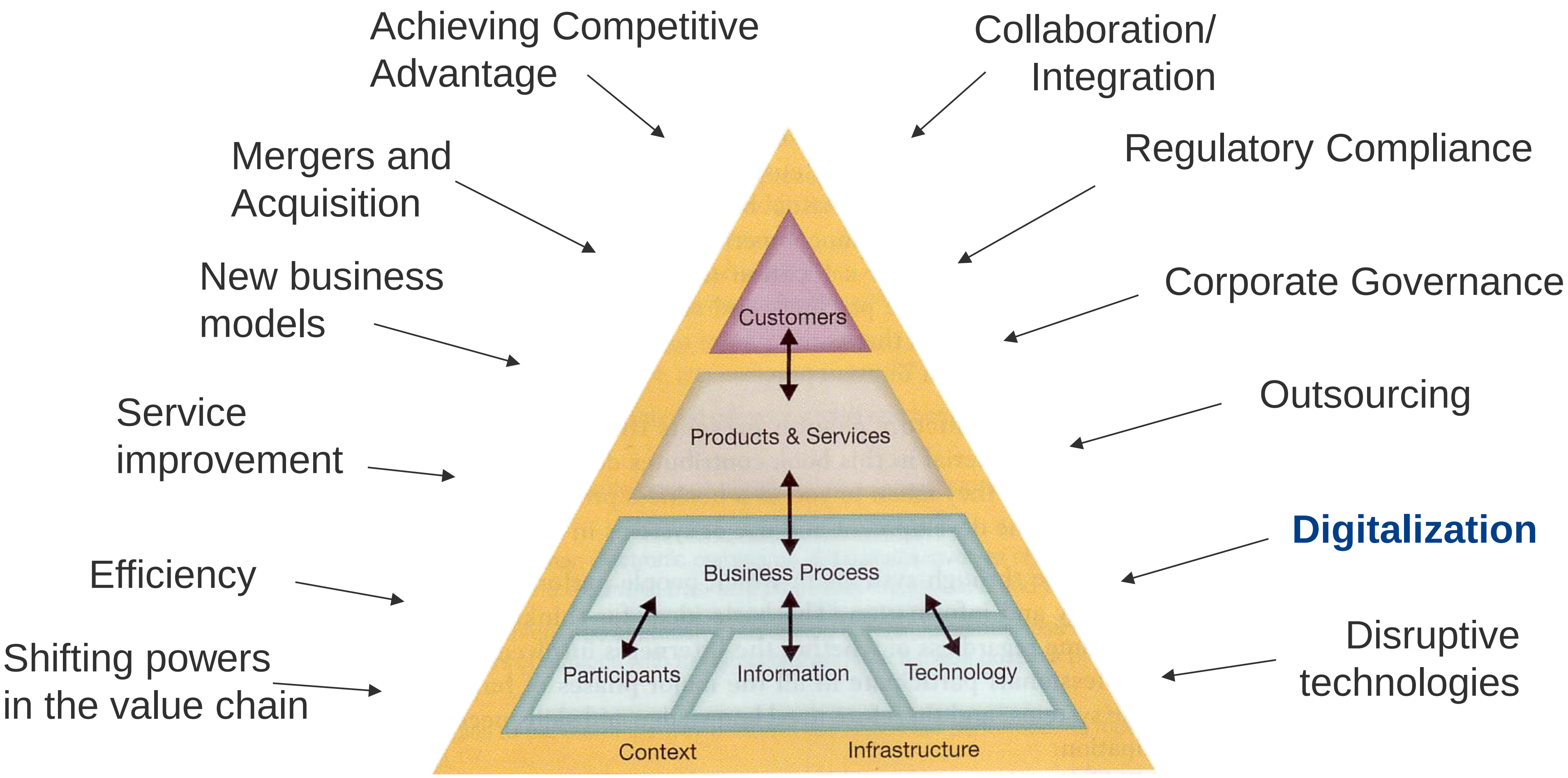
Enterprise Architecture –
Dealing with Complexity and Change

Agility

Increasingly dynamic environment: Challenges confronting an Enterprise



Achieving Competitive Advantage

Collaboration/ Integration

Mergers and Acquisition

Regulatory Compliance

New business models

Corporate Governance

Service improvement

Outsourcing

Digitalization

Efficiency

Disruptive technologies

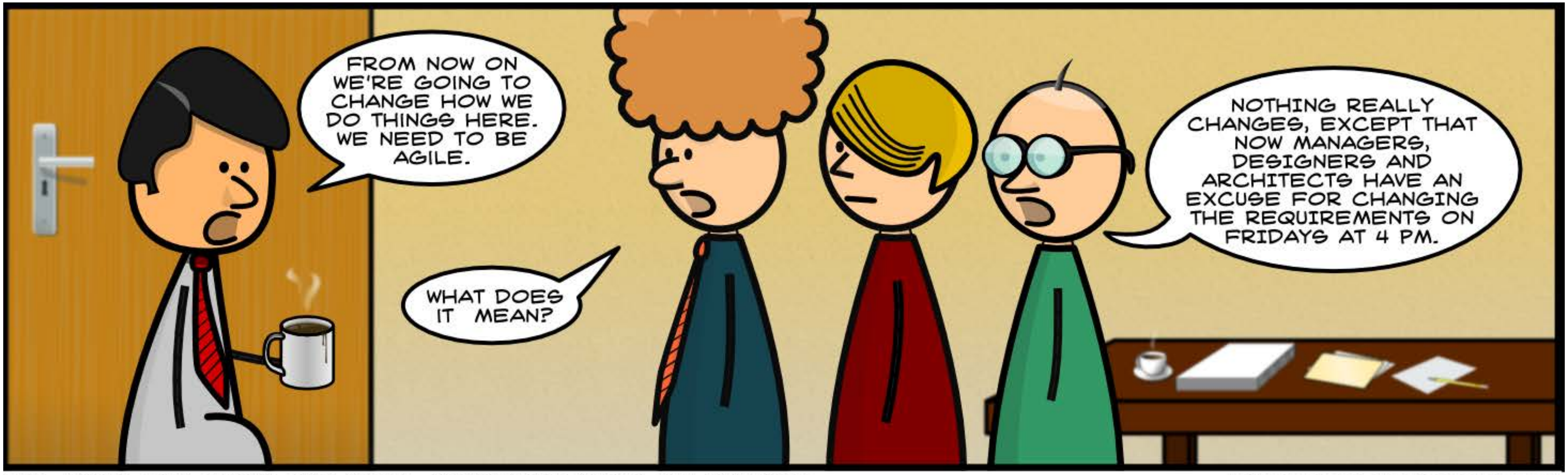
Shifting powers in the value chain

Agility: Demand for Continuous Change

- To improve their chances of survival, enterprises need to be agile.
- Agility is the ability of enterprises to
  - quickly **adapt themselves to changes** in their environment and
  - **seize opportunities** as they avail themselves
  - have **flexibility** to deal with individual customer requirements, to reduce response time to external demands, and to react on events

Source: Op ’t Land, M.; Proper, E.; Waage, M.; Cloo, J. and Steghuis, C.: Enterprise Architecture - Creating Value by Informed Governance, Springer-Verlag 2009, page 6. http://www.springerlink.com/content/k8jp3r/#section=132347&page=2&locus=10

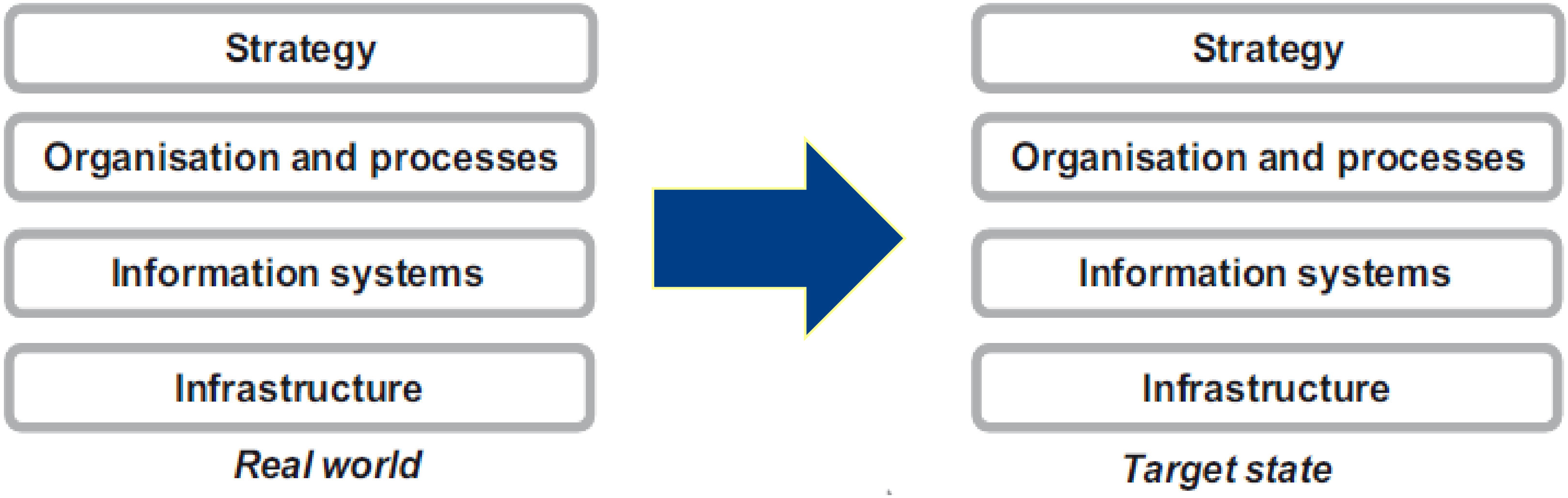
Agility


FROM NOW ON WE'RE GOING TO CHANGE HOW WE DO THINGS HERE. WE NEED TO BE AGILE.
WHAT DOES IT MEAN?
NOTHING REALLY CHANGES, EXCEPT THAT NOW MANAGERS, DESIGNERS AND ARCHITECTS HAVE AN EXCUSE FOR CHANGING THE REQUIREMENTS ON FRIDAYS AT 4 PM.

#113 - "AGILE DEVELOPMENT, EXAPLAINED" - BY SALVATORE IOVENE, FEB. 21ST 2009

HTTP://WWW.GEEKHEROCOMIC.COM/

Change Projects



Strategy

Organisation and processes

Information systems

Infrastructure

Real world

Strategy

Organisation and processes

Information systems

Infrastructure

Target state

Problems for Change in Today's Enterprises

In practice, enterprises see themselves hampered in their ability to change in several ways, which is a consequence of uncoordinated projects:

- ♦ being uninformed about their own products, services, capabilities, internal structures
- ♦ traditionally, organisations were designed with efficiency and effectiveness in mind rather than agility
- ♦ no common understanding and governance of key data resources
- ♦ a plethora of legacy applications and infrastructures
- ♦ duplicated functionality in terms of people and/or technology
- ♦ interwoven and unclear responsibilities
- ♦ organisational silos, self-contained business units who operate on their own, with no sharing of data
- ♦ silo applications, i.e. self-contained and isolated applciations, which only provide functionality to a specific business process

- ■ Solution: ***Enterprise Architecture***

Source: Op 't Land, M.; Proper, E.; Waage, M.; Cloo, J. and Steghuis, C.: Enterprise Architecture - Creating Value by Informed Governance, Springer-Verlag 2009, page 6. http://www.springerlink.com/content/k8jp3r/#section=132347&page=2&locus=10

Need for Enterprise Architecture Management : Transparency

- Many organisations lack transparency due to the number and frequency of their organisational changes.
- Some of the questions they cannot answer are
  - How can we successfully integrate new firms after an acquisition?
  - Can we introduce new products and services, using the existing business processes and the underlying applications?
  - Which business units and users will be affected by an application’s migration?
  - What applications and infrastructure technologies do we require to run new or redesigned business processes?

(Ahlemann et al 2012, p. 6)

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**".

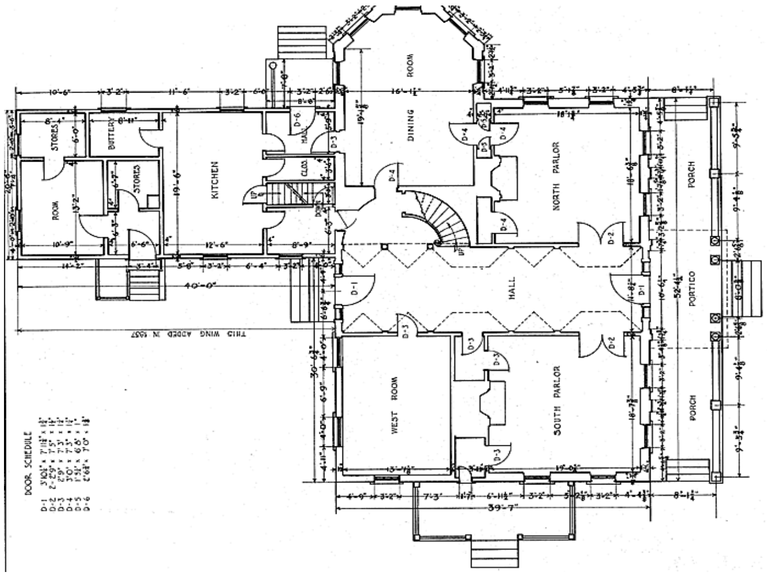
(John Zachmann, 2012)

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*

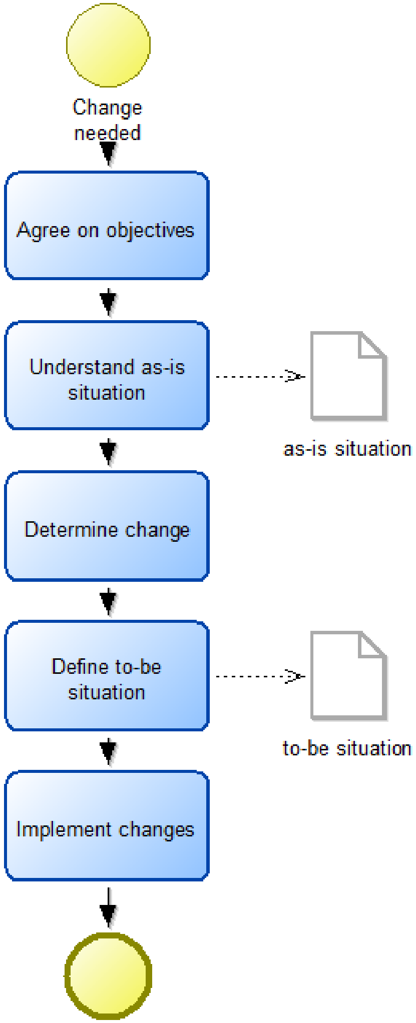
- **Business structures** (e.g. organisation structure, business processes)
- **IT structures** (e.g. information systems, infrastructure)
- and their **relationships**

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.

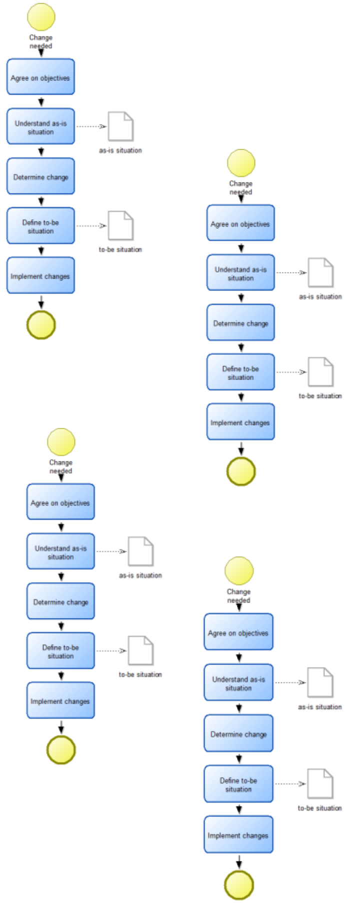
Adapted from Zachman (2012)

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

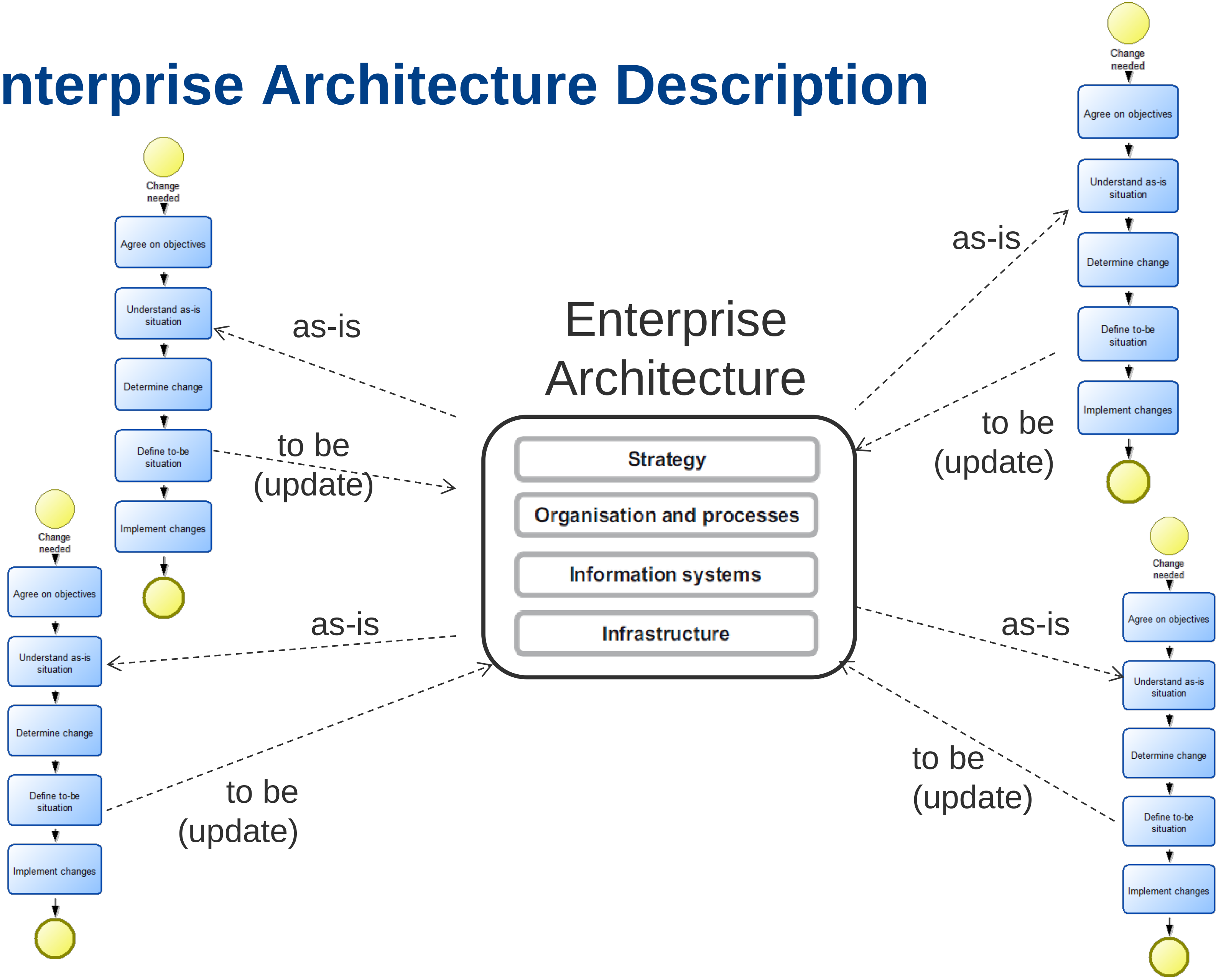
- running concurrently or
- building on the result of previous projects

… 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

Enterprise Architecture Description



Enterprise Architecture

Strategy

Organisation and processes

Information systems

Infrastructure

as-is

to be (update)

Change needed

Agree on objectives

Understand as-is situation

Determine change

Define to-be situation

Implement changes

The Need for Architecture Description

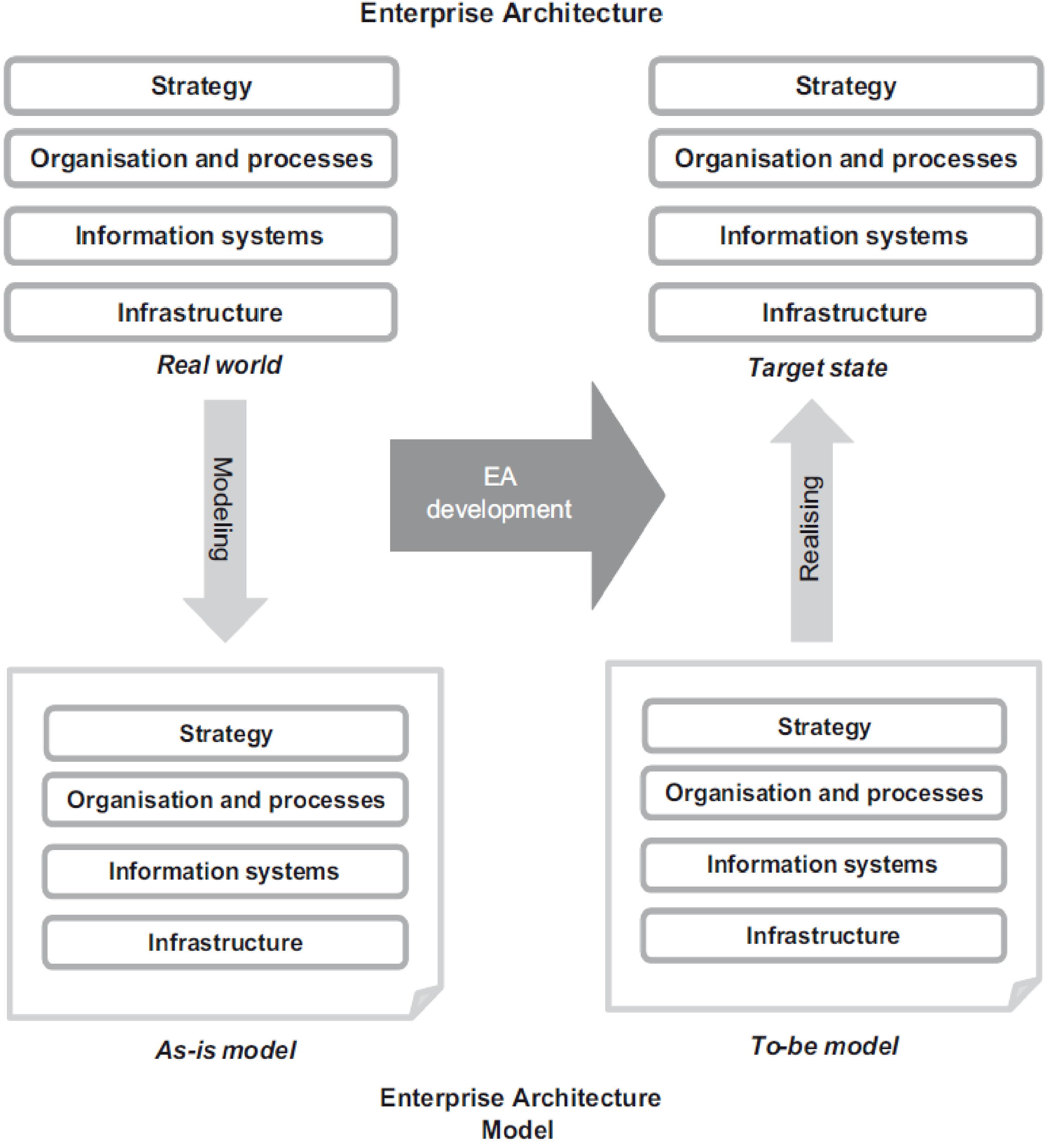
- **Complexity:** If you can't describe it, you can't create it (whatever "it" is).
- **Change:** If you don't retain the descriptive representations after you create them (or if you never created them in the first place) and you need to change the resultant implementation, you have only three options:
  - 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)

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 aligning 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

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

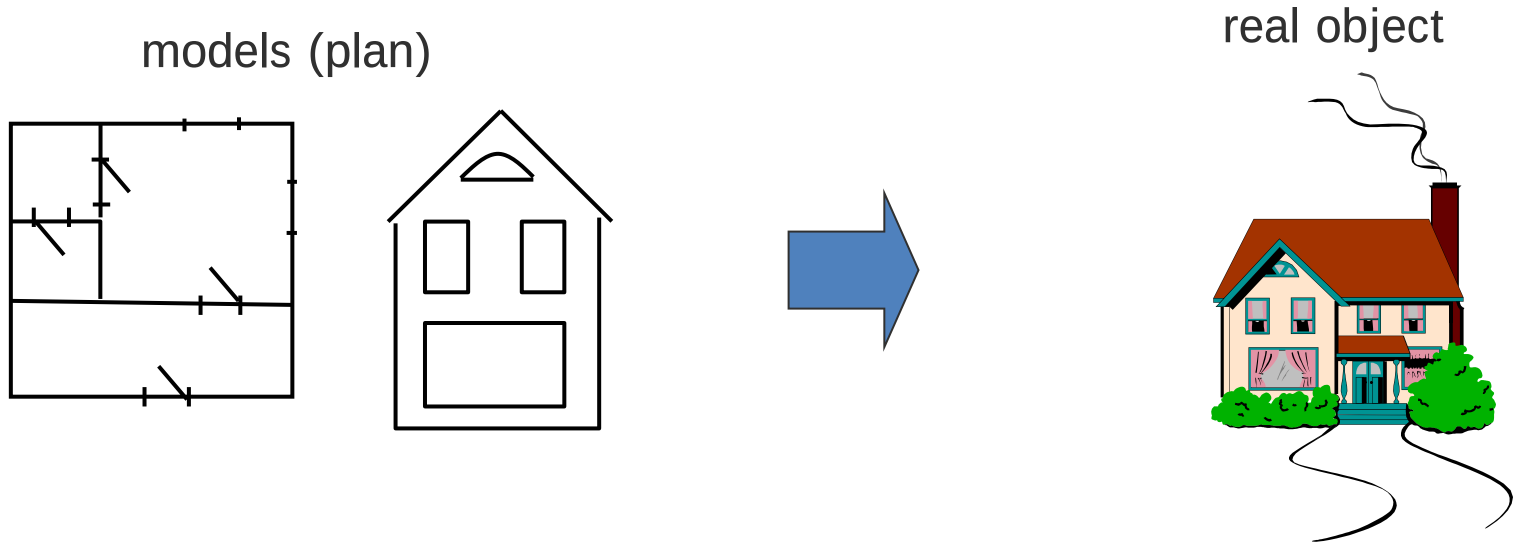
Architecture and Architecture Description

- 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/

Archtecture Description and Architecture Models

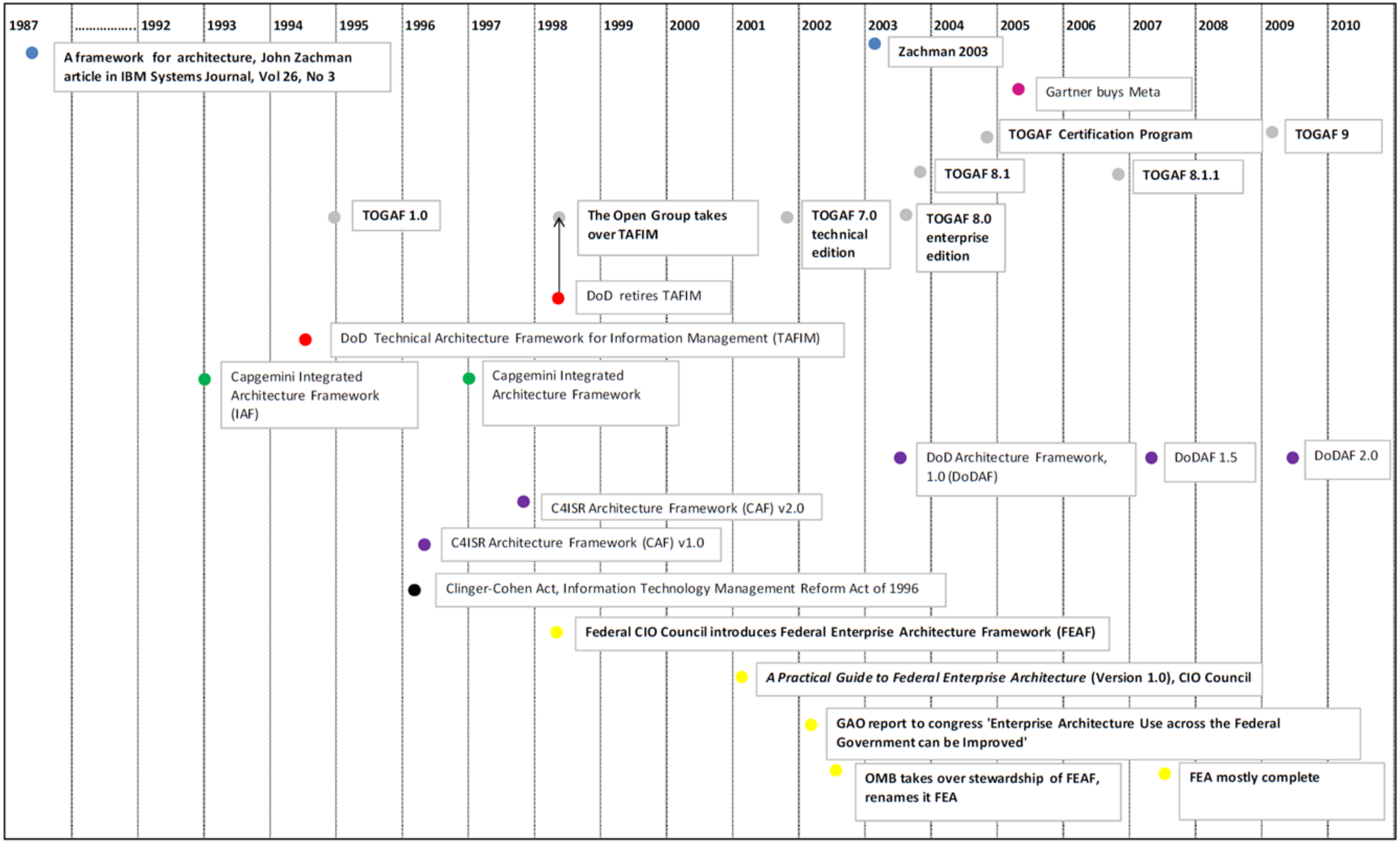
- An *Architecture Description* consists of one or several *Architecture Models*
- A Model is a reproduction of a *relevant* part of reality which contains the essential aspects to be investigated.



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



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

Zachman 2003

Gartner buys Meta

TOGAF Certification Program

TOGAF 9

TOGAF 8.1

TOGAF 8.1.1

TOGAF 1.0

The Open Group takes over TAFIM

TOGAF 7.0 technical edition

TOGAF 8.0 enterprise edition

DoD retires TAFIM

DoD Technical Architecture Framework for Information Management (TAFIM)

Capgemini Integrated Architecture Framework (IAF)

Capgemini Integrated Architecture Framework

DoD Architecture Framework, 1.0 (DoDAF)

DoDAF 1.5

DoDAF 2.0

C4ISR Architecture Framework (CAF) v2.0

C4ISR Architecture Framework (CAF) v1.0

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

Federal CIO Council introduces Federal Enterprise Architecture Framework (FEAF)

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

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

OMB takes over stewardship of FEAF, renames it FEA

FEA mostly complete

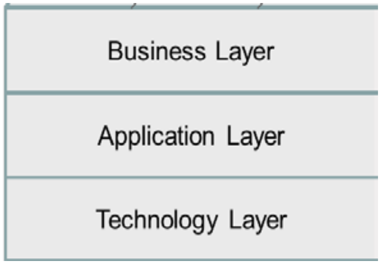
(Bespoke Systems 2012)

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

What
How
Where
Who
When
Why
Executive Perspective
Business Mgmt Perspective
Architect Perspective
Engineer Perspective
Technician Perspective
Enterprise Perspective
Scope Contexts
Business Concepts
System Logic
Technology Physics
Tool Components
Operations Instances
Inventory Sets
Process Flows
Distribution Networks
Responsibility Assignments
Timing Cycles
Motivation Intentions

  - ♦ ***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

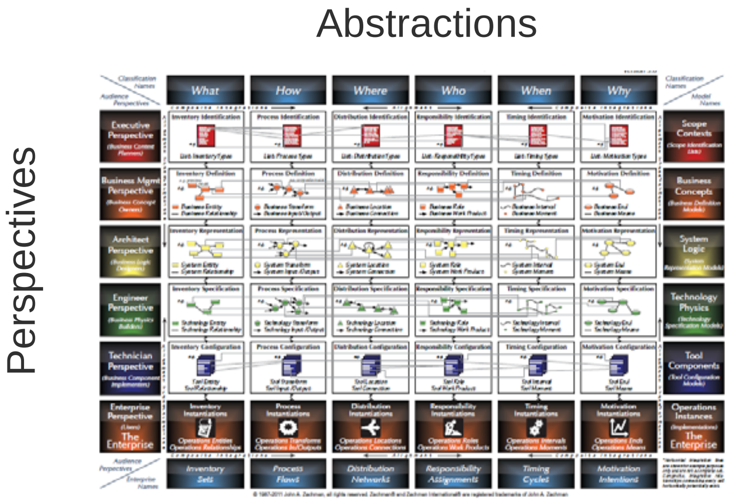
Zachman Framework

- 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

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



(Zachman 2012)

Dimension 1 – Perspectives

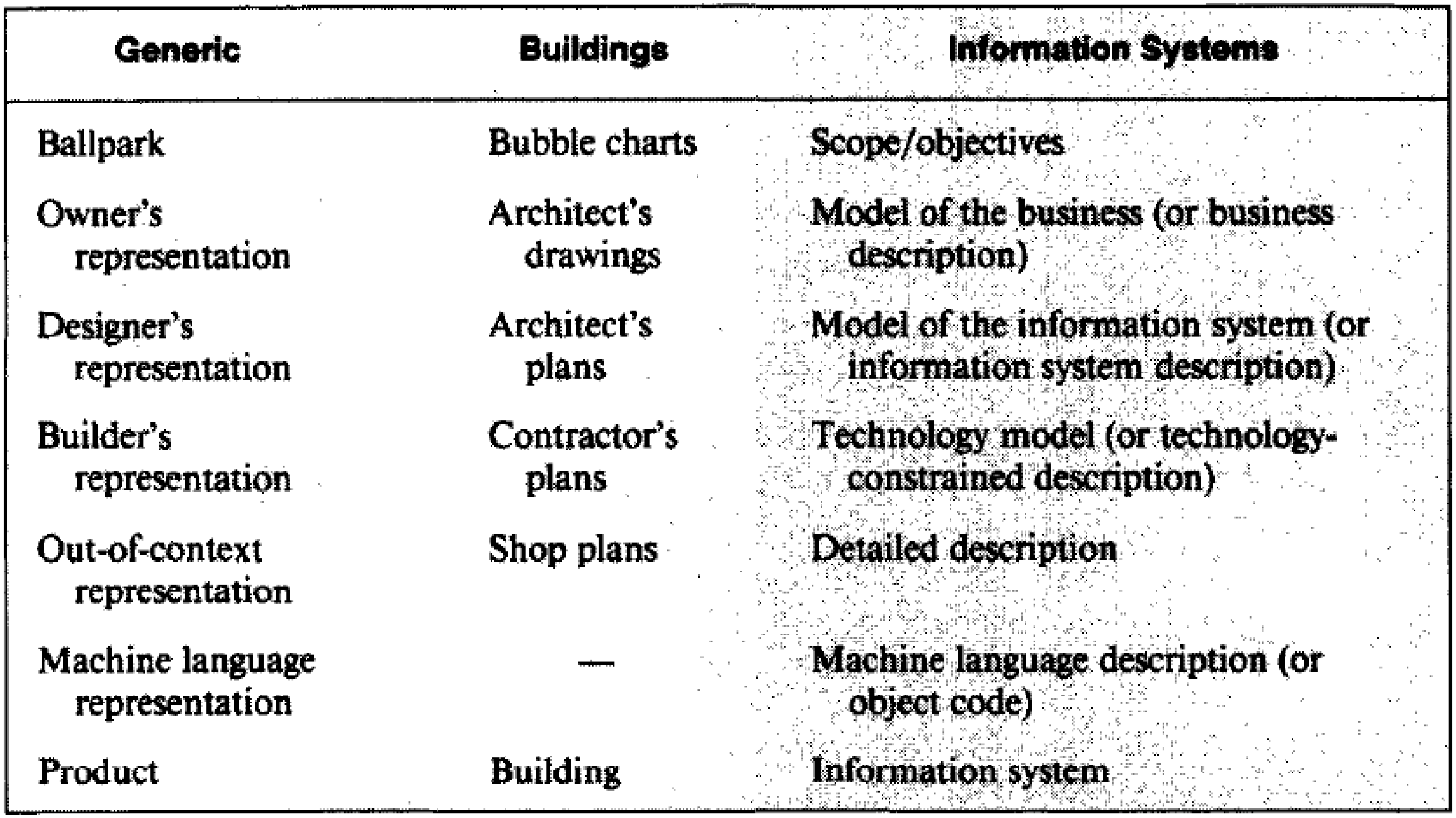
Zachman originally used the analogy of classical architecture

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)

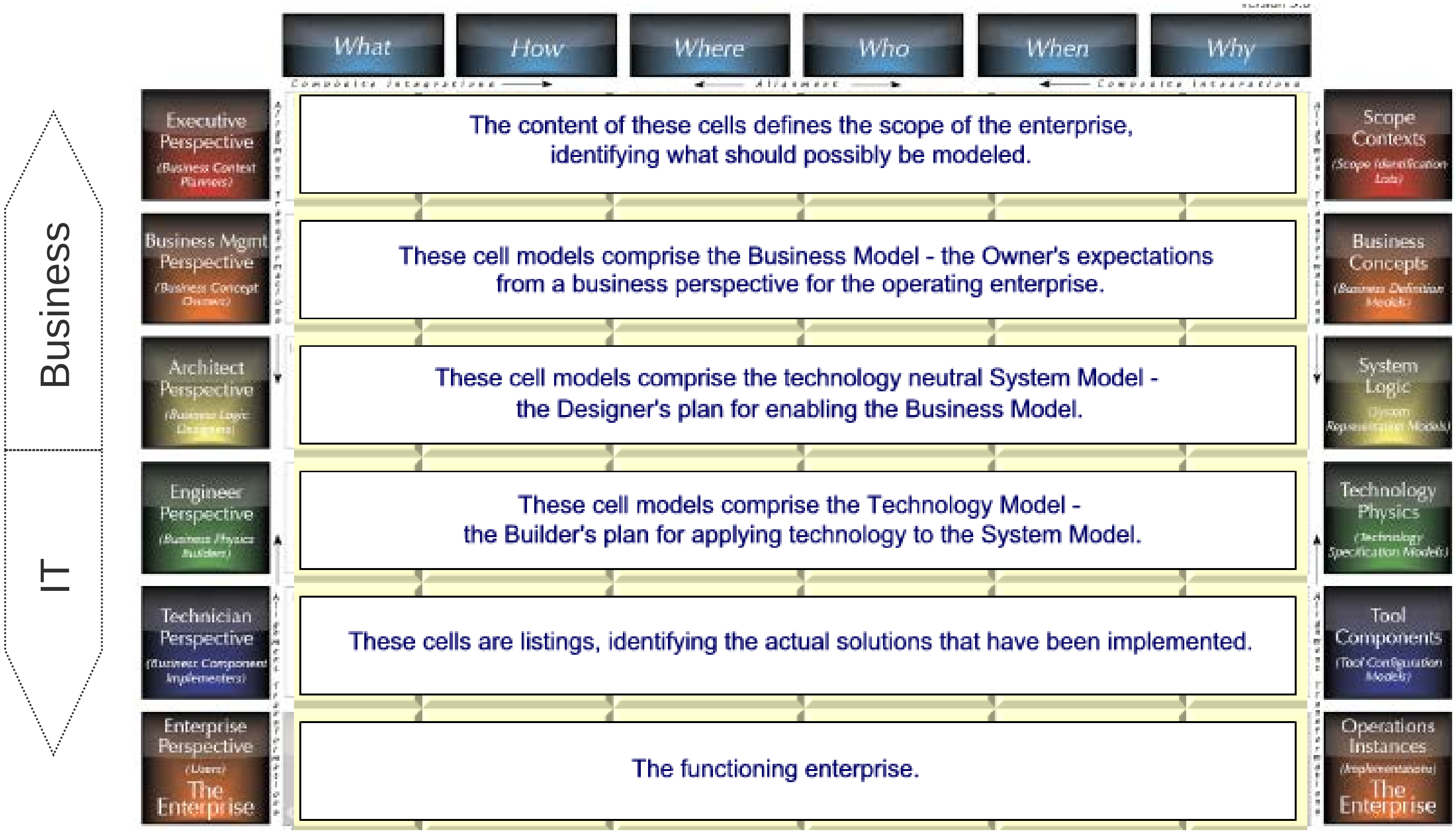
Dimension 1: Architectural Representations with analogies in Building and Information Systems



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


What
How
Where
Who
When
Why
Composite Integrations
Alignment
Composite Integrations
Business
IT
Executive Perspective (Business Context Planners)
Business Mgmt Perspective (Business Concept Owners)
Architect Perspective (Business Logic Designers)
Engineer Perspective (Business Physics Builders)
Technician Perspective (Business Component Implementers)
Enterprise Perspective (Users) The Enterprise
Scope Contexts (Scope Identification Lists)
Business Concepts (Business Definition Models)
System Logic (System Representation Models)
Technology Physics (Technology Specification Models)
Tool Components (Tool Configuration Models)
Operations Instances (Implementations) The 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.

Scope (Boundaries)

Requirements (Concepts)

Design (Logic)

Plan (Physics)

Part (Configurations)

Product (Instances)

- 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

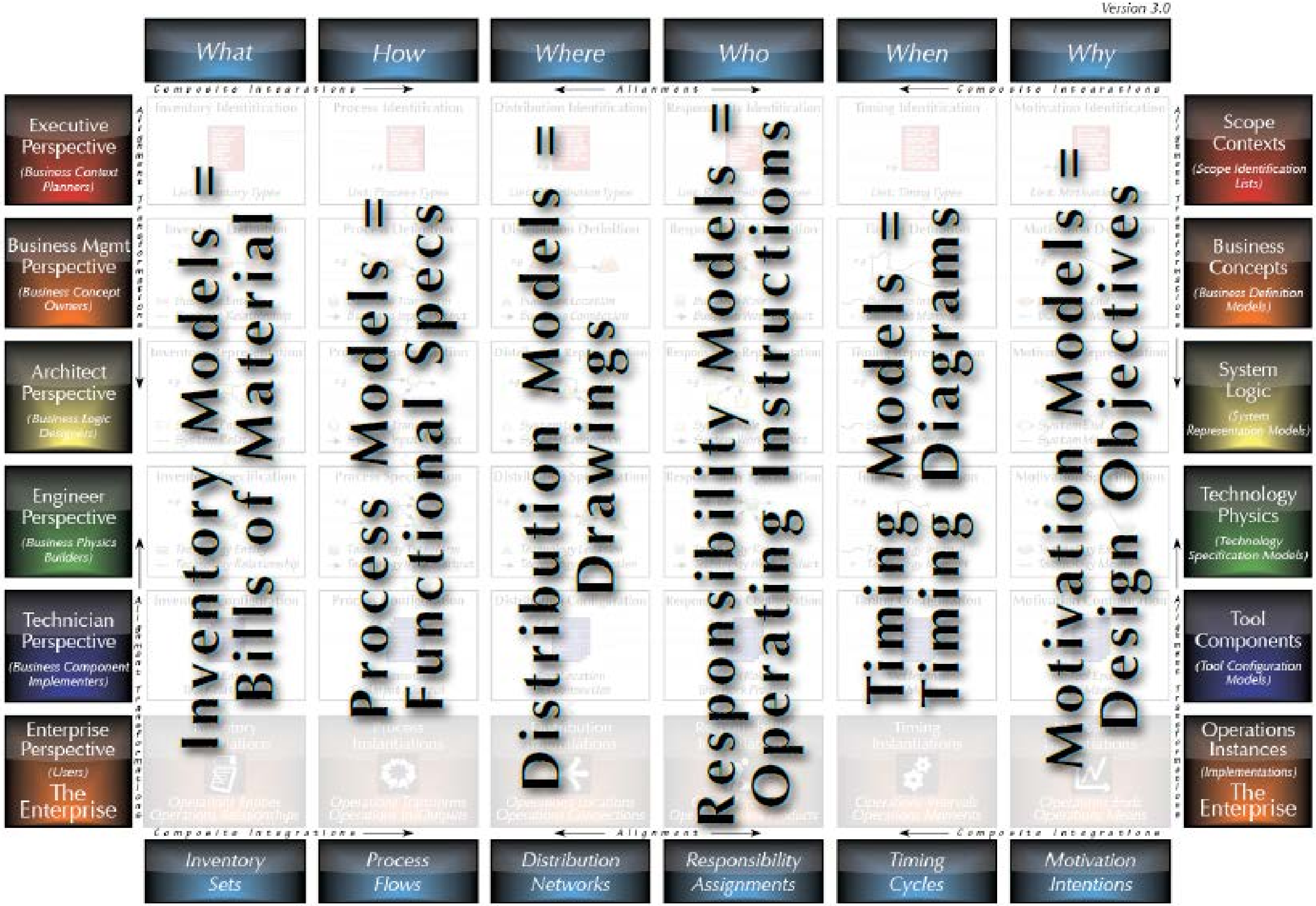
Dimension 2: Aspects of an Architecture

- 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


Version 3.0
What
How
Where
Who
When
Why
Executive Perspective (Business Context Planners)
Business Mgmt Perspective (Business Concept Owners)
Architect Perspective (Business Logic Designers)
Engineer Perspective (Business Physics Builders)
Technician Perspective (Business Component Implementers)
Enterprise Perspective (Users)
The Enterprise
Scope Contexts (Scope Identification Lists)
Business Concepts (Business Definition Models)
System Logic (System Representation Models)
Technology Physics (Technology Specification Models)
Tool Components (Tool Configuration Models)
Operations Instances (Implementations)
The Enterprise
Composite Integrations
Alignment
Composite Integrations
Inventory Sets
Process Flows
Distribution Networks
Responsibility Assignments
Timing Cycles
Motivation Intentions
Inventory Models = Bills of Material
Process Models = Functional Specs
Distribution Models = Drawings
Responsibility Models = Operating Instructions
Timing Models = Timing Diagrams
Motivation Models = Design Objectives

(Zachman 2012)

The Zachman Framework for Enterprise Architecture – Enterprise Ontology

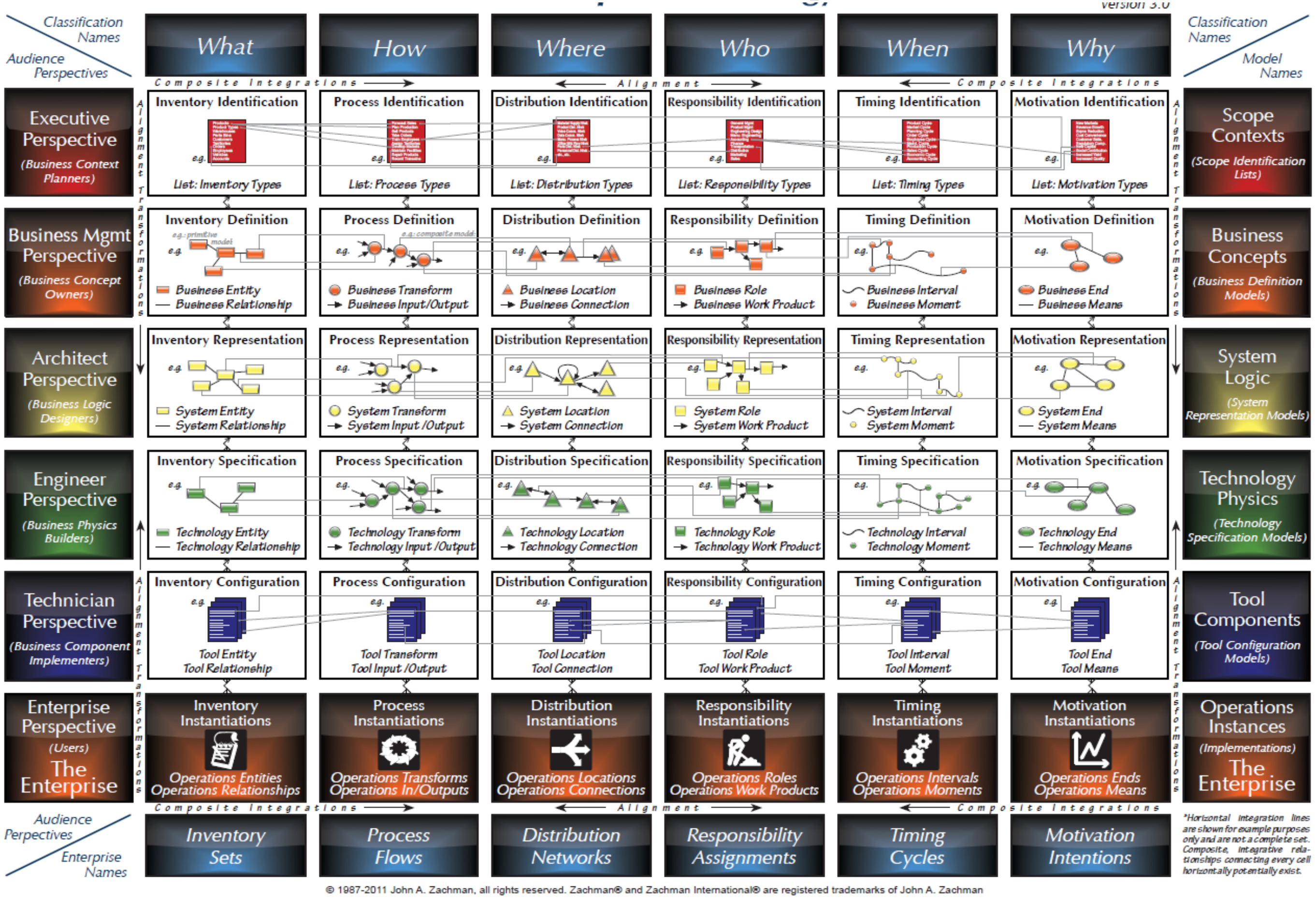
Each cell contains models

Abstractions/Aspects

Perspectives

Business

Information Technology



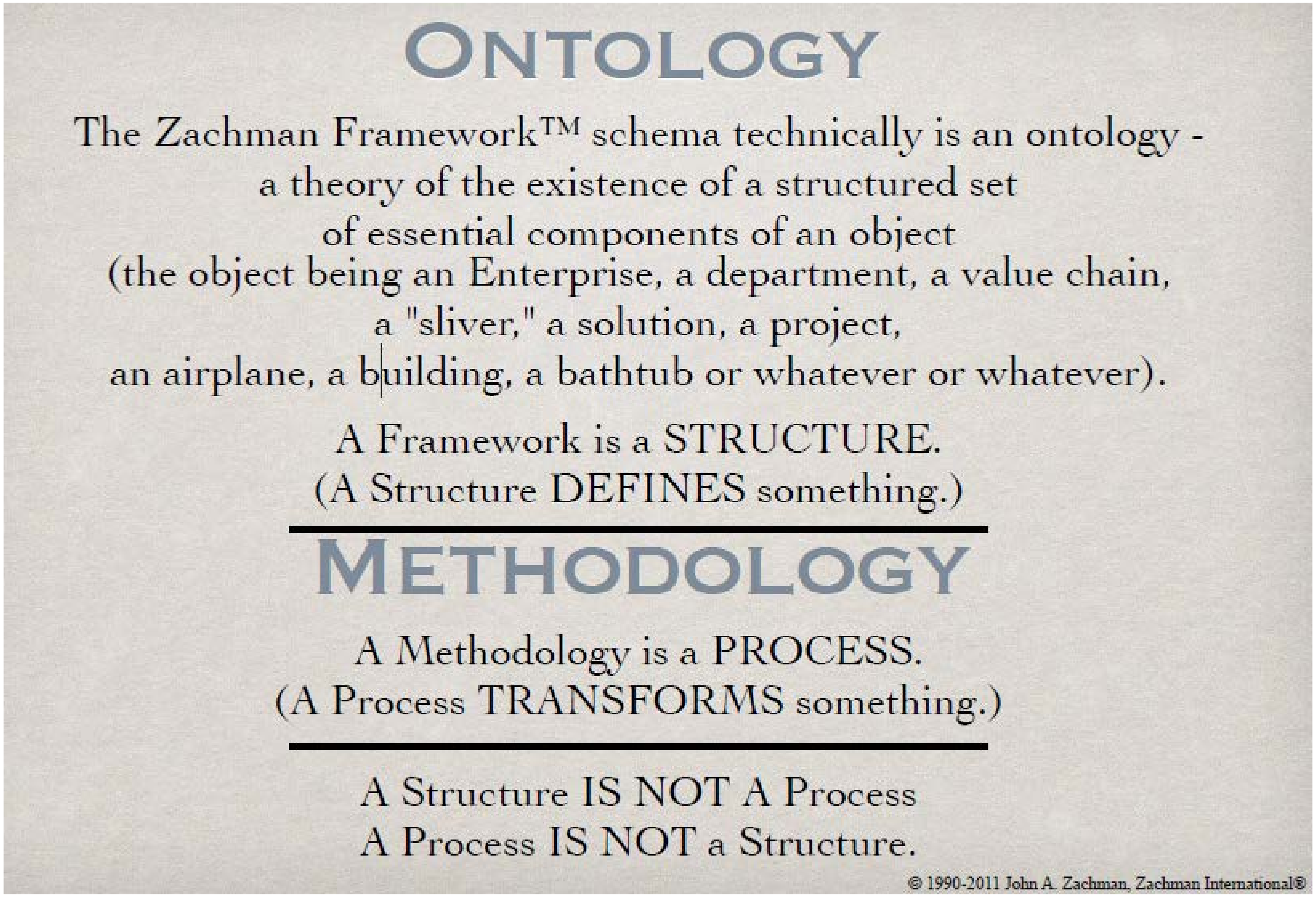
version 3.0

Composite Integrations → ← Alignment → ← Composite Integrations

Alignment Transformations

Classification Names / Audience Perspectives	What	How	Where	Who	When	Why	Classification Names / Model Names
Executive Perspective (Business Context Planners)	Inventory Identification e.g. List: Inventory Types	Process Identification e.g. List: Process Types	Distribution Identification e.g. List: Distribution Types	Responsibility Identification e.g. List: Responsibility Types	Timing Identification e.g. List: Timing Types	Motivation Identification e.g. List: Motivation Types	Scope Contexts (Scope Identification Lists)
Business Mgmt Perspective (Business Concept Owners)	Inventory Definition e.g.: primitive model e.g. Business Entity — Business Relationship	Process Definition e.g.: composite model: e.g. ● Business Transform → Business Input/Output	Distribution Definition e.g. ▲ Business Location → Business Connection	Responsibility Definition e.g. ■ Business Role → Business Work Product	Timing Definition e.g. ~ Business Interval ● Business Moment	Motivation Definition e.g. ● Business End — Business Means	Business Concepts (Business Definition Models)
Architect Perspective (Business Logic Designers)	Inventory Representation e.g. System Entity — System Relationship	Process Representation e.g. ○ System Transform → System Input /Output	Distribution Representation e.g. △ System Location → System Connection	Responsibility Representation e.g. □ System Role → System Work Product	Timing Representation e.g. ~ System Interval ○ System Moment	Motivation Representation e.g. ○ System End — System Means	System Logic (System Representation Models)
Engineer Perspective (Business Physics Builders)	Inventory Specification e.g. Technology Entity — Technology Relationship	Process Specification e.g. ● Technology Transform → Technology Input /Output	Distribution Specification e.g. ▲ Technology Location → Technology Connection	Responsibility Specification e.g. ■ Technology Role → Technology Work Product	Timing Specification e.g. ~ Technology Interval ● Technology Moment	Motivation Specification e.g. ● Technology End — Technology Means	Technology Physics (Technology Specification Models)
Technician Perspective (Business Component Implementers)	Inventory Configuration e.g. Tool Entity Tool Relationship	Process Configuration e.g. Tool Transform Tool Input /Output	Distribution Configuration e.g. Tool Location Tool Connection	Responsibility Configuration e.g. Tool Role Tool Work Product	Timing Configuration e.g. Tool Interval Tool Moment	Motivation Configuration e.g. Tool End Tool Means	Tool Components (Tool Configuration Models)
Enterprise Perspective (Users) The Enterprise	Inventory Instantiations Operations Entities Operations Relationships	Process Instantiations Operations Transforms Operations In/Outputs	Distribution Instantiations Operations Locations Operations Connections	Responsibility Instantiations Operations Roles Operations Work Products	Timing Instantiations Operations Intervals Operations Moments	Motivation Instantiations Operations Ends Operations Means	Operations Instances (Implementations) The Enterprise
Audience Perpectives / Enterprise Names	Inventory Sets	Process Flows	Distribution Networks	Responsibility Assignments	Timing Cycles	Motivation Intentions	*Horizontal integration lines are shown for example purposes only and are not a complete set. Composite, integrative relationships connecting every cell horizontally potentially exist.

The Zachman Framework is not a Methodology



ONTOLOGY

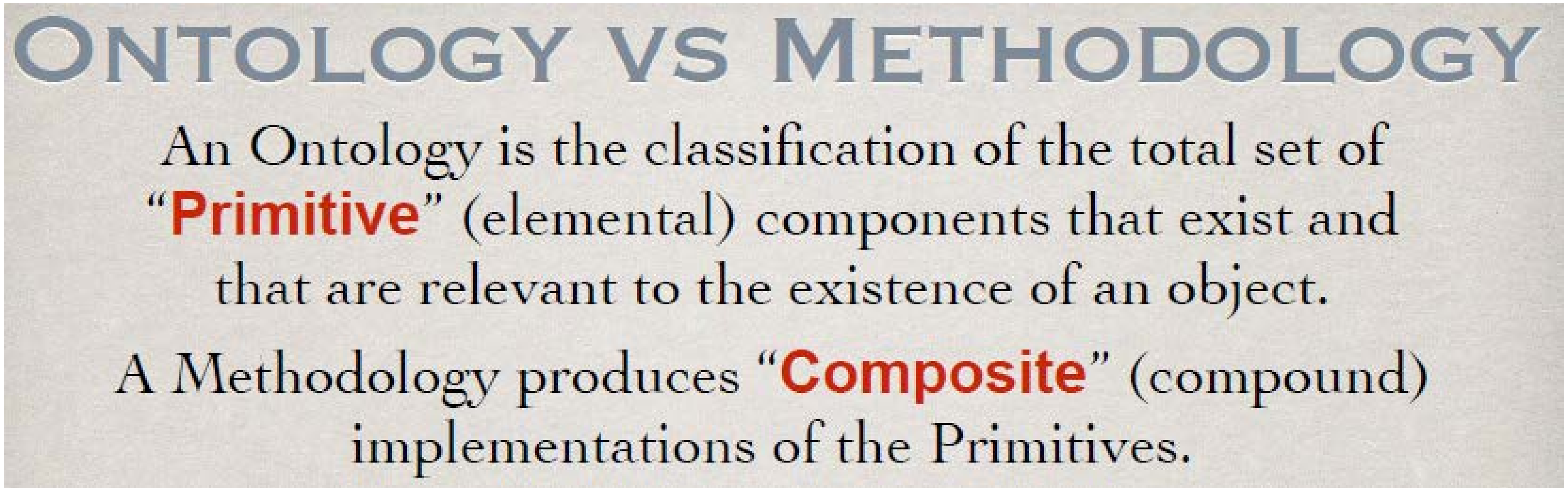
The Zachman Framework™ schema technically is an ontology -
a theory of the existence of a structured set
of essential components of an object
(the object being an Enterprise, a department, a value chain,
a "sliver," a solution, a project,
an airplane, a building, a bathtub or whatever or whatever).

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

METHODOLOGY

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

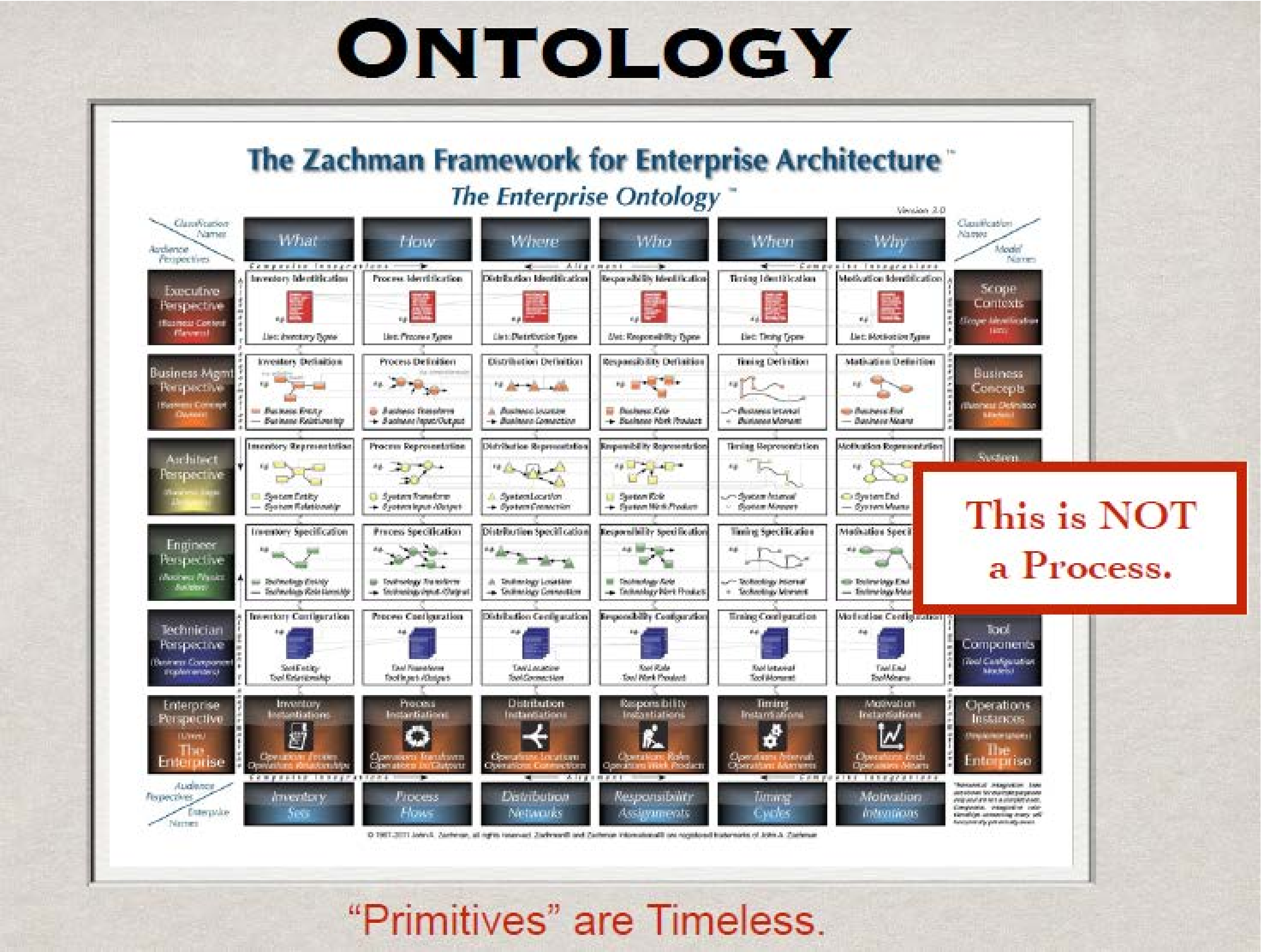
A Structure IS NOT A Process
A Process IS NOT a Structure.



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 ™

Version 3.0

Composite Integrations → ← Alignment → ← Composite Integrations

Classification Names / Audience Perspectives	What	How	Where	Who	When	Why	Classification Names / Model Names
Executive Perspective (Business Context Planners)	Inventory Identification e.g. List: Inventory Types	Process Identification e.g. List: Process Types	Distribution Identification e.g. List: Distribution Types	Responsibility Identification e.g. List: Responsibility Types	Timing Identification e.g. List: Timing Types	Motivation Identification e.g. List: Motivation Types	Scope Contexts (Scope Identification Lists)
Business Mgmt Perspective (Business Concept Owners)	Inventory Definition e.g. Business Entity Business Relationship	Process Definition e.g. Business Transform Business Input/Output	Distribution Definition e.g. Business Location Business Connection	Responsibility Definition e.g. Business Role Business Work Product	Timing Definition e.g. Business Interval Business Moment	Motivation Definition e.g. Business End Business Means	Business Concepts (Business Definition Models)
Architect Perspective (Business Logic Designers)	Inventory Representation e.g. System Entity System Relationship	Process Representation e.g. System Transform System Input/Output	Distribution Representation e.g. System Location System Connection	Responsibility Representation e.g. System Role System Work Product	Timing Representation e.g. System Interval System Moment	Motivation Representation e.g. System End System Means	System
Engineer Perspective (Business Physics Builders)	Inventory Specification e.g. Technology Entity Technology Relationship	Process Specification e.g. Technology Transform Technology Input/Output	Distribution Specification e.g. Technology Location Technology Connection	Responsibility Specification e.g. Technology Role Technology Work Product	Timing Specification e.g. Technology Interval Technology Moment	Motivation Speci e.g. Technology End Technology Mea	
Technician Perspective (Business Component Implementers)	Inventory Configuration e.g. Tool Entity Tool Relationship	Process Configuration e.g. Tool Transform Tool Input/Output	Distribution Configuration e.g. Tool Location Tool Connection	Responsibility Configuration e.g. Tool Role Tool Work Product	Timing Configuration e.g. Tool Interval Tool Moment	Motivation Configuration e.g. Tool End Tool Means	Tool Components (Tool Configuration Models)
Enterprise Perspective (Users) The Enterprise	Inventory Instantiations Operations Entities Operations Relationships	Process Instantiations Operations Transforms Operations In/Outputs	Distribution Instantiations Operations Locations Operations Connections	Responsibility Instantiations Operations Roles Operations Work Products	Timing Instantiations Operations Intervals Operations Moments	Motivation Instantiations Operations Ends Operations Means	Operations Instances (Implementations) The Enterprise
Audience Perspectives / Enterprise Names	Inventory Sets	Process Flows	Distribution Networks	Responsibility Assignments	Timing Cycles	Motivation Intentions	

Composite Integrations → ← Alignment → ← Composite Integrations

*Horizontal integration lines are shown for example purposes only and are not a complete set. Composite, integrative relationships connecting every cell horizontally potentially exist.

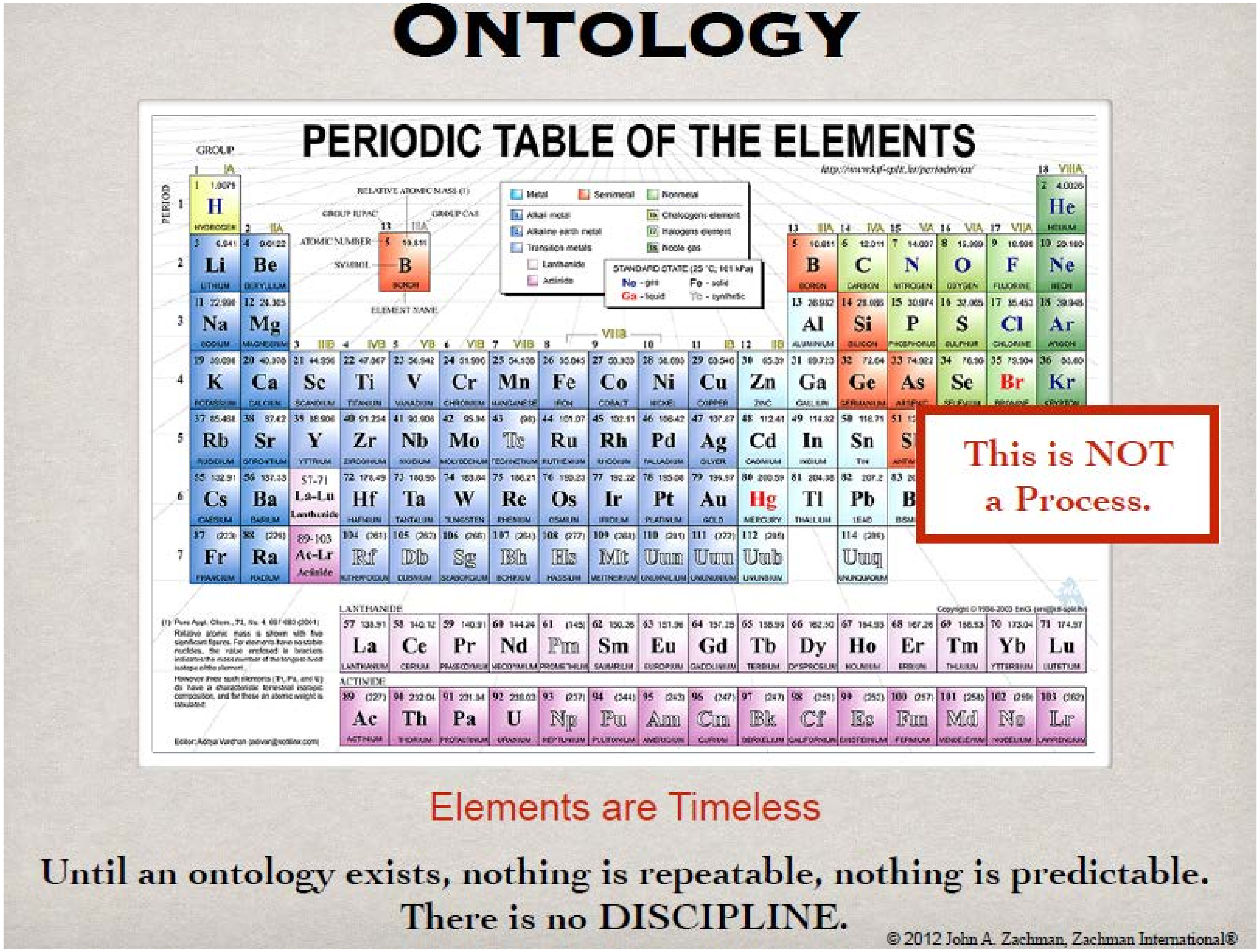
© 1987-2011 John A. Zachman, all rights reserved. Zachman® and Zachman International® are registered trademarks of John A. Zachman

This is NOT a Process.

“Primitives” are Timeless.

© 2012 John A. Zachman

Analogy: Chemistry



ONTOLOGY

This is NOT a Process.

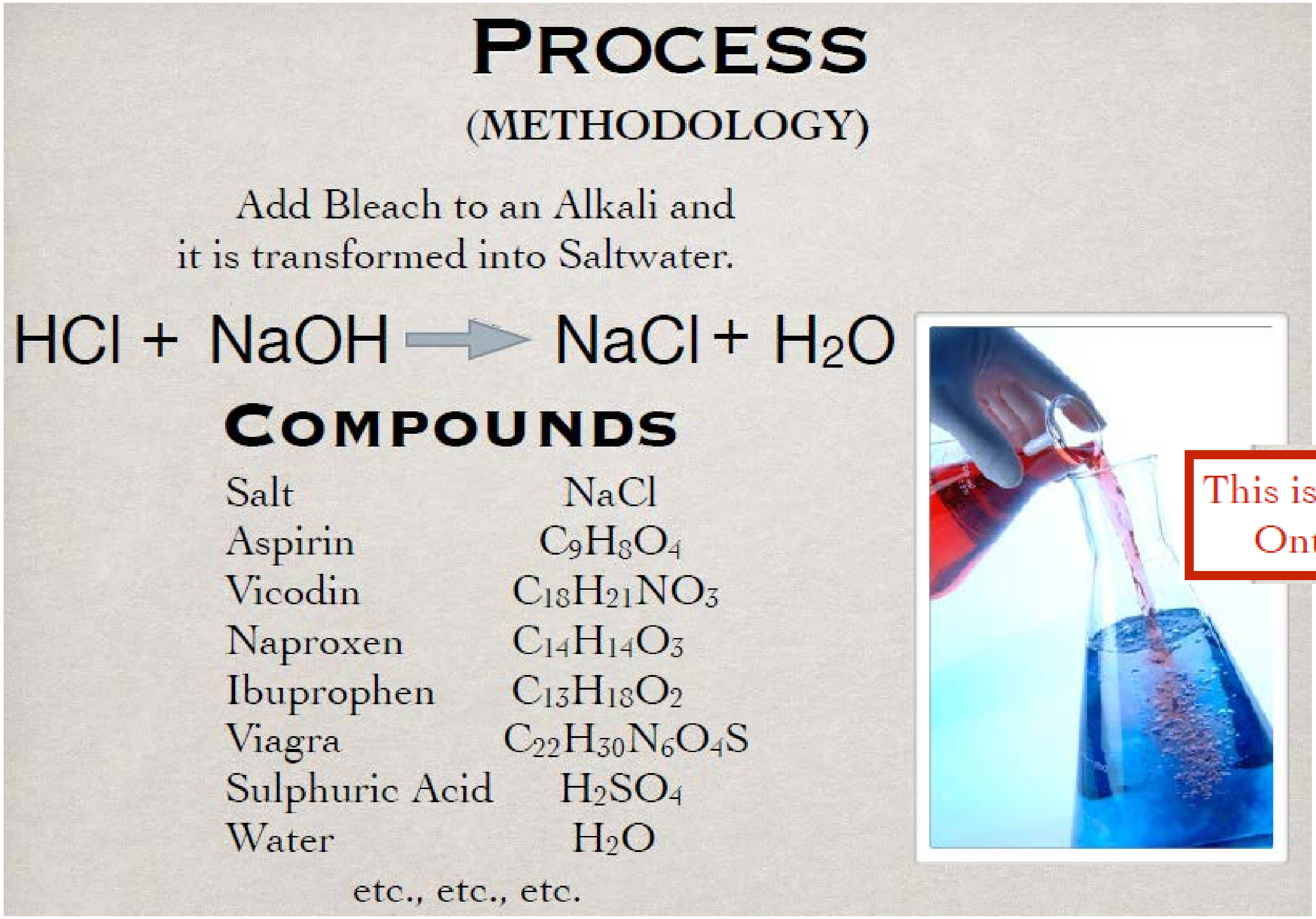
Elements are Timeless

Until an ontology exists, nothing is repeatable, nothing is predictable.
There is no DISCIPLINE.

© 2012 John A. Zachman, Zachman International®

© 2012 John A. Zachman

Analogy: Chemistry



PROCESS

(METHODOLOGY)

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

$$HCl + NaOH \longrightarrow NaCl + H_2O$$

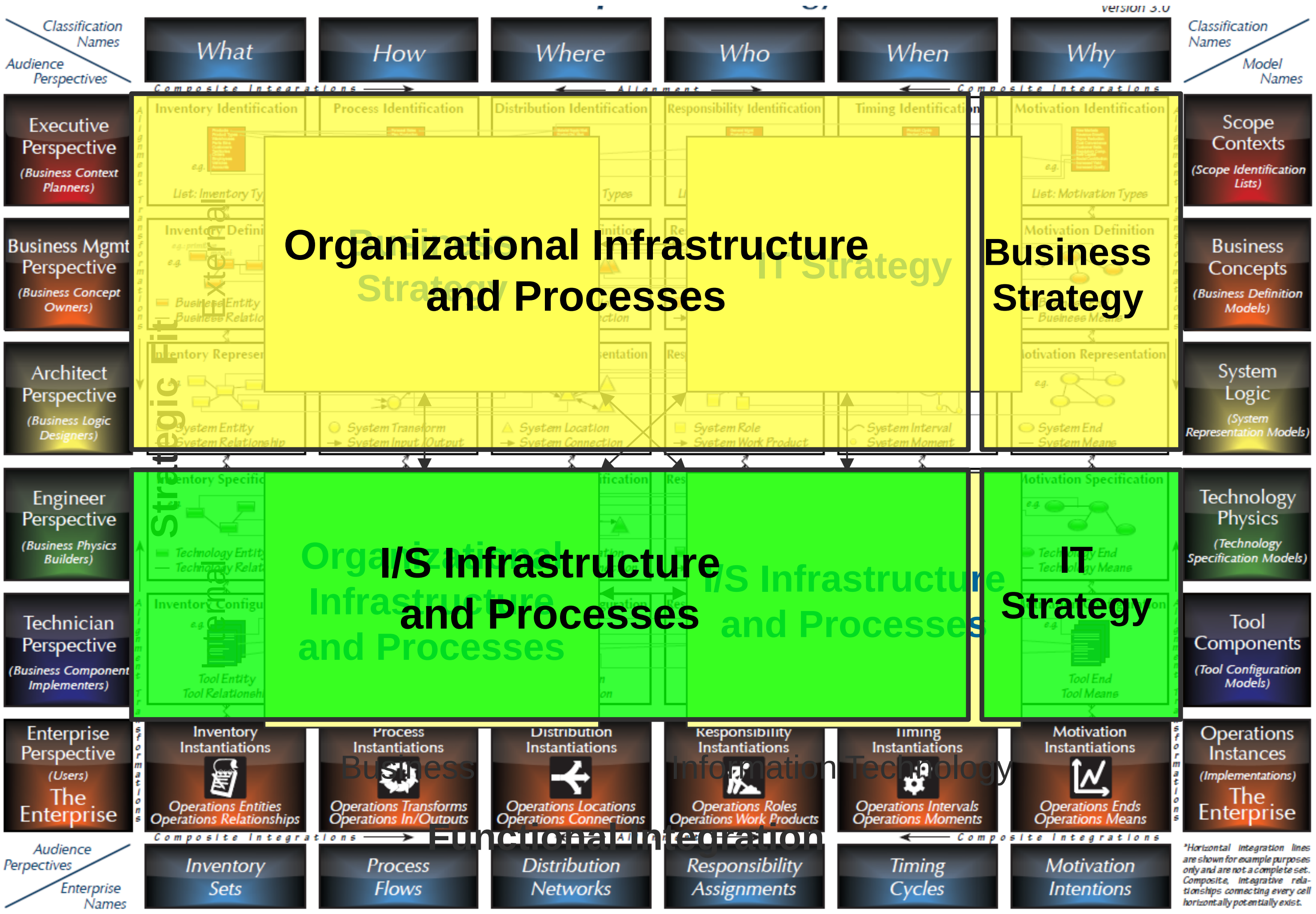
COMPOUNDS

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.

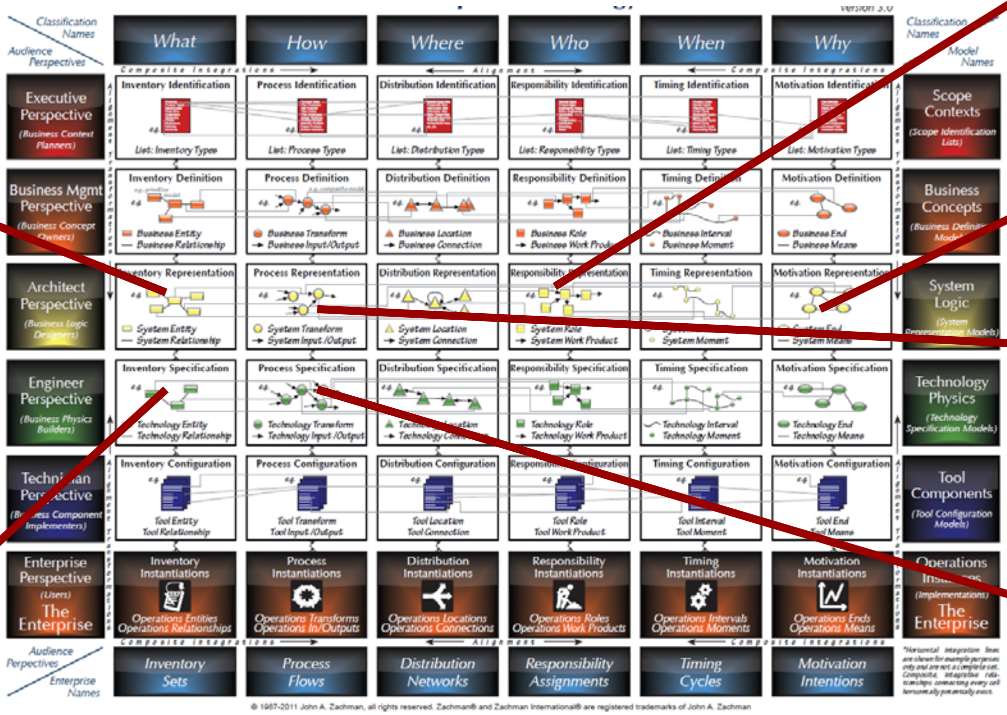
Strategic Alignment Model and Zachman Framework



© 1987-2011 John A. Zachman, all rights reserved. Zachman® and Zachman International® are registered trademarks of John A. Zachman

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

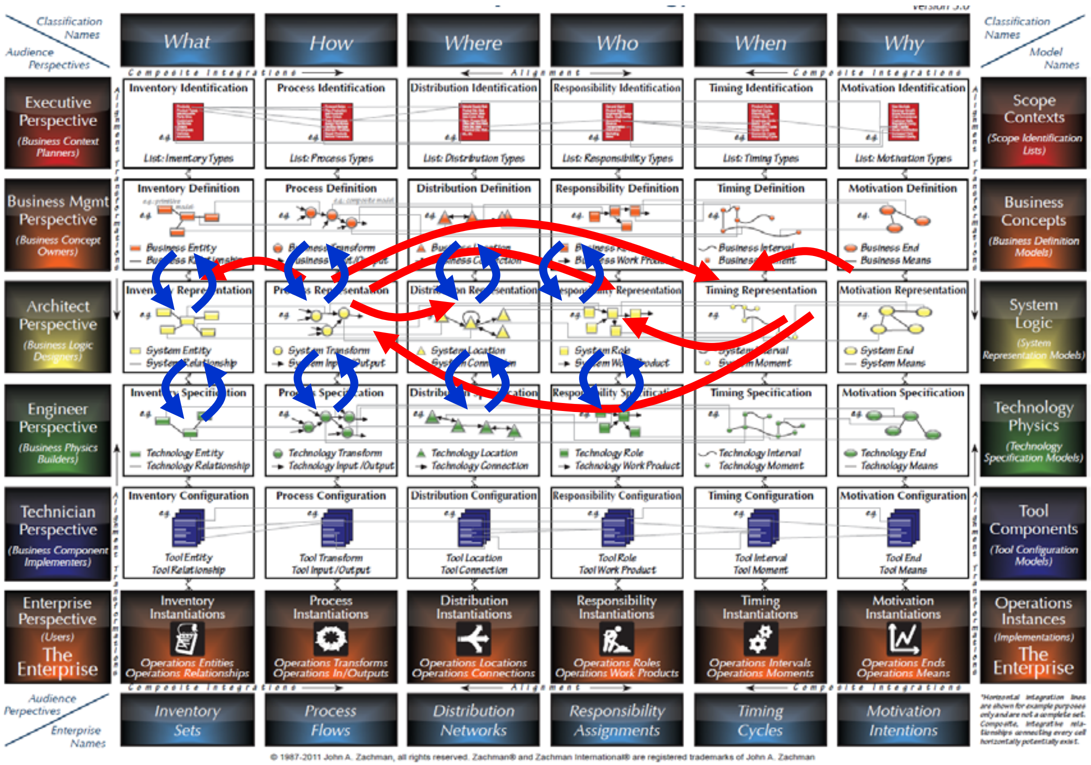
Motivation Model

Process model

Physical data model

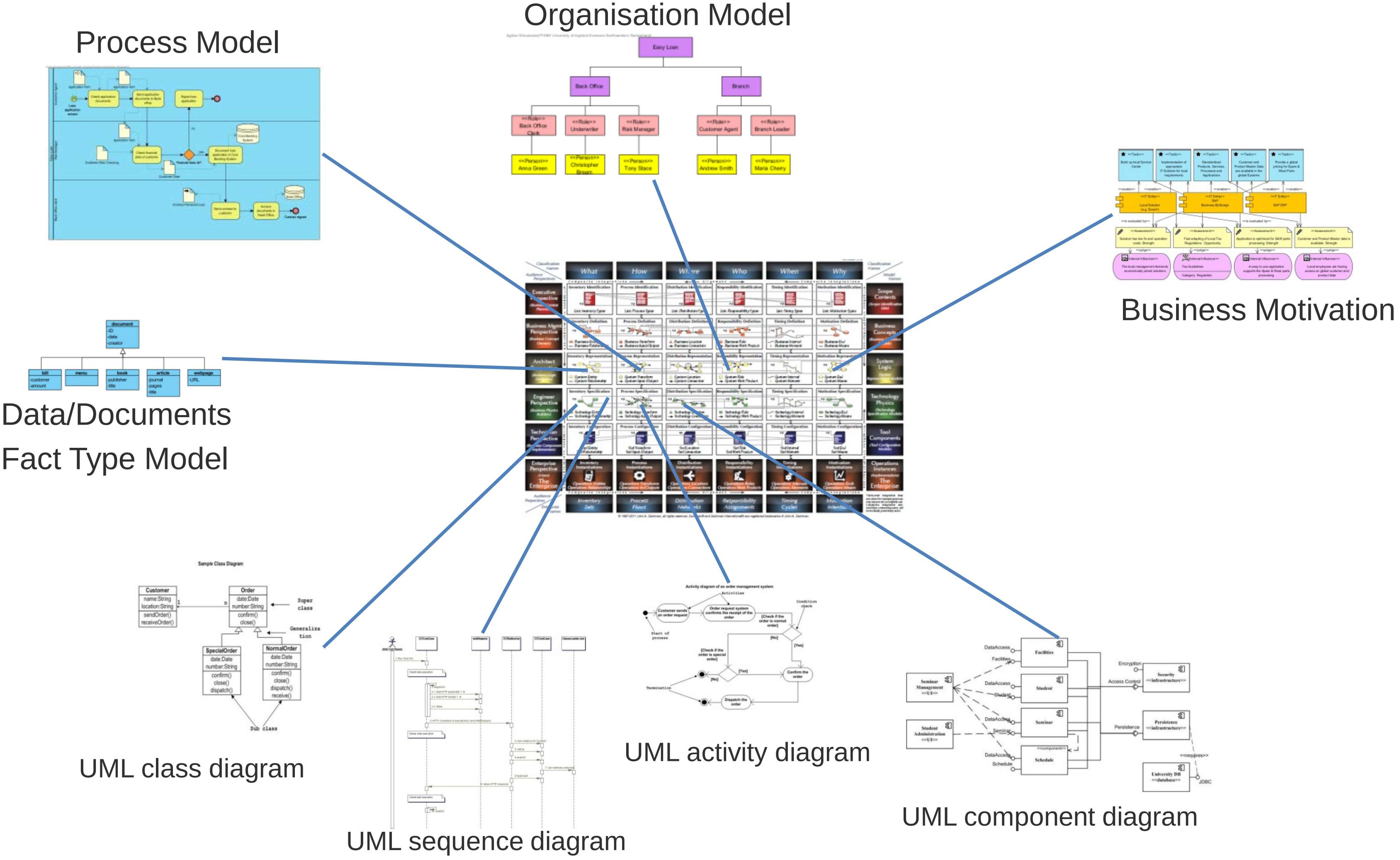
Workflow model

Relations between Models and Model Elements


Classification Names
Audience Perspectives
What
How
Where
Who
When
Why
Model Names
Executive Perspective (Business Context Planners)
Business Mgmt Perspective (Business Concept Owners)
Architect Perspective (Business Logic Designers)
Engineer Perspective (Business Physics Builders)
Technician Perspective (Business Component Implementers)
Enterprise Perspective (Users) The Enterprise
Scope Contexts (Scope Identification Lists)
Business Concepts (Business Definition Models)
System Logic (System Representation Models)
Technology Physics (Technology Specification Models)
Tool Components (Tool Configuration Models)
Operations Instances (Implementations) The Enterprise
Inventory Identification
Process Identification
Distribution Identification
Responsibility Identification
Timing Identification
Motivation Identification
Inventory Definition
Process Definition
Distribution Definition
Responsibility Definition
Timing Definition
Motivation Definition
Inventory Representation
Process Representation
Distribution Representation
Responsibility Representation
Timing Representation
Motivation Representation
Inventory Specification
Process Specification
Distribution Specification
Responsibility Specification
Timing Specification
Motivation Specification
Inventory Configuration
Process Configuration
Distribution Configuration
Responsibility Configuration
Timing Configuration
Motivation Configuration
Inventory Instantiations
Process Instantiations
Distribution Instantiations
Responsibility Instantiations
Timing Instantiations
Motivation Instantiations
Inventory Sets
Process Flows
Distribution Networks
Responsibility Assignments
Timing Cycles
Motivation Intentions
Composite Integrations
Alignment
© 1987-2011 John A. Zachman, all rights reserved. Zachman® and Zachman International® are registered trademarks of John A. Zachman

- 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



Organisation Model

Process Model

Business Motivation

Data/Documents Fact Type Model

UML class diagram

UML activity diagram

UML component diagram

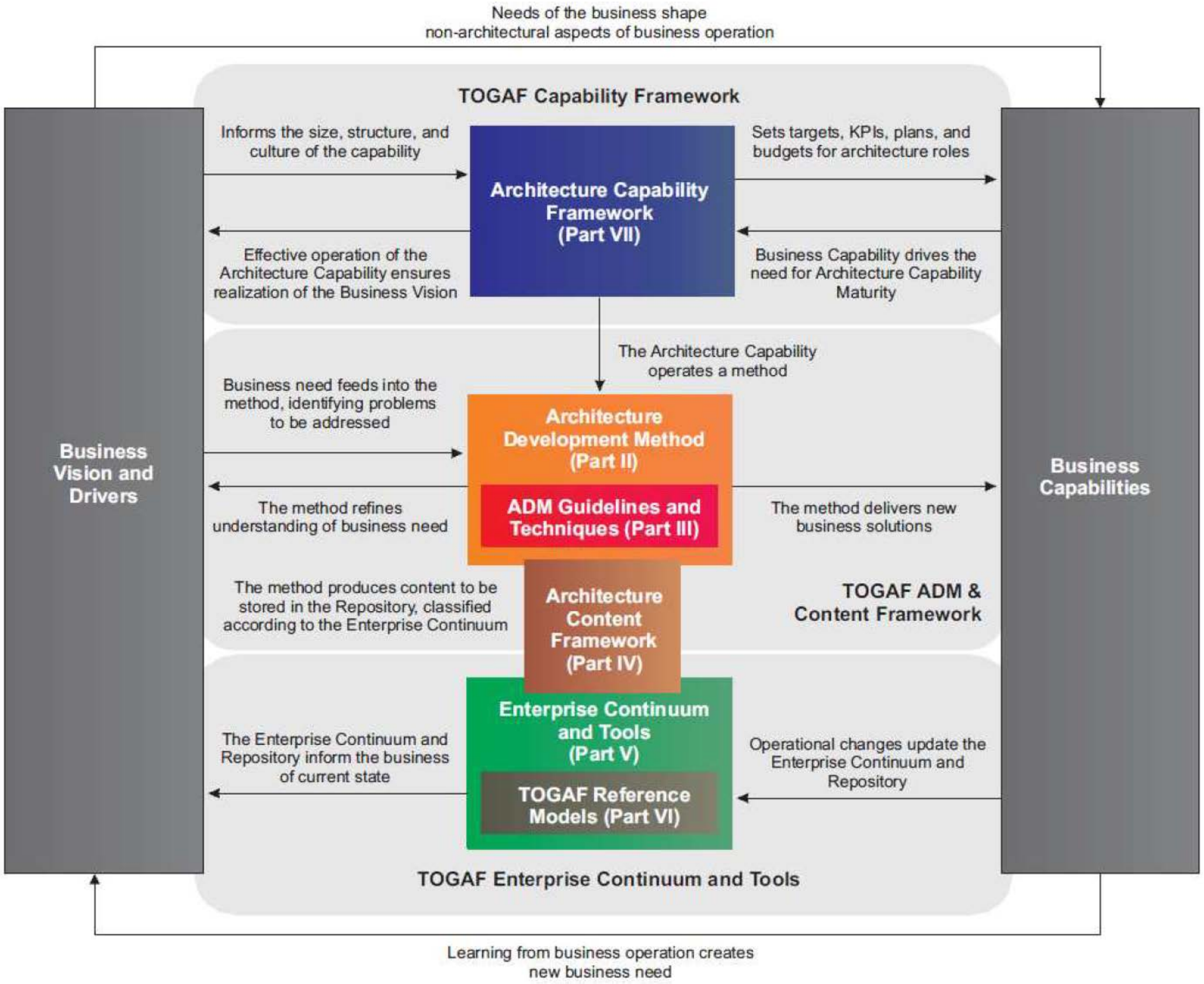
UML sequence diagram

TOGAF – The Open Group Architecture Framework

TOGAF – The Open Group Architecture Framework

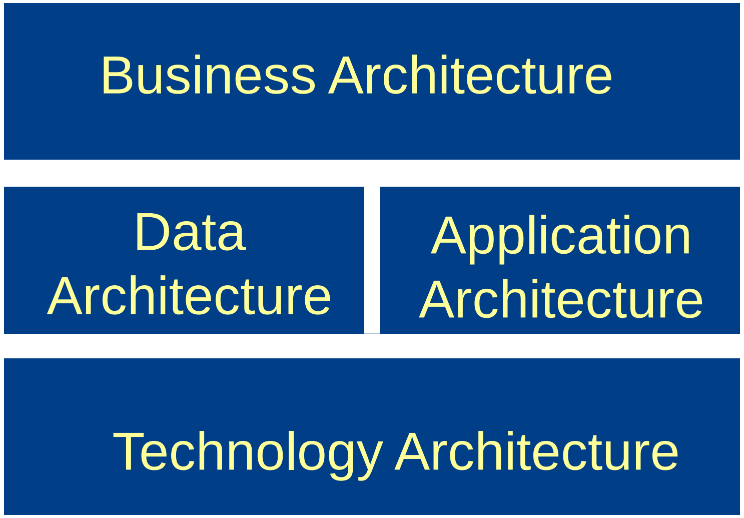
- Developed and continuously evolved since the mid-90’s by The Open Group’s Architecture Forum
- While Zachman is more an ontology, TOGAF is a methodology
- At the heart of the framework is the Architecture Development Method (ADM)
- http://www.opengroup.org/togaf/

Structure of the TOGAF Document


Needs of the business shape non-architectural aspects of business operation
TOGAF Capability Framework
Informs the size, structure, and culture of the capability
Architecture Capability Framework (Part VII)
Sets targets, KPIs, plans, and budgets for architecture roles
Effective operation of the Architecture Capability ensures realization of the Business Vision
Business Capability drives the need for Architecture Capability Maturity
The Architecture Capability operates a method
Business need feeds into the method, identifying problems to be addressed
Architecture Development Method (Part II)
ADM Guidelines and Techniques (Part III)
Business Vision and Drivers
Business Capabilities
The method refines understanding of business need
The method delivers new business solutions
The method produces content to be stored in the Repository, classified according to the Enterprise Continuum
Architecture Content Framework (Part IV)
TOGAF ADM & Content Framework
Enterprise Continuum and Tools (Part V)
The Enterprise Continuum and Repository inform the business of current state
TOGAF Reference Models (Part VI)
Operational changes update the Enterprise Continuum and Repository
TOGAF Enterprise Continuum and Tools
Learning from business operation creates new business need

(The Open Group 2009, p. 4)

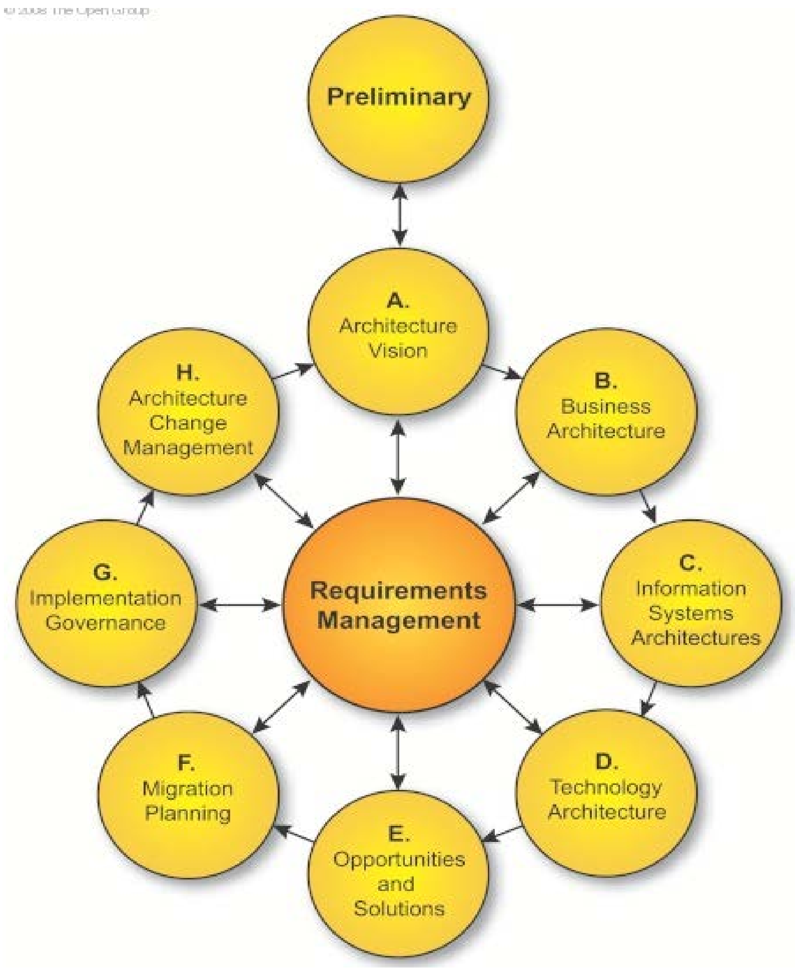
TOGAF Architectures



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

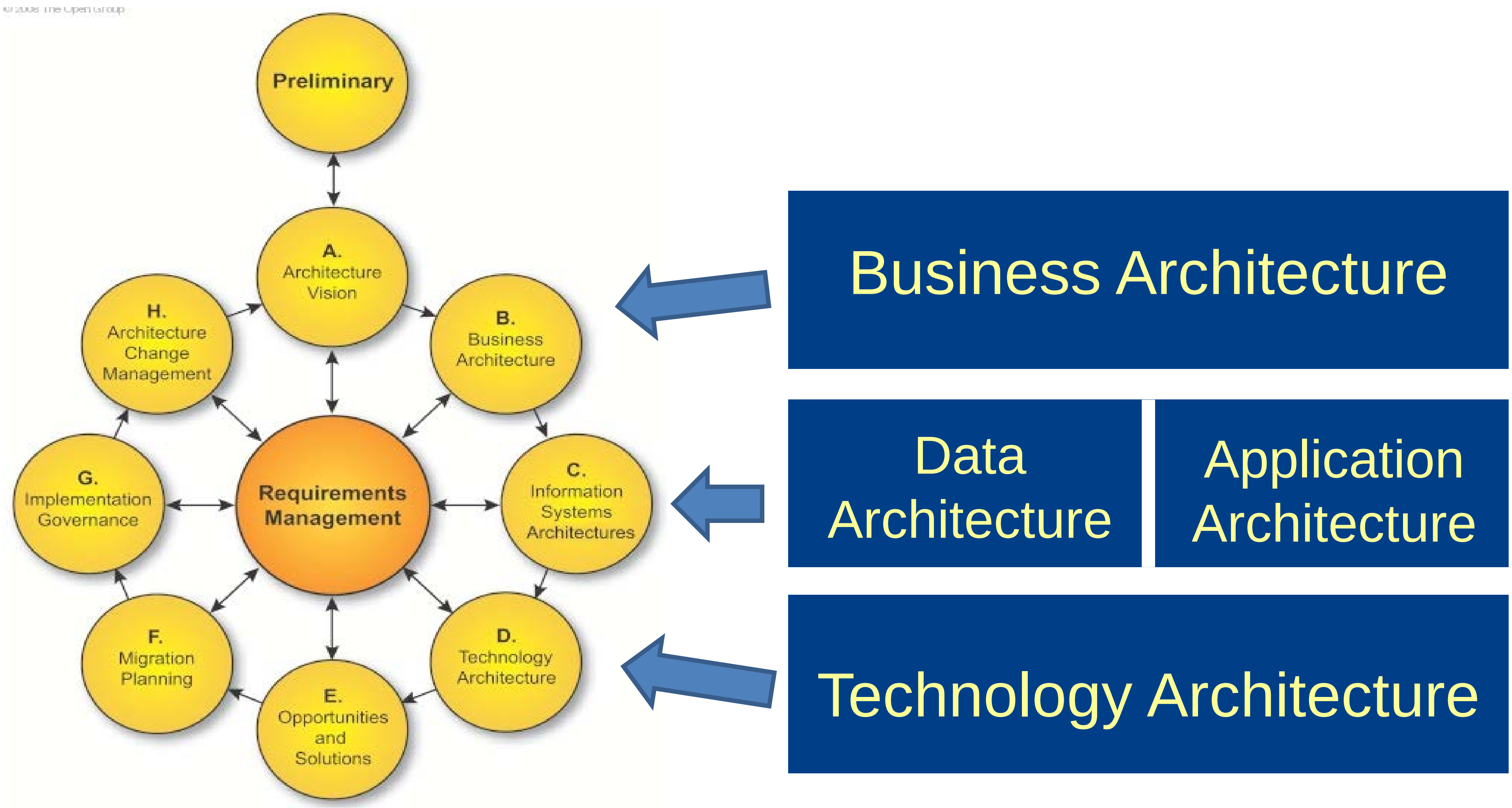
TOGAF Architecture Development Method (ADM)



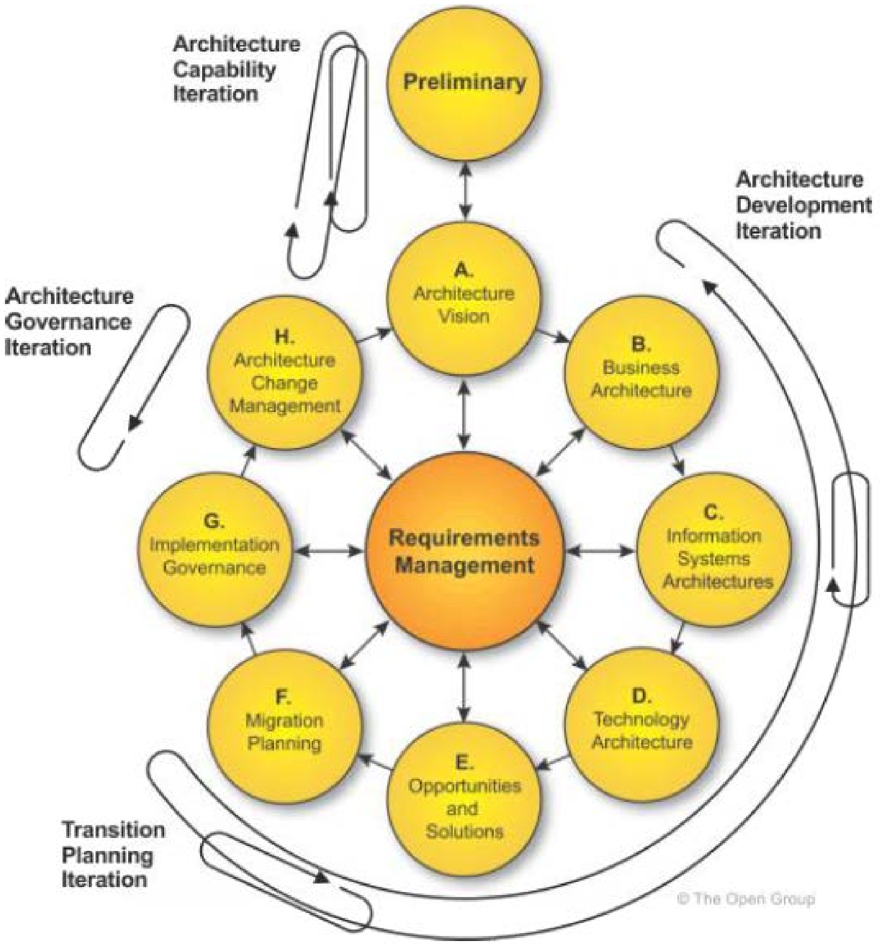
- TOGAF addresses the whole enterprise architecture lifecycle
- The TOGAF Architecture Development Method (ADM) is a generic method for developing an enterprise architecture
- The goals, approaches, required input, activities and deliverables are documented for each phase separately
- The ADM method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

(The Open Group 2009)

TOGAF Architecture Views



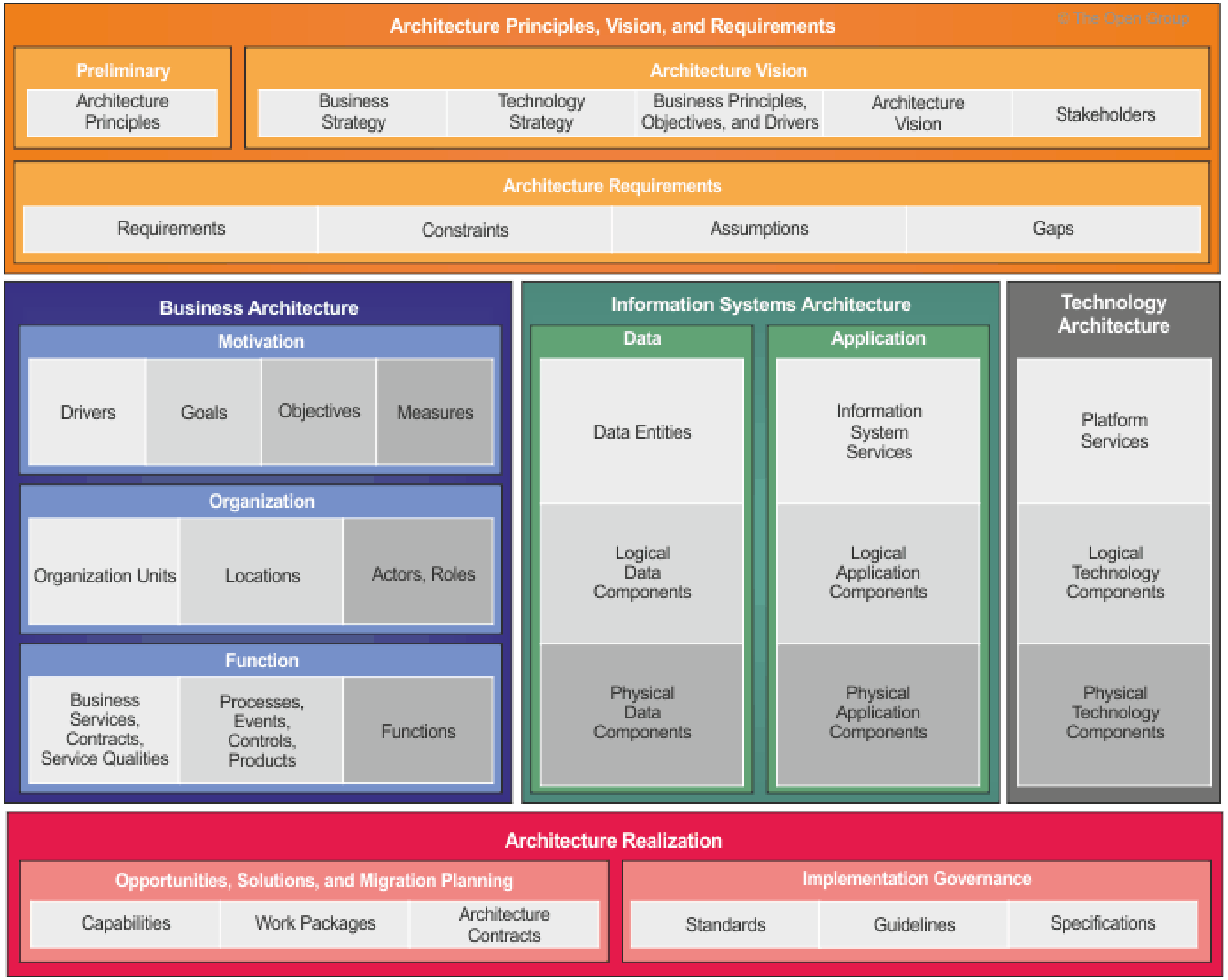
TOGAF Architecture Development Method (ADM)


Architecture Capability Iteration
Preliminary
Architecture Development Iteration
Architecture Governance Iteration
A. Architecture Vision
H. Architecture Change Management
B. Business Architecture
G. Implementation Governance
Requirements Management
C. Information Systems Architectures
F. Migration Planning
D. Technology Architecture
E. Opportunities and Solutions
Transition Planning Iteration
© The Open Group

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

(The Open Group 2011)

TOGAG Content Metamodel



Architecture Principles, Vision, and Requirements

- Preliminary
  - Architecture Principles
- Architecture Vision
  - Business Strategy
  - Technology Strategy
  - Business Principles, Objectives, and Drivers
  - Architecture Vision
  - Stakeholders
- 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

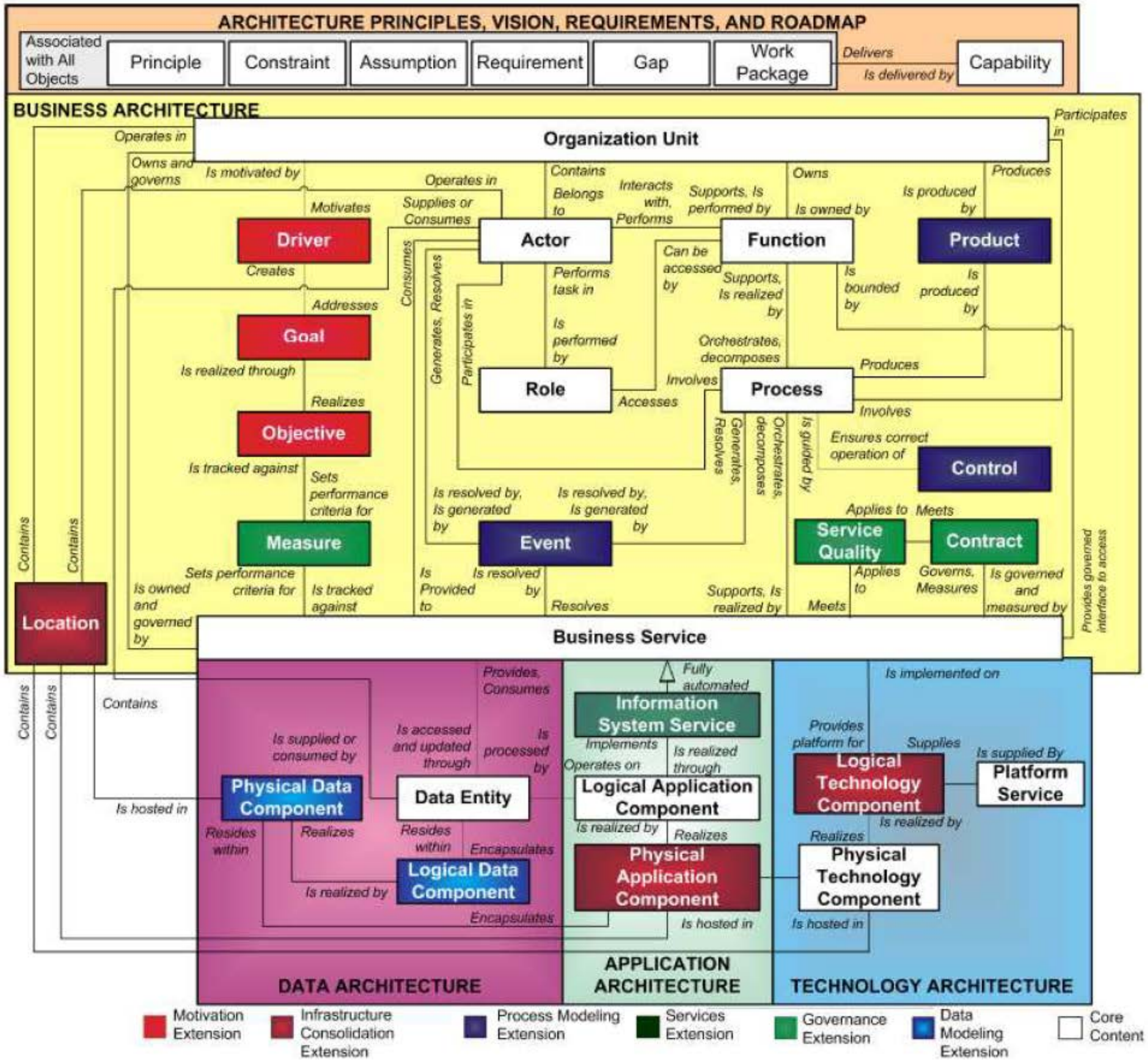
http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap33.html (The Open Group 2011)

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

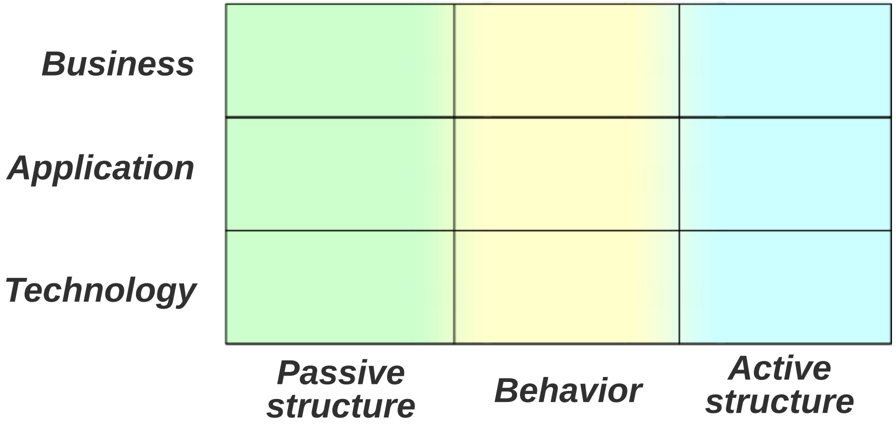
(The Open Group 2011, Part IV)

ArchiMate

ArchiMate

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

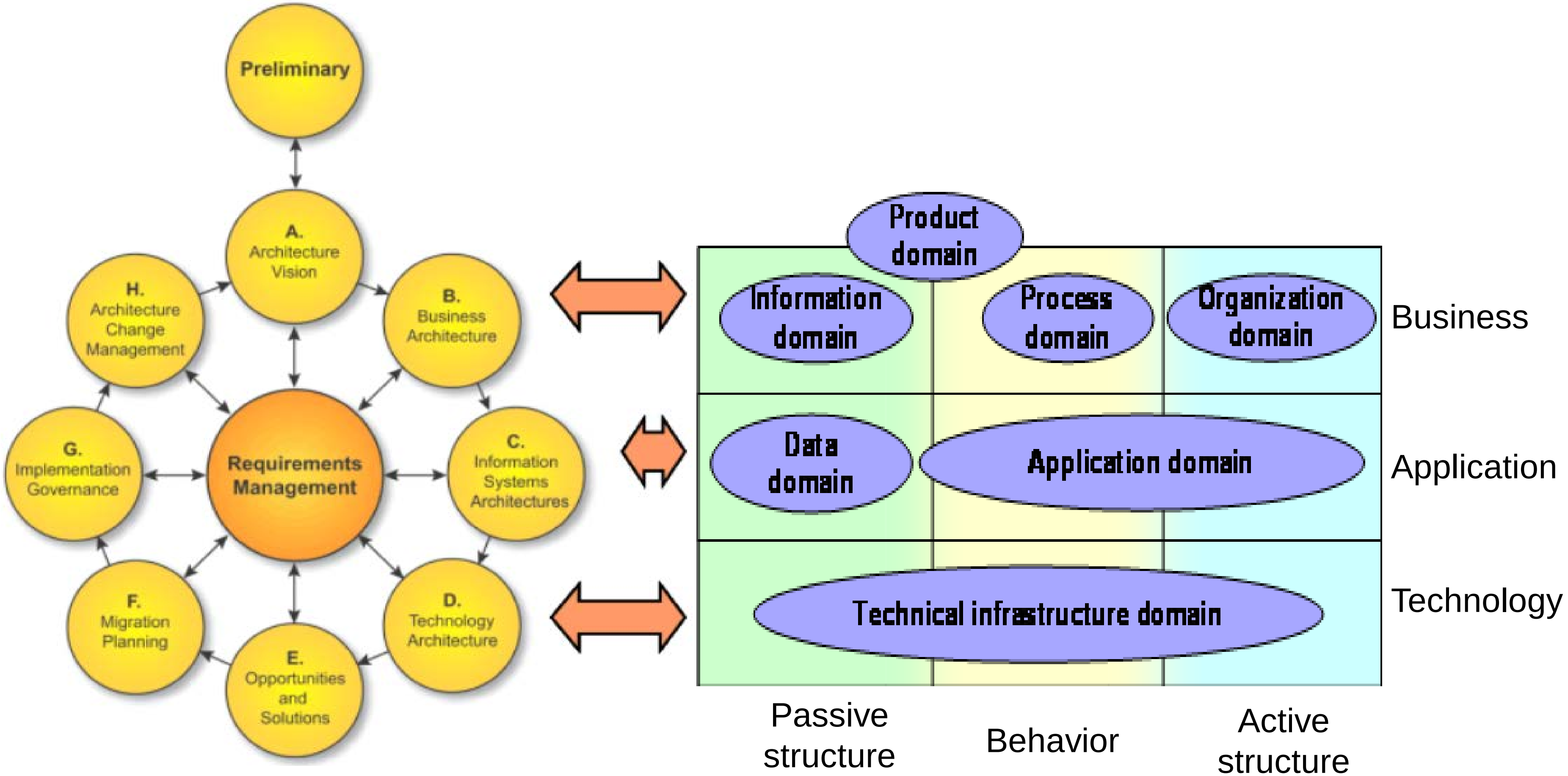
- Three architecture layers:
  - ♦ ***Business***
  - ♦ ***Application***
  - ♦ ***Technology***



Business
Application
Technology
Passive structure
Behavior
Active structure

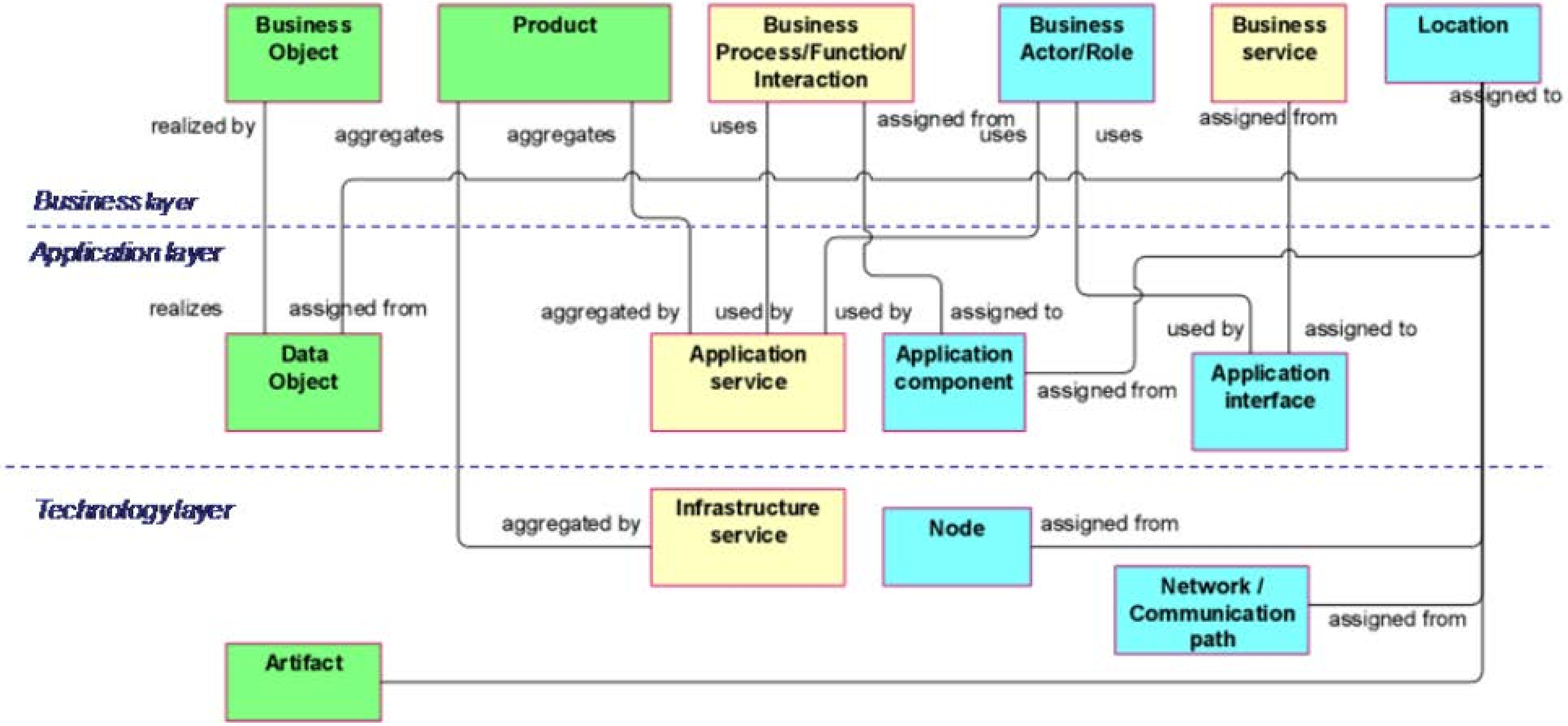
- Three main types of elements:
  - ♦ ***Active structure*** element: an entity that is capable of performing behavior.
  - ♦ ***Behavior*** element: a unit of activity performed by one or more active structure elements.
  - ♦ ***Passive structure*** element: an object on which behavior is performed.

ArchiMate and TOGAF



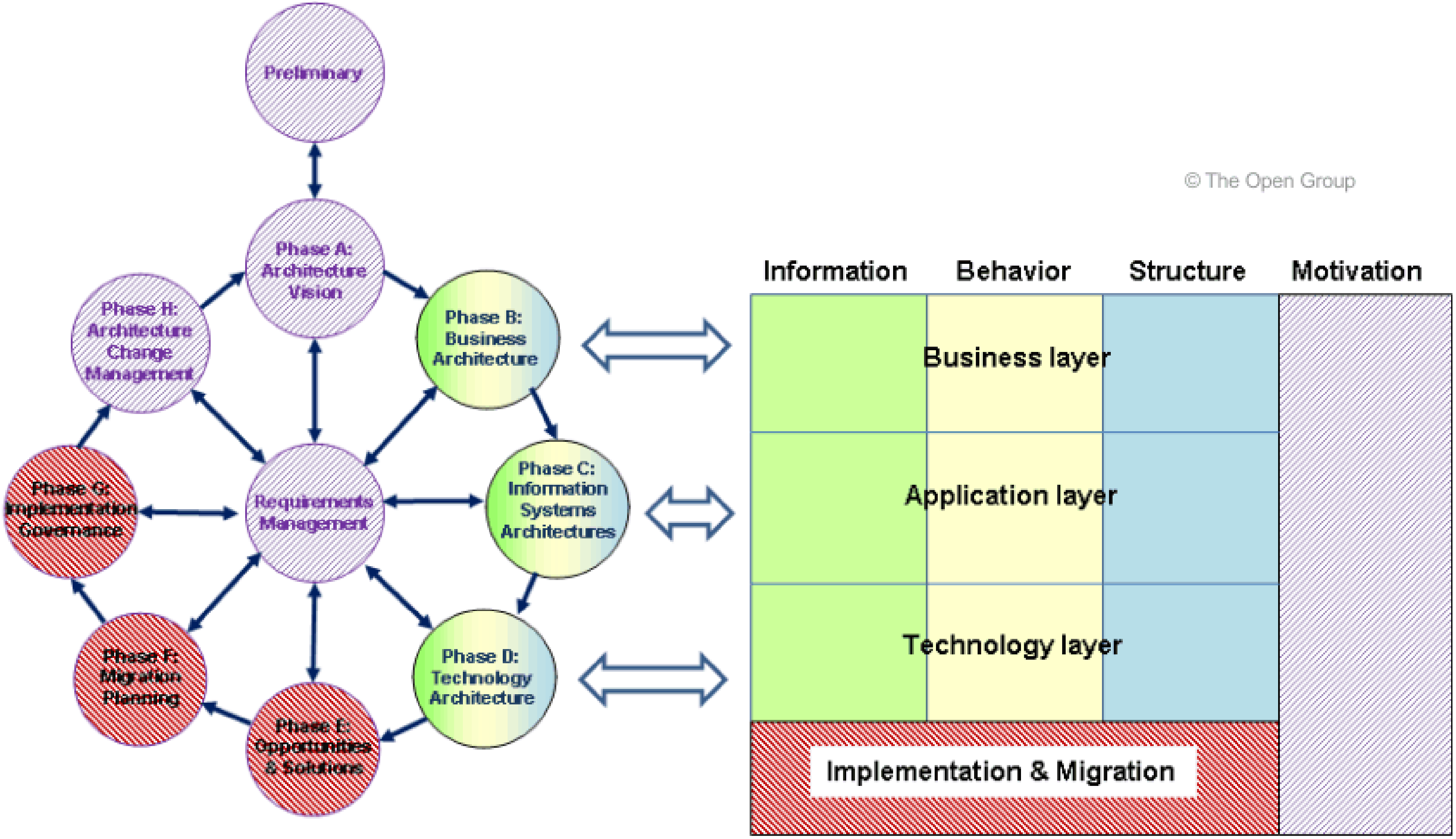
TOGAF ADM

Cross-Layer Dependencies: Business-IT Alignment



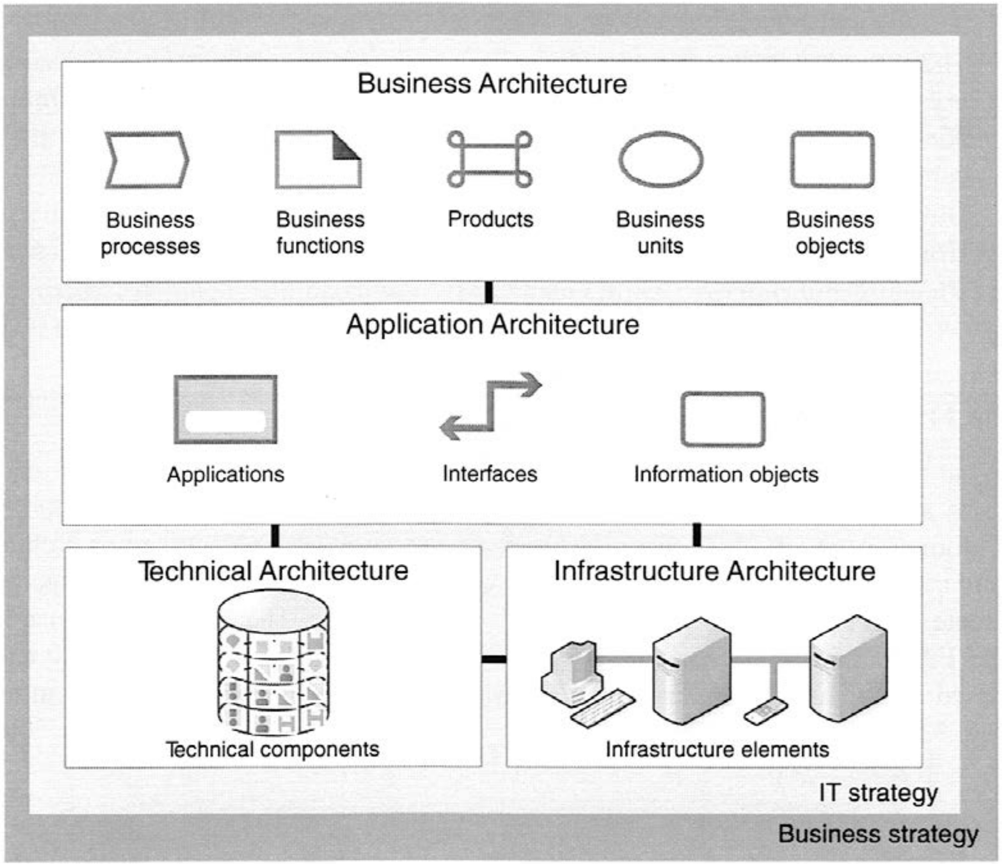
Source: ArchiMate 2.0 Specification, http://pubs.opengroup.org/architecture/archimate2-doc/chap06.html

Extensions of ArchiMate to cover the whole TOGAF ADM


Preliminary
Phase A: Architecture Vision
Phase B: Business Architecture
Phase C: Information Systems Architectures
Phase D: Technology Architecture
Phase E: Opportunities & Solutions
Phase F: Migration Planning
Phase G: Implementation Governance
Phase H: Architecture Change Management
Requirements Management
© The Open Group
Information
Behavior
Structure
Motivation
Business layer
Application layer
Technology layer
Implementation & Migration

Best Practice Enterprise Architecture

Best Practice Enterprise Architecture



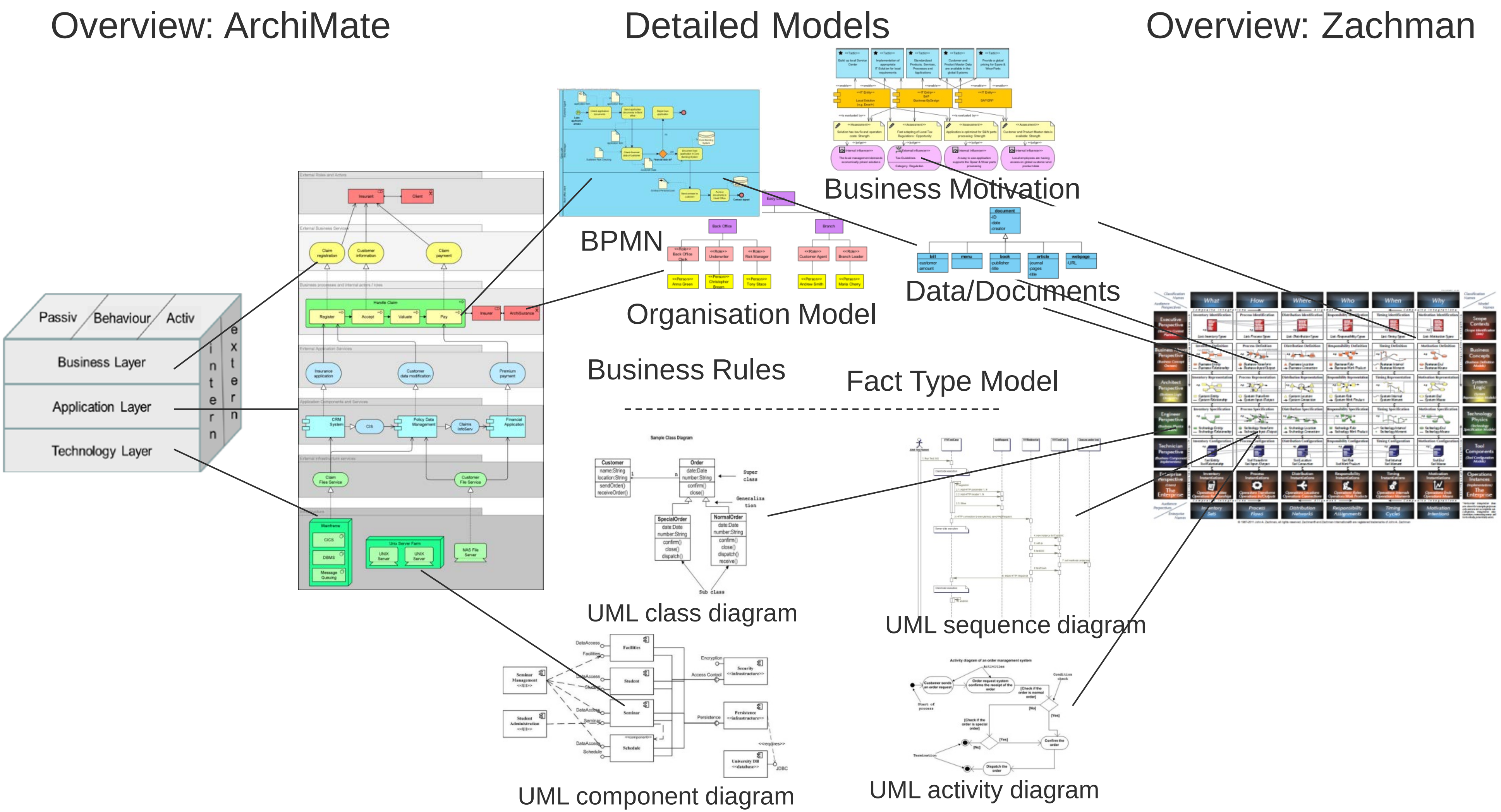
- The Bast Practice Architecture from Inge Hanschke (2010) is another example of a three-layer enterprise architecture framework.
- In contrast to TOGAF
  - it is quite simple
  - it differentiates between the technical architecture and the infrastructure architecture
  - it does not have a separate 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