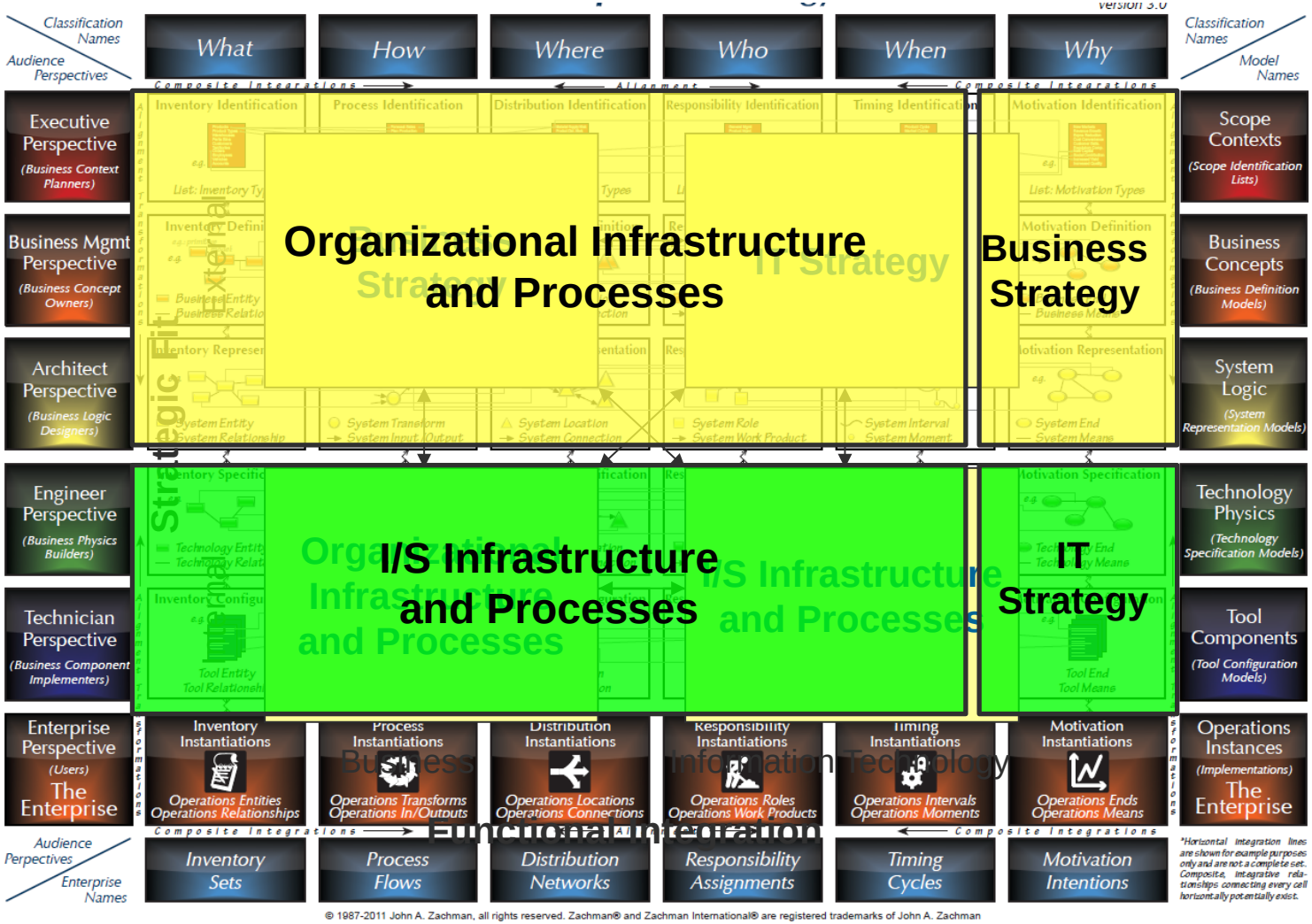
COMPOUNDS

Salt	NaCl
Aspirin	$C_9H_8O_4$
Vicodin	$C_{18}H_{21}NO_3$
Naproxen	$C_{14}H_{14}O_3$
Ibuprophen	$C_{13}H_{18}O_2$
Viagra	$C_{22}H_{30}N_6O_4S$
Sulphuric Acid	H_2SO_4
Water	H_2O

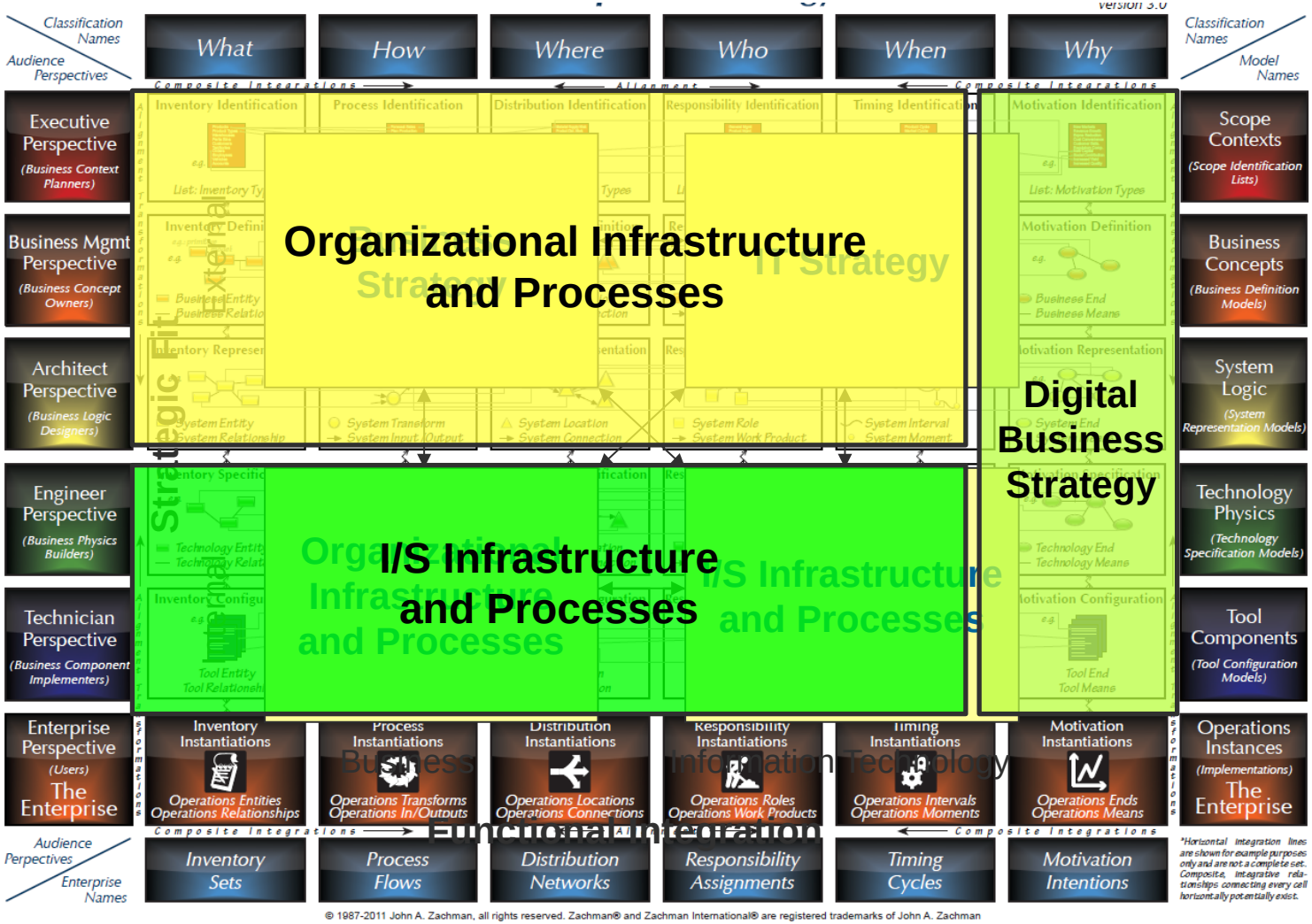
etc., etc., etc.

This is NOT an Ontology.

Strategic Alignment Model and Zachman Framework



Strategic Alignment Model and Zachman Framework



Models 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

Organisation model

Logical data model

Motivation Model

Process model

Physical data model

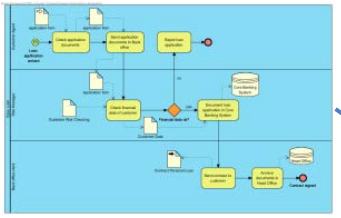
Workflow model

Relations between Models and Model Elements

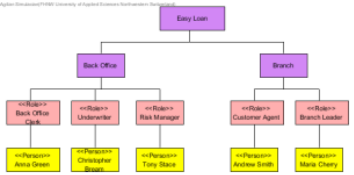
- There are relations between (elements of) the models
- Horizontal Relations: In same perspective, e.g.
  - Data used in a process
  - Application implementing a process activitiy
- Vertical relations: Between different perspectives
  - Implementation of an application
  - Database model for an entity relationship model

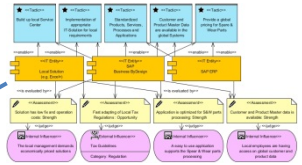
Enterprise Architecture Modeling – Examples of Models Kinds

Process Model

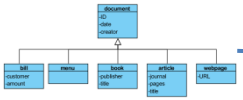


Organisation Model



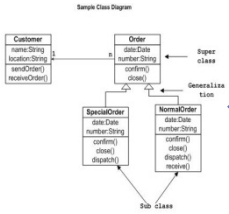


Business Motivation

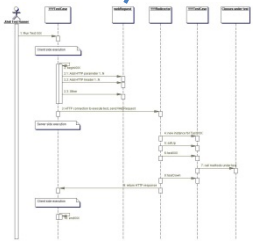


Data/Documents

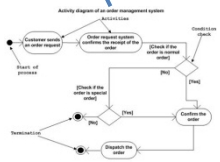
Fact Type Model



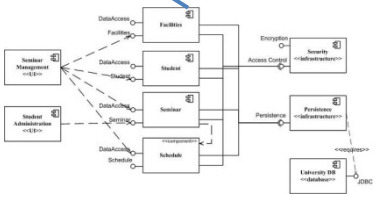
UML class diagram



UML sequence diagram



UML activity diagram



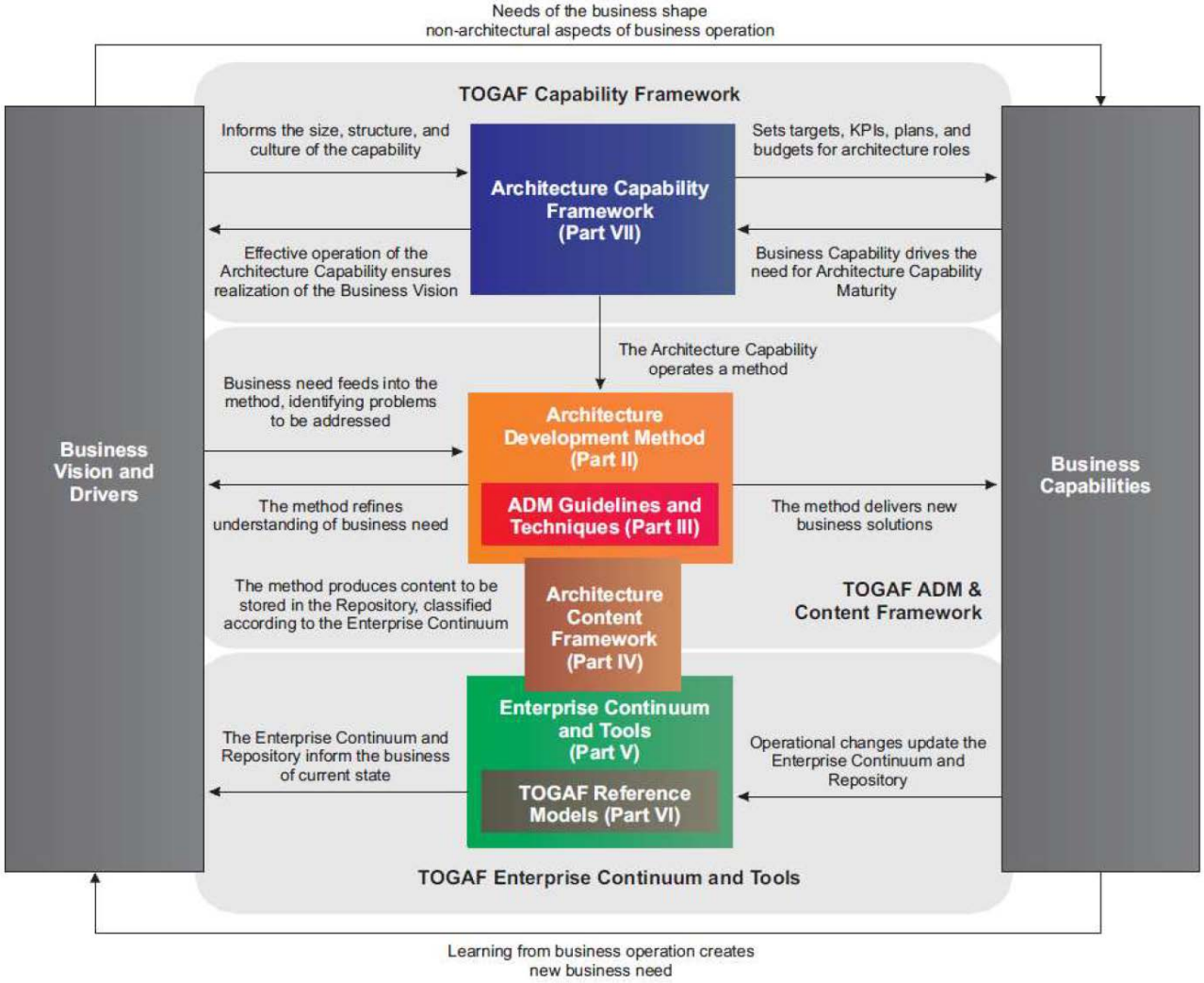
UML component diagram

TOGAF – The Open Group Architecture Framework

TOGAF – The Open Group Architecture Framework

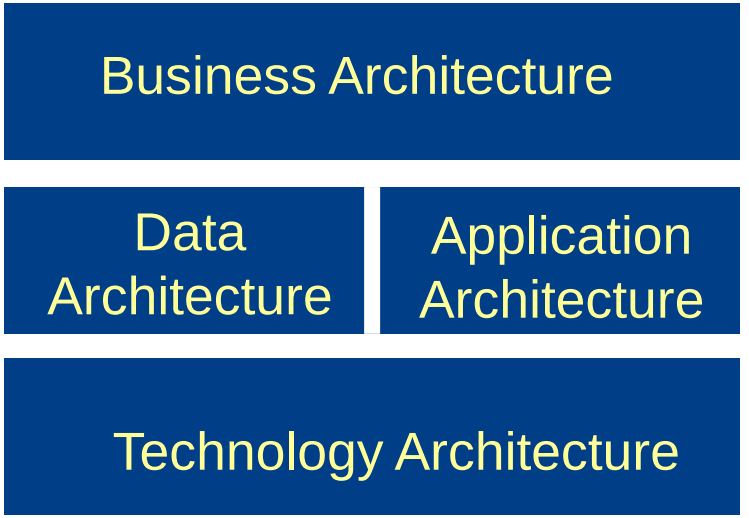
- Developed and continuously evolved since the mid-90's by The Open Group's Architecture Forum
- At the heart of the framework are the
  - ♦ Architecture Development Method (ADM) and
  - ♦ The Ccontent Framework
- http://www.opengroup.org/togaf/

Structure of the TOGAF Document



(The Open Group 2009, p. 4)

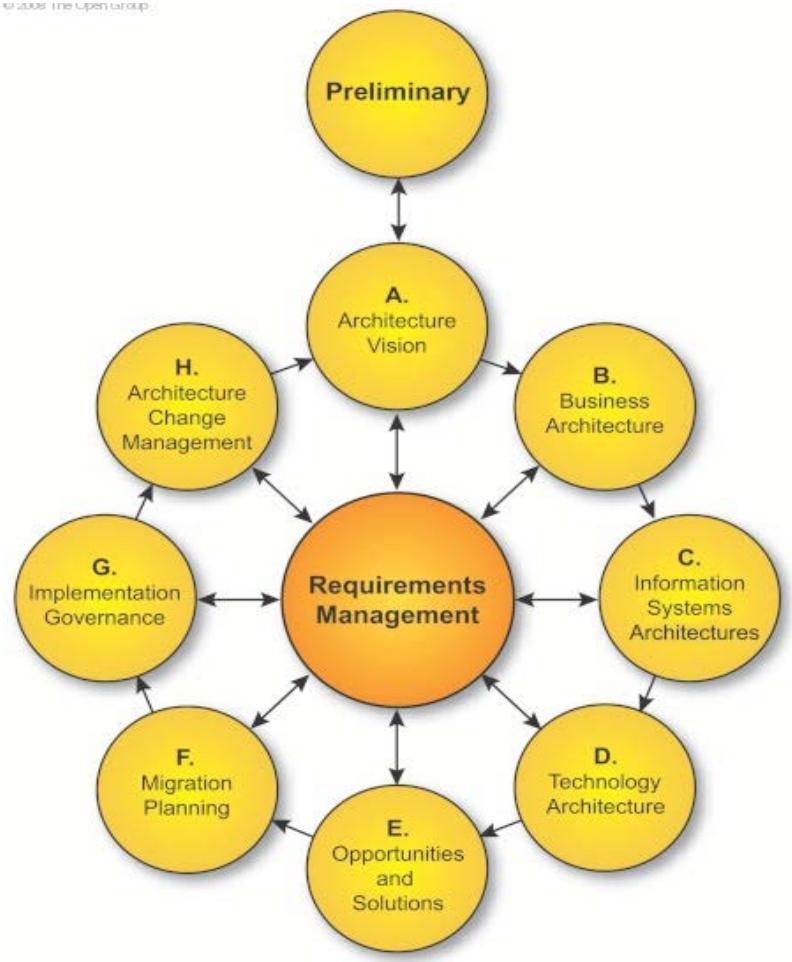
TOGAF Architectures



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 releations as well as principles for the organisation and the management of resources
  - **Application Architecture**
    - information systems and their relations to business processes
- **Technology Architecture**
  - currenct technical realisation and future enterprise-specific standards like operating system, middleware, infrastructure

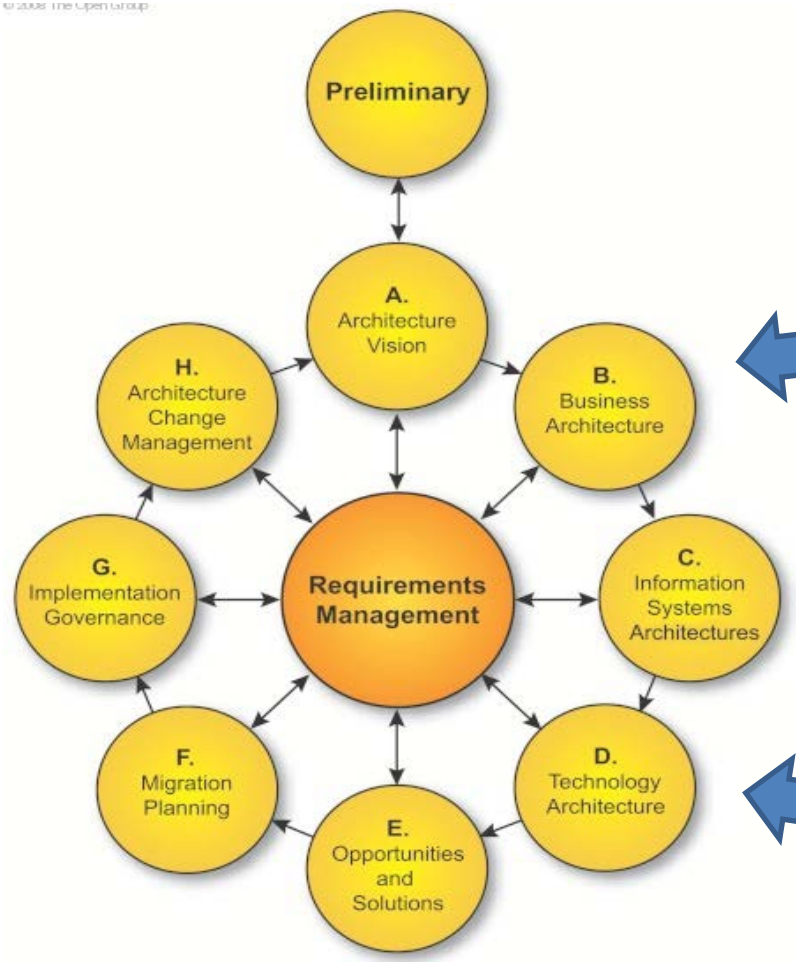
TOGAF Architecture Development Method (ADM)



- 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



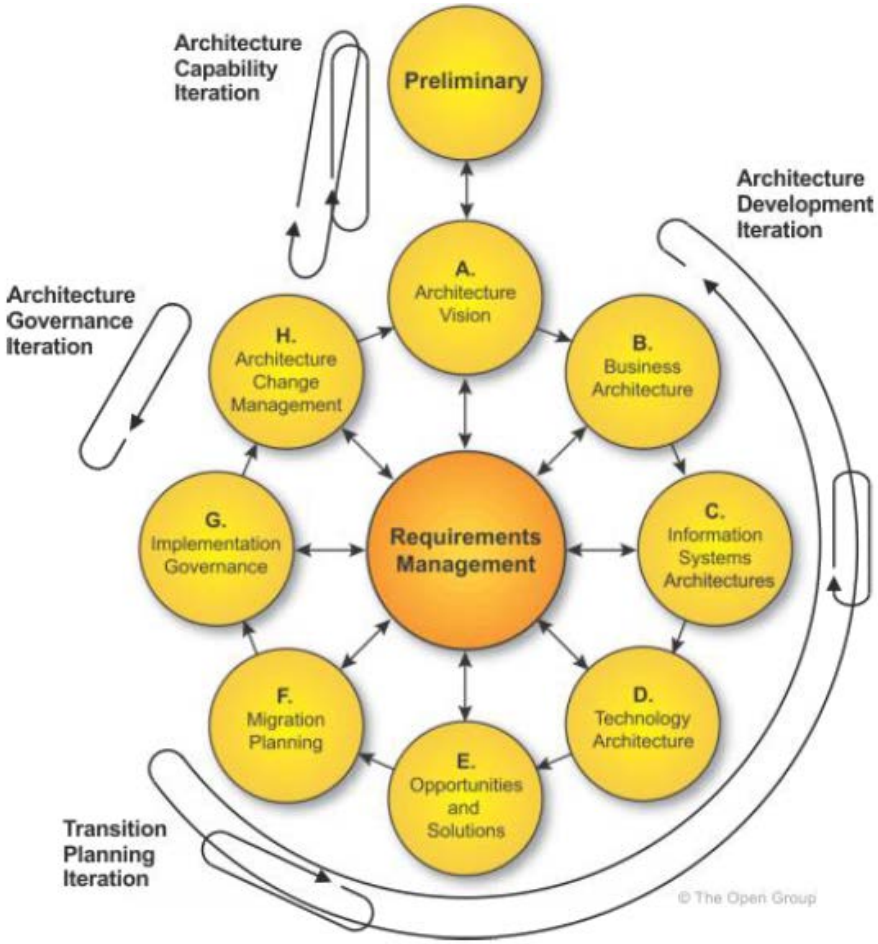
Business Architecture

Data Architecture

Application Architecture

Technology Architecture

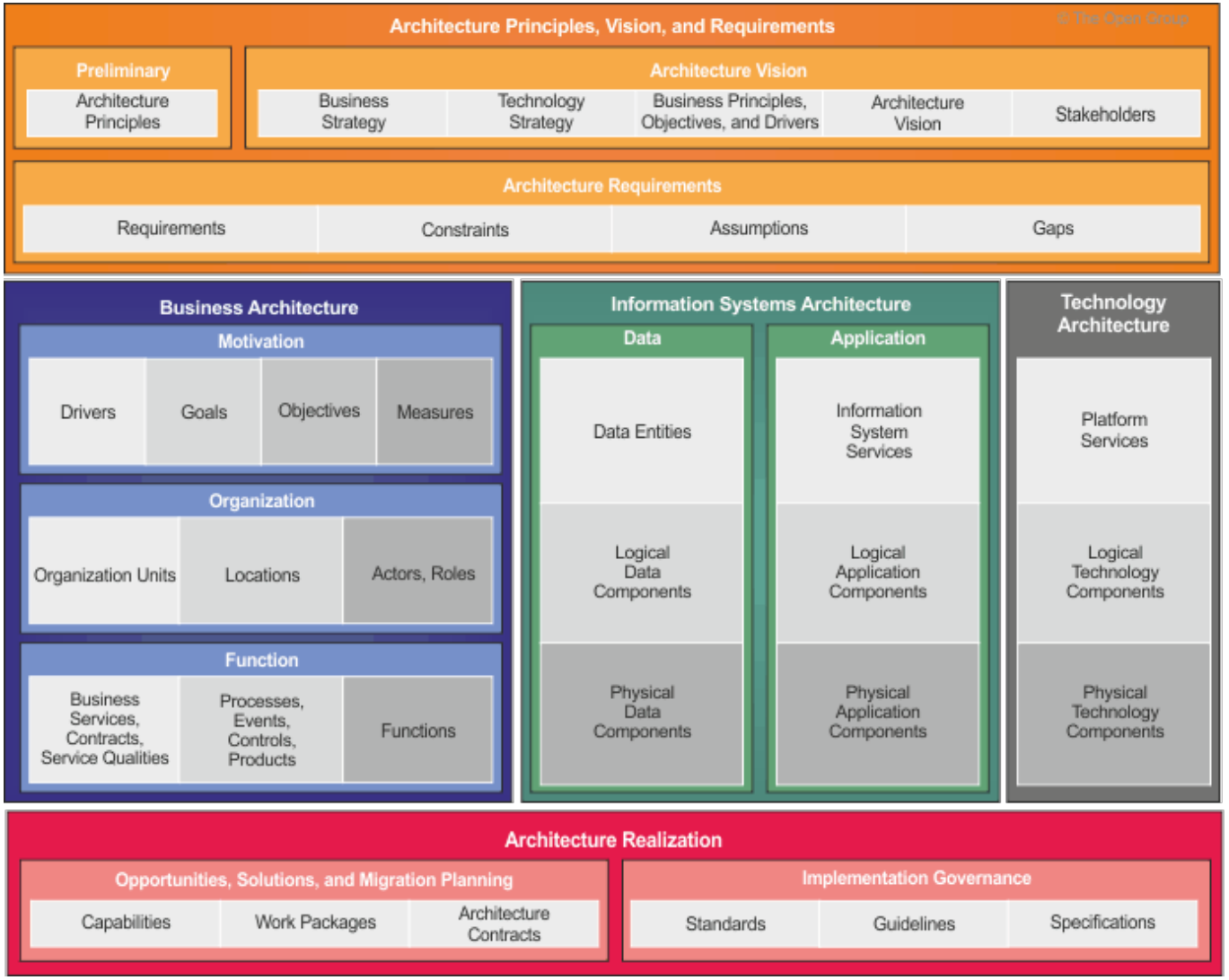
TOGAF Architecture Development Method (ADM)



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

(The Open Group 2011)

TOGAF Content Metamodel



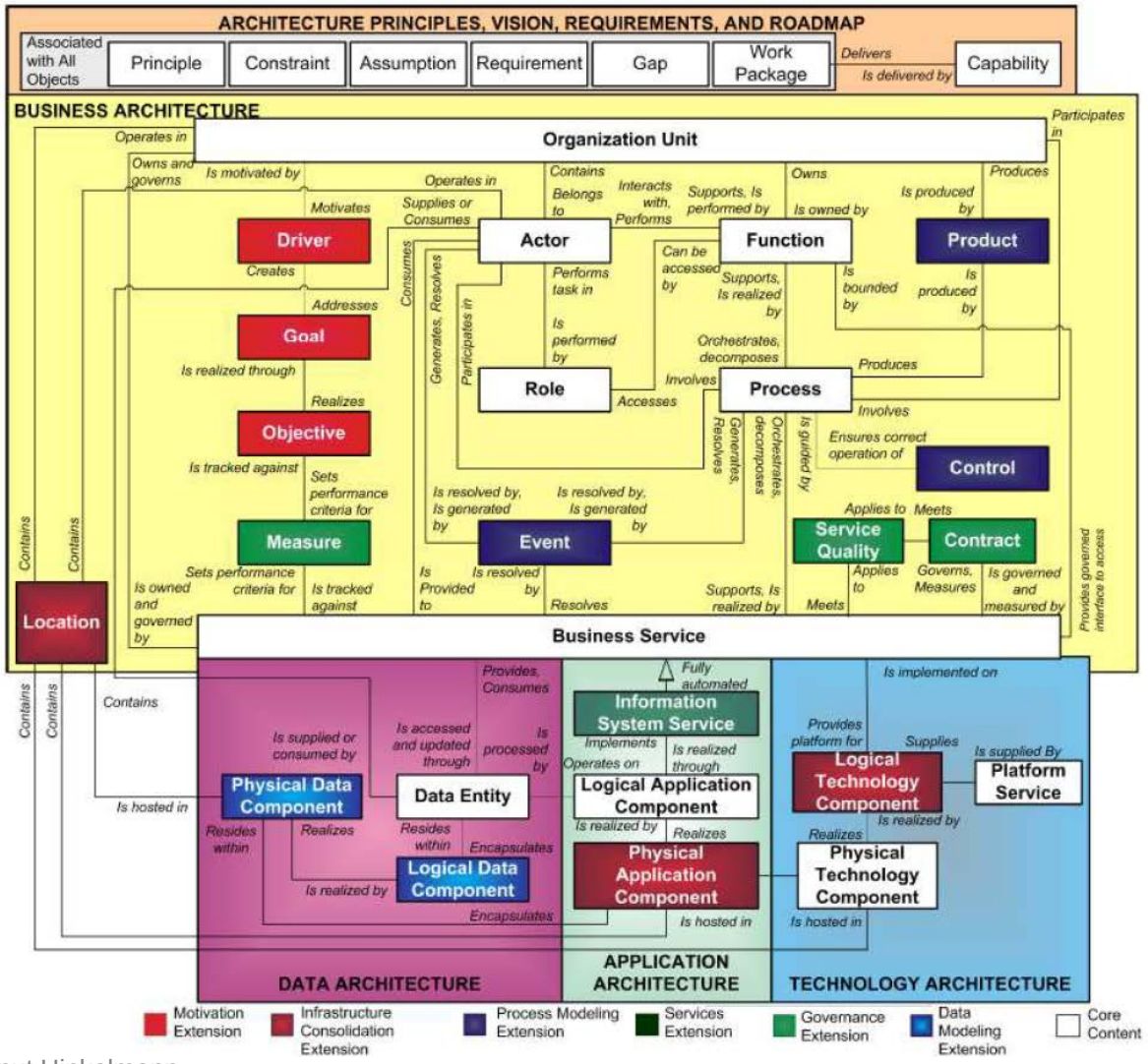
http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap33.html (The Open Group 2011)

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 buiness users or actors
  - identifies artifacts that can be used to represent them.

http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap33.html (The Open Group 2011, Part IV)

TOGAF: Architecture Content



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)

(The Open Group 2009, p. 379)

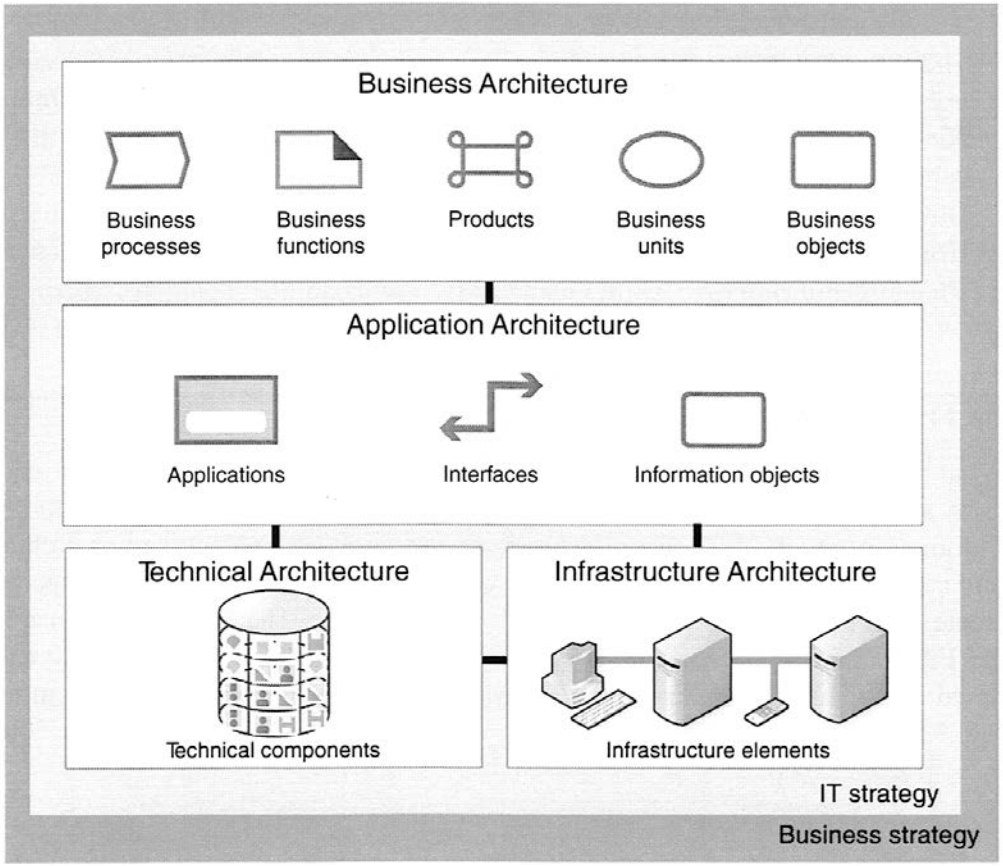
TOGAF: Architecture Content Framework

- 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)

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

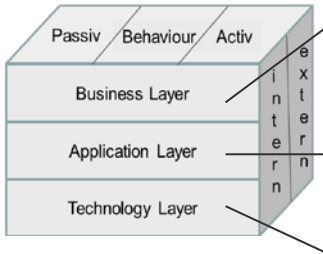
- **Business Architecture**
  - Describing main entities that determine the business: business processes, functions, products, business units and business objects.
- **Application Architecture**
  - documentation of the information systems landscape, i.e. information systems, their data und interfaces und 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

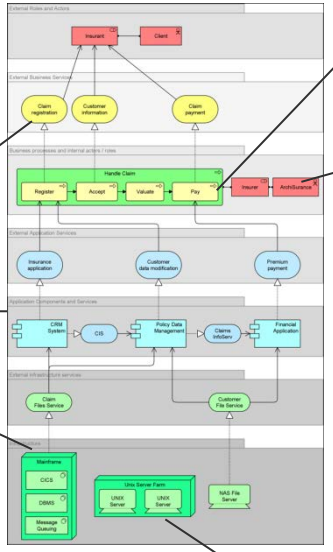
Enterprise Architecture Modeling

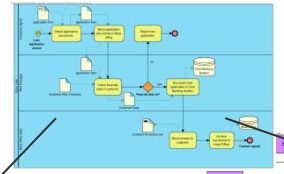
Overview: ArchiMate

Detailed Models

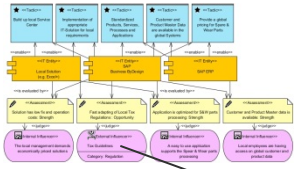
Overview: Zachman





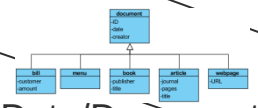


BPMN



Business Motivation

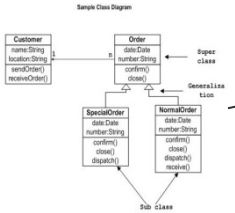
Organisation Model



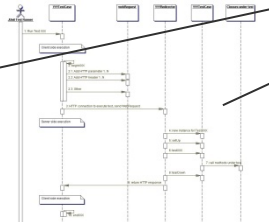
Data/Documents

Business Rules

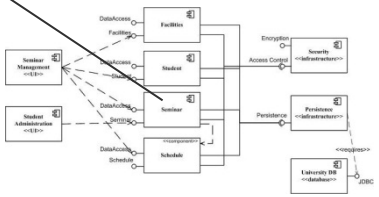
Fact Type Model



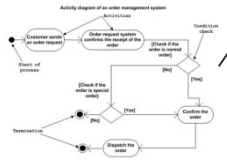
UML class diagram



UML sequence diagram



UML component diagram



UML activity diagram