Architecting the enterprise: Using a Standards Approach

INCOSE International Symposium, July 2010

Richard A. Martin
Tinwisle Corporation
Convener
ISO TC 184/SC 5/WG 1

Why are we here?

Architecture is a people thing we often associate with design intelligence.

We believe all built systems have it.

Understanding architecture helps us to understand better the world we build.

Standards are a people thing as well.

Standards are an architectural pattern.

We believe standards help build better systems for use in the real world.

What will we accomplish?

- Promotion of architecture as an aid in communication among people to enable better understanding of systems
- Identify the prominence of people in architecture and the systems they build for enterprises
- Use of standards from ISO and IEC for architecture practice in general and for architecting enterprise systems in particular

Which standards

ISO/IEC 10746 – ODP Reference model
ISO/IEC 15414 – ODP Enterprise language
ISO/IEC 15288 – Life cycle processes

ISO 14258 – Rules for modeling
ISO 15704 – Frameworks for EA
EN/ISO 19439 – Framework for modeling
EN/ISO 19440 – Constructs for modeling
ISO/IEC 42010 – Architecture description

Tutorial outline

- Situation awareness
- Harmonizing perspectives and Principles

Morning break

- Architecture description
- Frameworks for EA and EA modeling

Lunch break

- TC184/SC5 EA and EAM standards

Afternoon break

- Using standards together

Architectural Discord

Discussions about architecture of any kind are metaphor-rich and consequently subject to much misinterpretation.

Since architecture is pervasive (in good, bad, and ugly forms), context is critical.

The enterprise context ranges from two people in a garage through tens of thousands cooperating in a global supply chain, to millions governed by nation states.

Architectural Harmony

Common concepts – we are all doing very similar things even though we are not using the same words or methods.

Those concepts are independent of methodology and range across many domains of practice.

We recognize the value of "good" architecture and want to encourage its use.

Architectural Value

- Architecture is the means by which enterprise mission and objectives are represented.
- Architecture representations are critical communication vehicles for managers and engineers, especially in large, complicated, and dispersed enterprises and programs
- Architectural standards facilitate completeness and consistency of representation.

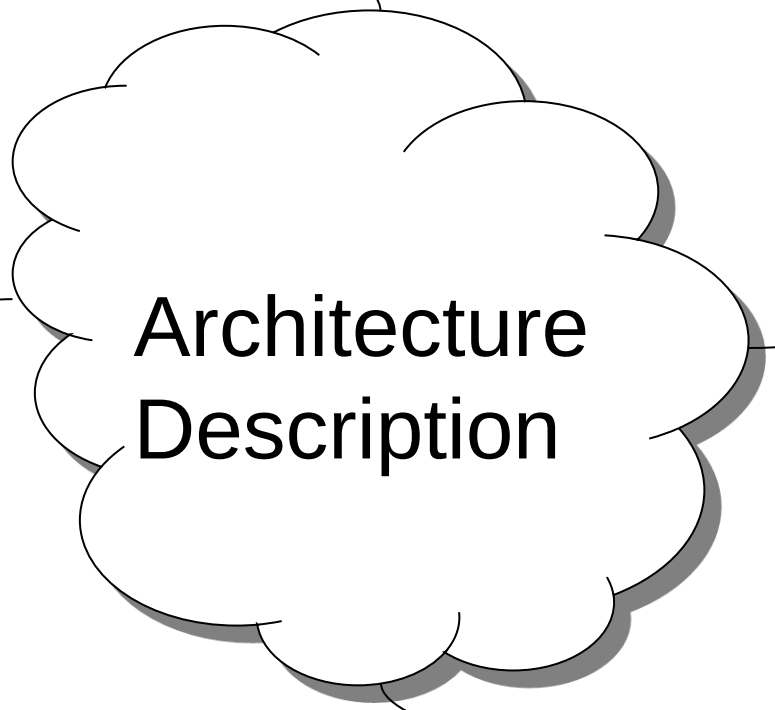
Many Architectures

Many diverse perspectives result in many views and viewpoints about architecture - at least as many as lecturers on the topic – after all it is a human thing!

You will see a few of the many in this tutorial

As an abstract concept, architecture necessarily has a dominating human aspect

AD as boundary object



Literature:
Documentation for current and future generations of users and developers

Blueprint:
Specification of the system to be implemented

Language:
Medium of communication for achieving common understanding

Decision:
Choices about the system to be implemented and rationale

What is architecture?

Characterized with aspects for:

- Form of realization
- Functions to enable or perform
- Experience of use
  - Some call that "objective experience" or "fitness for purpose", some call it "beauty"

Architecture is "conceptual design"

Derived from principles and patterns

It is MBSE!

our focus

Dawn of Architecture

- Marcus **Vitruvius** Pollio, c. 90 - 20 B.C.E.
  - Architect for Augustus Caesar
- de Architectura – 30 B.C.E. in 10 volumes
- First complete text on architecture and the "standard" used for 1500 years
- Three aspects [public] architecture must posses:

	(Gwilt – 1826)	(Teubner – 1899)	(Wotton – 1624)
➢	Strength	(firmitatis)	[firmness]
➢	Utility	(utilitatis)	[commodity]
➢	Beauty	(venustatis)	[delight]

Architecture as Metaphor

The building or landscaped space as a system or enterprise

Intention for an expected or desired utility

Classifications in civil architecture

- Column: Doric, Ionic, Corinthian, Tuscan, etc.
- Building: Gothic, Baroque, Tudor, etc.
- Movements: Art Deco, Post Modern, etc.
- Architects: Wright, Pei, Johnson, etc.

Necessity for component knowledge

Architecture Evolution I

- Evolution of Architecture for a specific purpose
- The work of the practicing architect
- Applying component knowledge to create the form and function fit for purpose
- Conceptual design focused on a product
- Expectation of a stable result

Architecture Evolution II

- Long-term changes in concepts, styles, capabilities
- The work of the architecture discipline
- Applying knowledge of practice to create the form and function fit to advance the discipline
- Conceptual design focused on the process of architecting
- Expectation of repeated use in practice

Architecture definitions

ISO 42010 (FCD) -

3.2 architecture (of a system) fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution

INCOSE IEWG Knowledge Claims - Architecture concerns the arrangement of function and feature that maximizes the objective of the system structure.

A revision version

ISO 15704 (NP+CD)

4.1 **architecture:** conceptualization of the form, function, and fitness for purpose of a system in its environment, as embodied in the elements of the system, the relationships between those elements, the relationship of the system to its environment and the principles guiding the design and evolution of the system.

A different perspective?

Architecture is the art of building in which human requirements and construction materials are related so as to furnish practical use as well as an aesthetic solution, thus differing from the pure utility of engineering construction.

Attributed to Webster's Dictionary although not found in Webster's Online Dictionary, but take a look at the online entry for architecture sometime!

More about architecture

Architecture:

- is conceptual and not physically real
- has a structuring objective
- is about the relationships among objects that produce behavior

---

Observation – ISO/IEC 42010 is about architecture descriptions and ISO 15704 is a framework for enterprise architectures and models

Only two types

There are two, and only two, types of architectures that deal with enterprise integration

I. system architectures that deal with the design of a system, e.g. the computer control system part of an overall enterprise integration system;

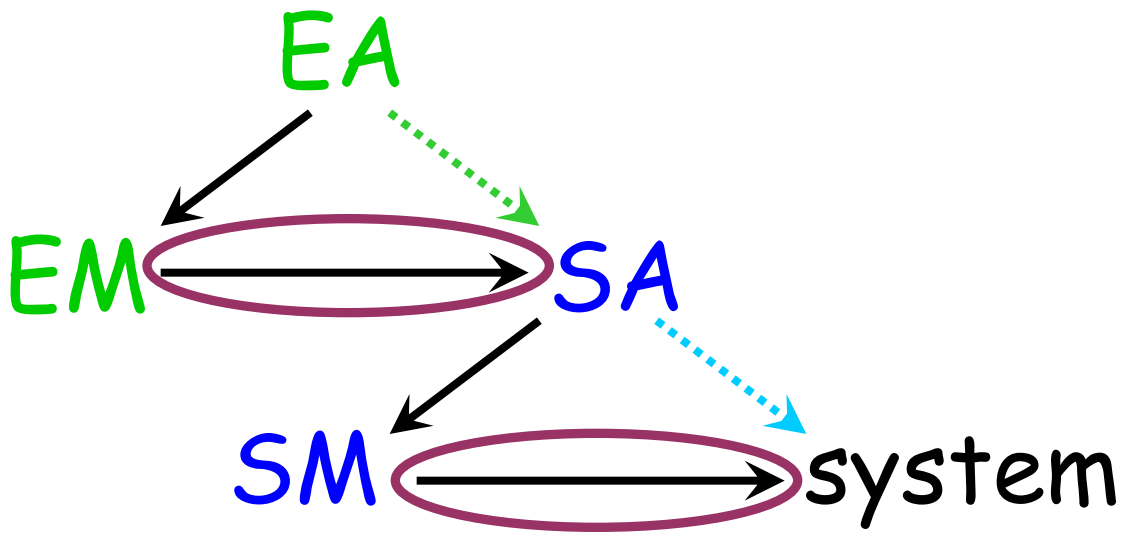
II. enterprise architectures that deal with the development and implementation of a project such as an enterprise integration or other enterprise development programme.

Models and architecture

Architectural intent is embodied in enterprise models (EM).

Architectural realization is embodied in instance manifestations of those models.

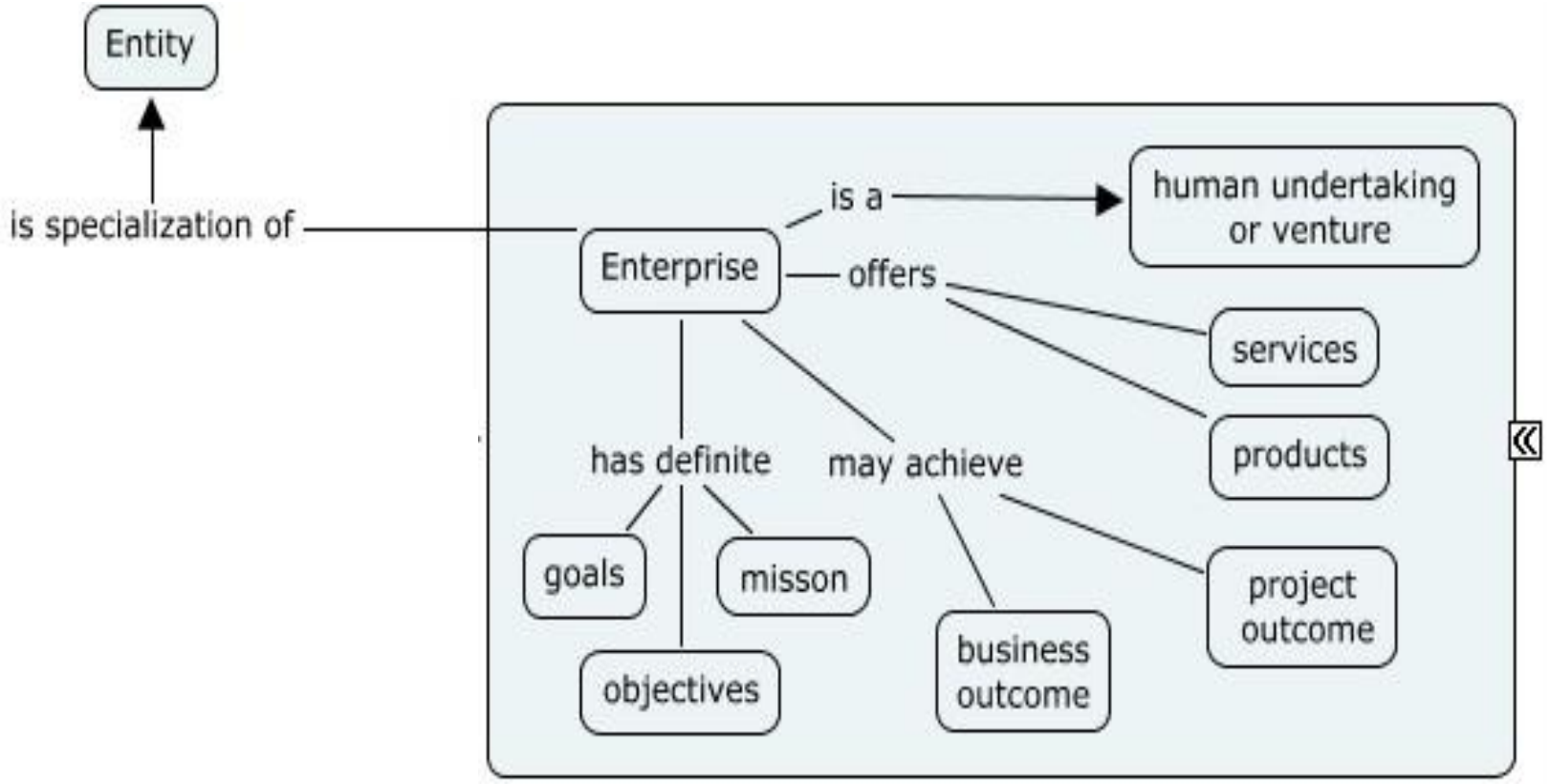
Models are a utility of architecture



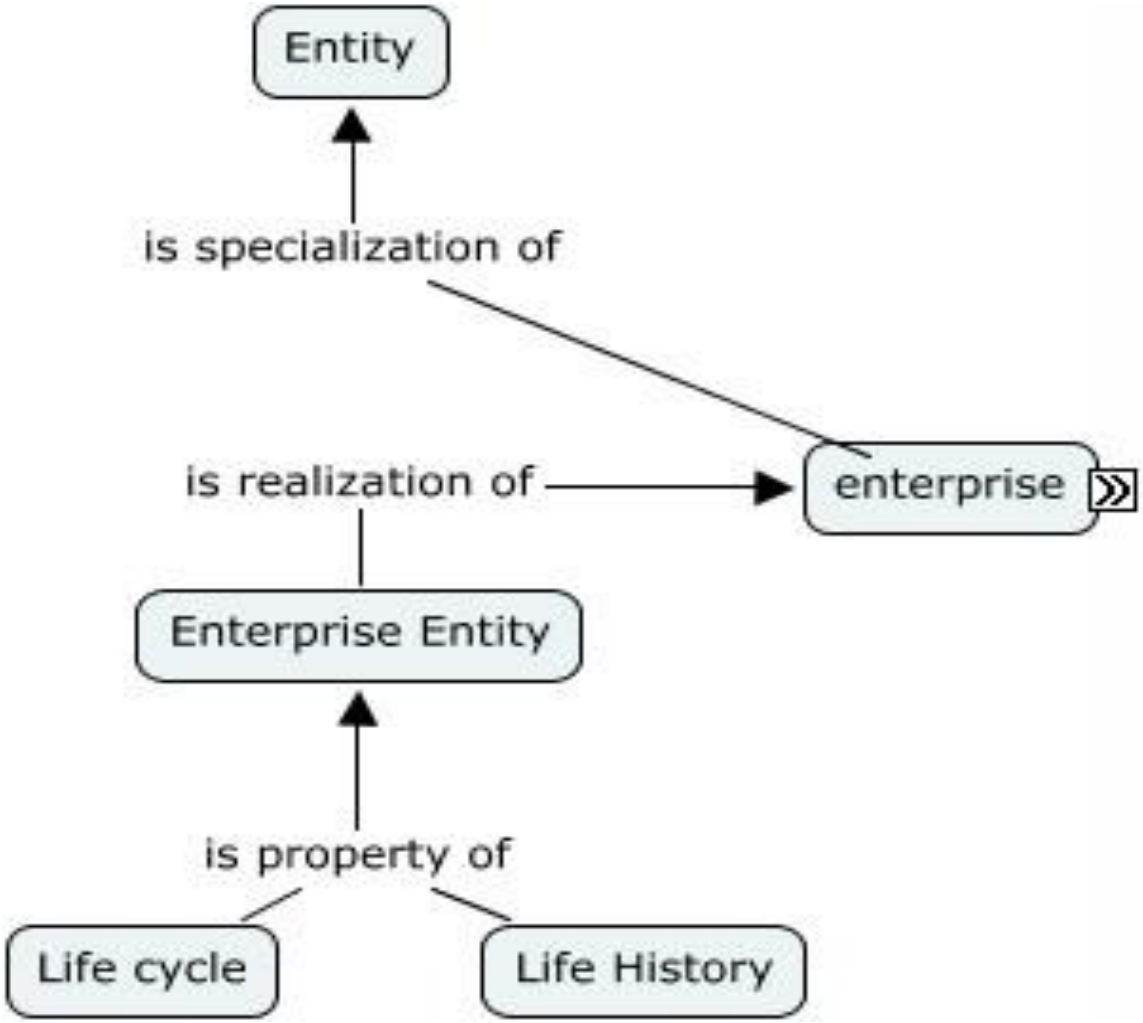
Enterprise as system

- Enterprise is a kind of system
  - Purpose
  - Emergent properties
  - Life cycle
  - Component parts
- Enterprise is distinguished by the continuous, purposeful engagement of people

An Enterprise



An Enterprise Entity

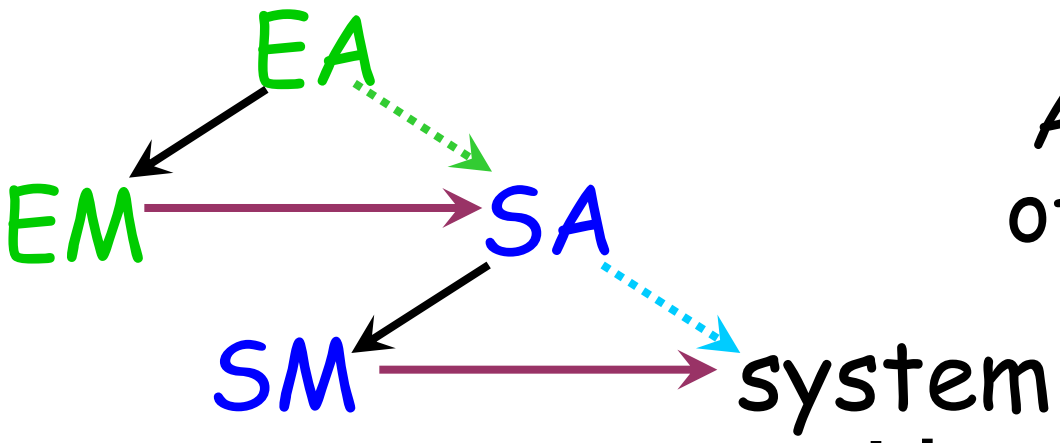


Abstraction quiz

7853981633974483096156608…

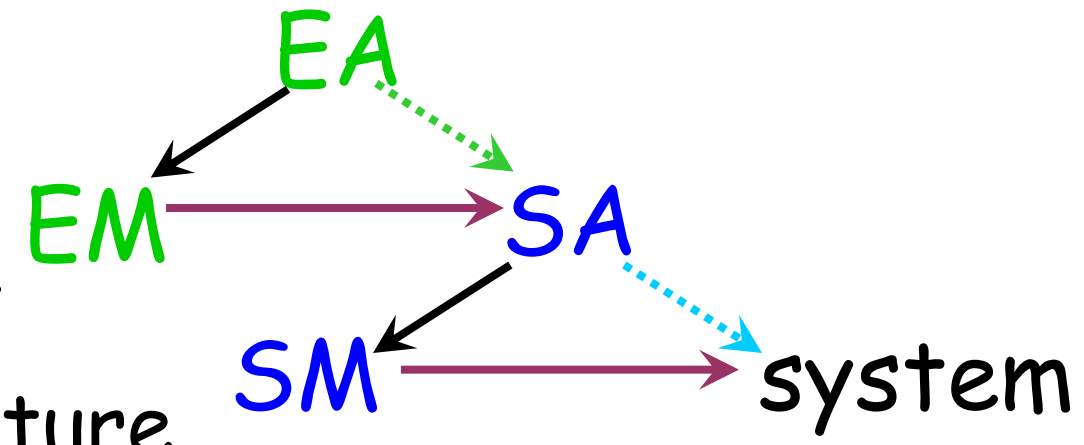
What is the next element of the sequence?

Different perspectives



An enterprise of enterprises and other systems

Software architecture as a sub-type of system architecture



Metamorphic Analogy

- *Danaus plexippus* – Common North America Monarch butterfly
- Four stage life-cycle of Lepidoptera
  - Egg — 4 days
  - Larva (instar) — 2 weeks
  - Chrysalis (pupa) — 2 weeks
  - Adult — 2-4 / 24-36 weeks
- Annually migrate from central Mexico to northern US and southern Canada

Monarch Migration

- Fall migration from north to south
  - Each butterfly travels 1200 - 2000 miles at 25 – 30 miles per day, a trip of 2 months
- Over-winter in Oyamel fir trees in mountains of central Mexico
- Spring migration from south to north
  - Fly to south US and produce generation one
  - Gen 1 fly to central US to produce Gen 2
  - Gen 2 fly to northern US & CA, produce Gen 3
  - Gen 3 produces Gen 4 (& Gen 4 sometimes Gen 5)
  - Gen 4 flies back to Mexico

Systems within a system

- Each Monarch has a well-defined life cycle – phases are of different duration
- Species *Danaus plexippus* has a well-defined annular life cycle – not shared by any individual member of the species
- Two systems with different scale, life cycles, emergent properties, & parts, but nonetheless completely dependent upon each other

Harmonizing international standards for enterprise and system architecture

Harmonizing standards

The great thing about standards is that there are so many from which to choose.

Internationally, the interoperation of standards is a BIG concern.

Culture, language, and use domain are all barriers to interoperation.

Breaking down standard silos requires cooperation and accommodation.

- TC184/SC5/WG1 & JTC1/SC7/WG42

ISO TC184

ISO/IEC JTC1/SC7/WG42

CEN TC310/WG1

ISO/TC 184
~~Industrial~~ Automation Systems & Integration

SC 1
Physical Device Control

- **WG7** Data modeling for integration of physical devices
- **WG8** Distributed installation in industrial applications

SC 2
Robots and robotic devices (reorganizing)

- **WG1** Vocabulary
- **WG3** Industrial safety
- **WG7** Personal care safety
- **WG8** Service robots

SC 4
Industrial Data

- **PPC** Policy & planning committee
- **QC** - Quality Committee
- **WG2** Standard parts library
- **WG3** Product Modeling
- **JWG8**** Manufacturing process and management information
- **WG11** Express language, implementation methods
- **WG12** Common resources
- **WG13** Industrial data quality

SC 5
Architecture, Communications & Integration Frameworks

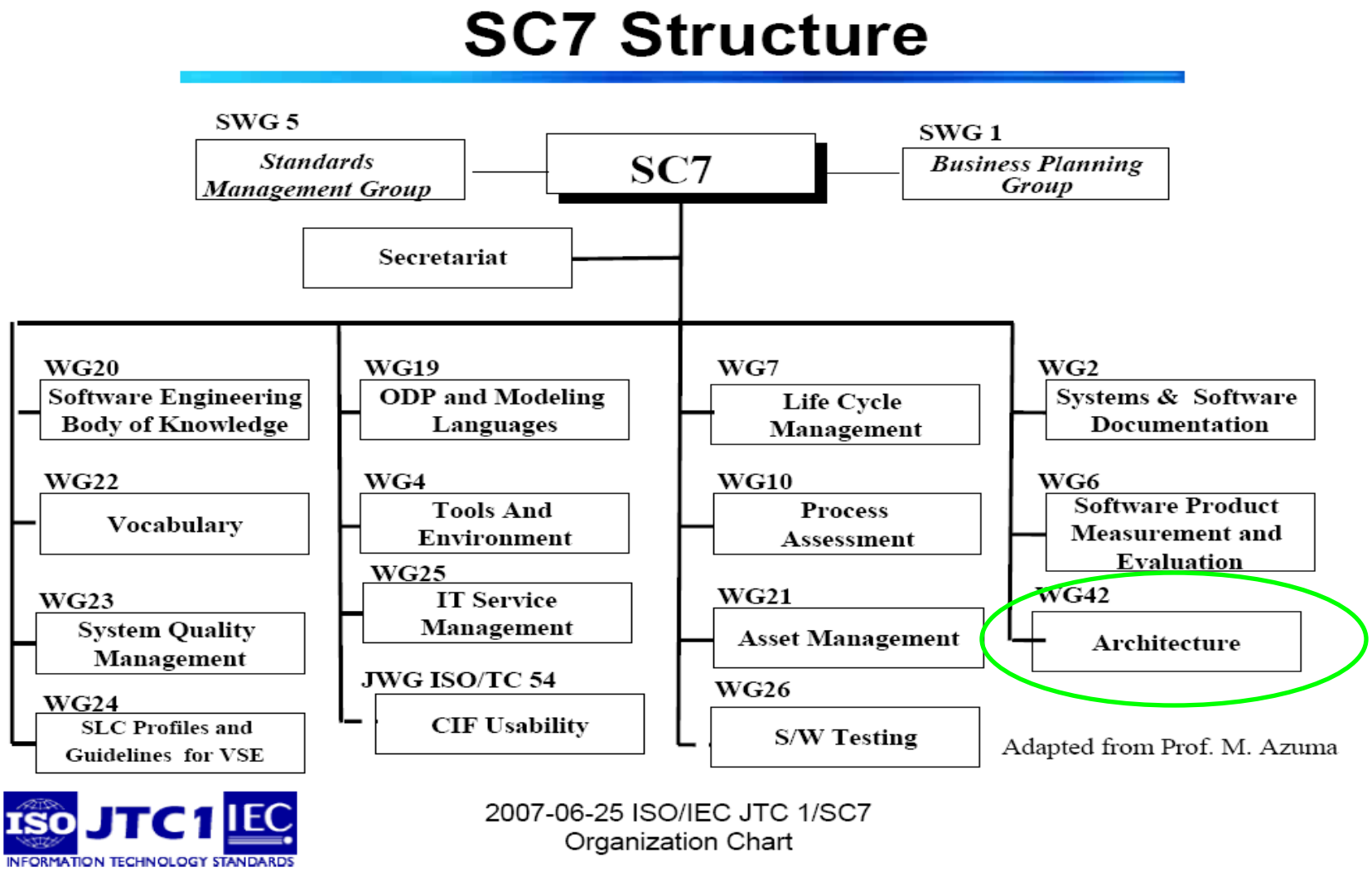
- **WG1** Modeling & Architecture
- **WG4** Manufacturing Software & Environment
- **WG5** Application Integration Frameworks
- **WG6** Application Service Interface
- **WG7** Diagnostics/Maintenance/Control Integration
- **JWG8**** Manufacturing Process and Management Information
- **WG9** KPI for Manufacturing Management
- **WG10** Energy Efficient Manufacturing
- **JWG15** Enterprise-Control System Integration

Advisory Group

SC5 Chair: Emanuel delaHostria (USA)
Contact: ISO/TC 184/SC 5 Secretariat
gre_winchester@nema.org

ISA SP95/**IEC** SC65E

ISO/IEC JTC1/SC7



ISO's EA Groups

ISO/TC184/SC5/WG1: Developing enterprise architecture standards based on manufacturing industry for international architecture applications

ISO/IEC/JTC1/SC7/WG42: Developing a systems- and software-related architecture standard

ISO/IEC/JTC1/SC7/WG19: Developing frameworks and languages for distributed processing (ODP)

Related EA activity

ISO/TC184/SC4/WG3: Developing Data Standards including ISO 10303, 100s of standards including AP233

ISO/TC184/SC5/SG1 & ISO/IEC/SC7/WG22: identifying vocabulary overlaps/conflicts and recommending commonality via "Harmonization"

+ many other indirectly related efforts

Architecture standards

**** Proposed new title***

**** Proposed new title***

ISO/IEC 42010 Systems and software engineering – Architecture description

ISO 14258 Rules and Guidelines for Modelling

ISO 15704 Framework for enterprise architectures and models

(Needs for Frameworks, Methodologies, Languages, Tools, Models, Modules)

Frameworks	*Languages*	*Modules*
CEN/ISO 19439 *Framework for Modelling*	*CEN/ISO 19440* *Constructs for Modelling*	*ISO 16100 Mfg. Software Capability Profiling*
ISO 15745 Framework for Application Integration	*ISO 18629 Process Specification Language*	*IEC/ISO 62264 Control Systems Integration*
ISO 10746 *Ref. Model - ODP*	*ISO/IEC 15414* *ODP Enterprise Language*	SC5 & WG1 SC7 & WG42
ISO 15288 & ISO 12207 *Life Cycle Mgmt.*	*BPML (2001). Business Process Modelling Language*	
OMG MDA *Model Driven Architecture*	*ebXML Electronic Business using eXtensible Mark-up Language*	

K. KOSANKE and M. ZELM (CIMOSA Association)

D. CHEN (LAPS, University Bordeaux 1)

Architecture description

ISO/IEC 42010:2007 Recommended practice for architectural description of software-intensive systems

- Identifying and arranging products to document an architecture
- Distinguishes an architecture description from the mechanisms for description generation

Generalizing standards

ISO 15704:2000 – Requirements for enterprise-reference architectures and methodologies

- Merging of previous work - PERA, IEM, GRAI GIM, CIMOSA, and GERAM
- Presents principles for enterprise architecture
- Extends ISO 14258 concepts of model, view, life cycle, recursion, and iteration with life history and genericity

Unified model framework

ISO 19439 - Enterprise integration: Framework for enterprise modelling

- Based upon CEN ENV 40003:1990
- Objective is to further enable model based execution using enactable models
- Aligned with ISO 15704 (a GERA model)
- Articulates 3 dimensions of enterprise modeling as a framework:

  Phase, View, and Genericity

Modeling constructs

ISO 19440 Enterprise integration – Constructs for enterprise modelling

- Based upon CEN ENV 12204:1996
- Aligned with ISO 15704 (an EML artifact)
- Articulates modeling constructs for manufacturing automation
- Elaborates the CIMOSA Baseline example of 19439 with constructs

ISO 19440 (cont.)

- Constructs for enterprise modeling
  - common semantics enable model unification
  - usable across phases of model development
  - support process-oriented approach
- Arrangement and specialization using templates into structures for a specific purpose
- No mapping between functional operations and capabilities
- No explicit versioning mechanism

Standards reflect practice

- WG1 standards reflect industrial modeling practice of the 1990's
  - **P**urdue **E**nterprise **R**eference **A**rchitecture
  - **C**omputer **I**ntegrated **M**anufacturing **O**pen **S**ystems **A**rchitecture
  - **G**raphe a **R**esultats et **A**ctivite **I**nter-relies
- WG42 standard reflects software intensive systems practice of the 1990's
- Do they harmonize with current practice?
  - DoDAF/MoDAF, Zachman, TOGAF, Dual-Vee, etc., and other gallery members

Zachman framework

ENTERPRISE ARCHITECTURE - A FRAMEWORK ™

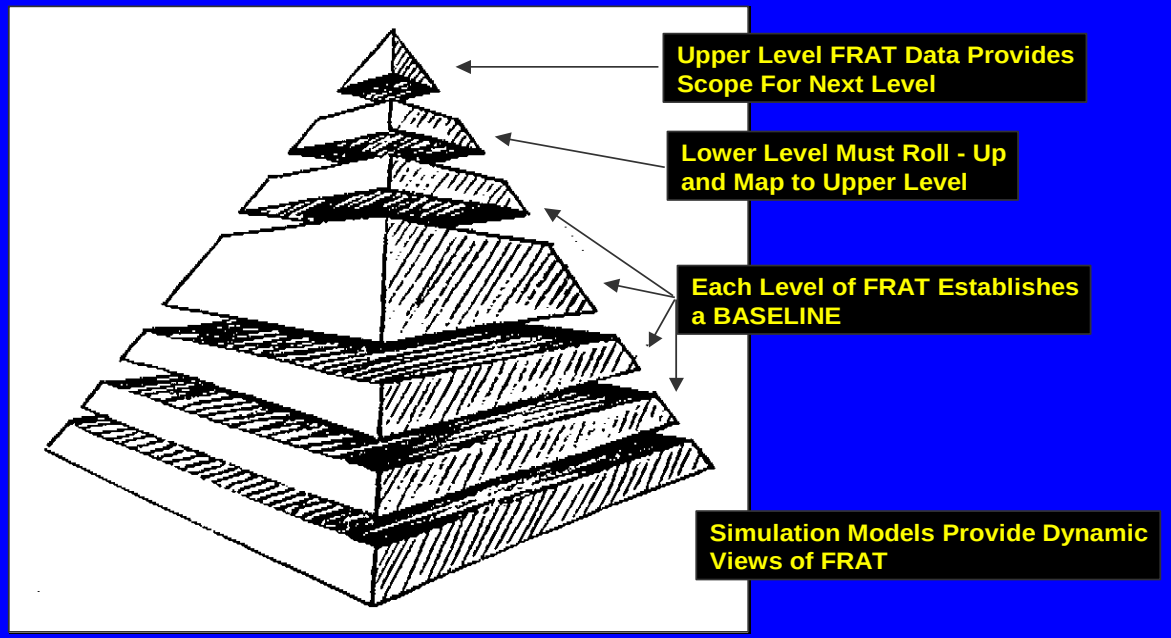
	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL) *Planner*	List of Things Important to the Business ENTITY = Class of Business Thing	List of Processes the Business Performs Function = Class of Business Process	List of Locations in which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organizations	List of Events Significant to the Business Time = Major Business Event	List of Business Goals/Strat Ends/Means=Major Bus. Goal/ Critical Success Factor	SCOPE (CONTEXTUAL) *Planner*
ENTERPRISE MODEL (CONCEPTUAL) *Owner*	e.g. Semantic Model Ent = Business Entity Reln = Business Relationship	e.g. Business Process Model Proc. = Business Process I/O = Business Resources	e.g. Logistics Network Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organization Unit Work = Work Product	e.g. Master Schedule Time = Business Event Cycle = Business Cycle	e.g. Business Plan End = Business Objective Means = Business Strategy	ENTERPRISE MODEL (CONCEPTUAL) *Owner*
SYSTEM MODEL (LOGICAL) *Designer*	e.g. Logical Data Model Ent = Data Entity Reln = Data Relationship	e.g. "Application Architecture" Proc .= Application Function I/O = User Views	e.g. "Distributed System Architecture" Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	e.g. Human Interface Architecture People = Role Work = Deliverable	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means =Action Assertion	SYSTEM MODEL (LOGICAL) *Designer*
TECHNOLOGY MODEL (PHYSICAL) *Builder*	e.g. Physical Data Model Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	e.g. "System Design" Proc.= Computer Function I/O = Screen/Device Formats	e.g. "System Architecture" Node = Hardware/System Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Format	e.g. Control Structure Time = Execute Cycle = Component Cycle	e.g. Rule Design End = Condition Means = Action	TECHNOLOGY CONSTRAINED MODEL (PHYSICAL) *Builder*
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	e.g. Data Definition Ent = Field Reln = Address	e.g. "Program" Proc.= Language Stmt I/O = Control Block	e.g. "Network Architecture" Node = Addresses Link = Protocols	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification End = Sub-condition Means = Step	DETAILED REPRESEN-TATIONS (OUT-OF CONTEXT) *Sub-Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

Zachman Institute for Framework Advancement - (810) 231-0531 Copyright - John A. Zachman, Zachman International

Role by **Interrogative** grid of cells containing models of the enterprise. A proto-typical **Framework**!

FRAT

Systems thinking -
each of the FRAT views can be described with

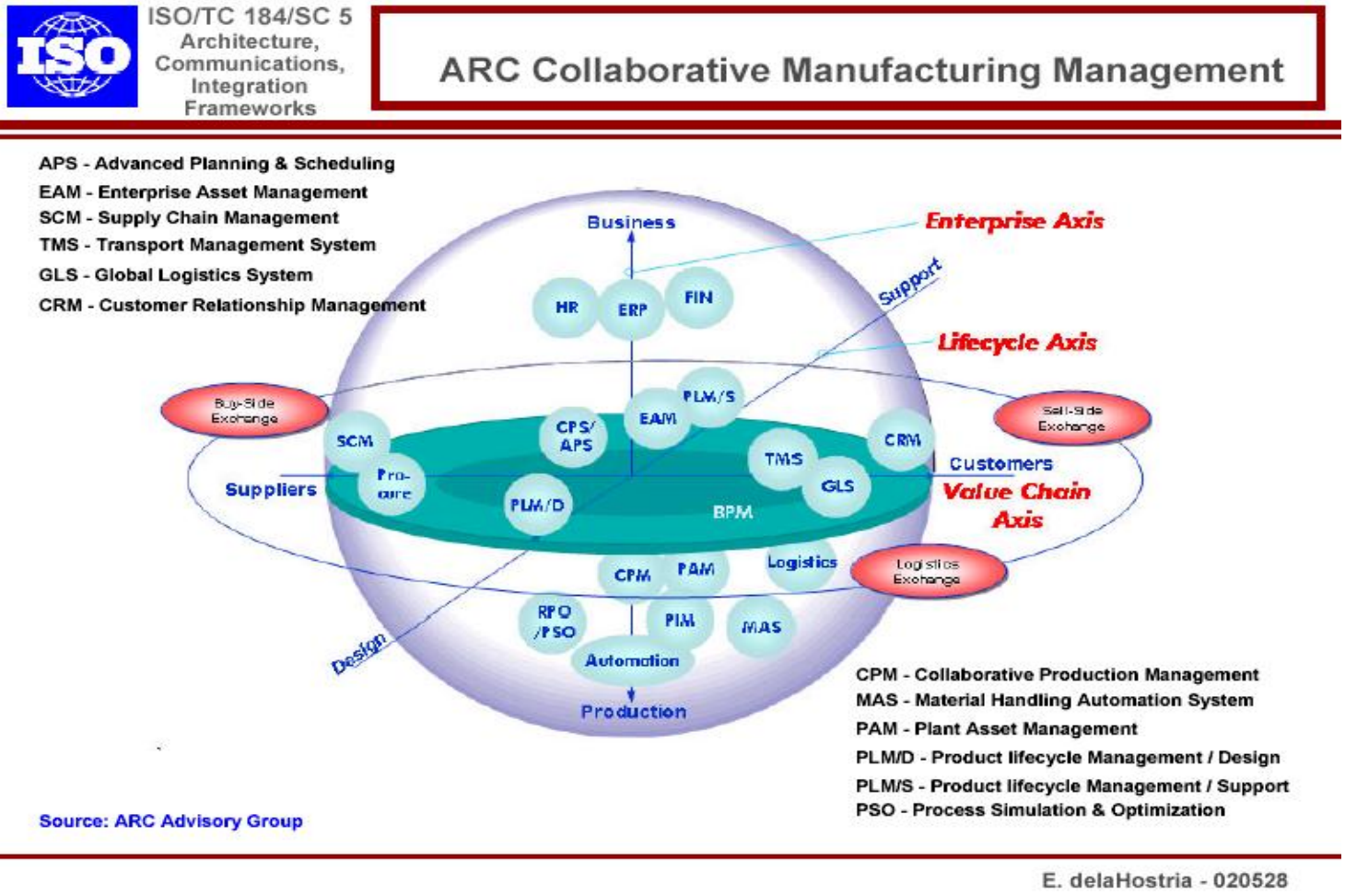


Function
Requirements
Answers
Test

Detail elaboration adds both depth and breadth to the system description

Source: B. W. Mar, B. G. Morais, FRAT – A Basic Framework for Systems Engineering, INCOSE 2002

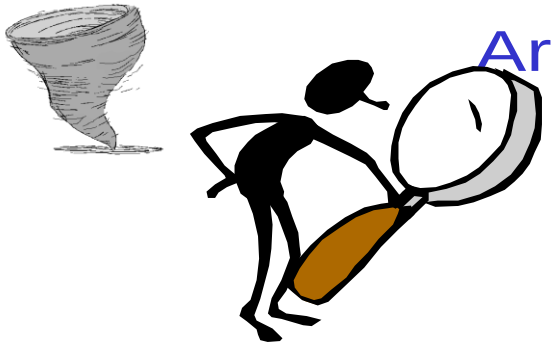
ARC CMM



Aligning functional applications along axis to identify dimensions of the global manufacturing enterprise

Source: ISO/TC 184/SC5 N913, E. delaHostria, Chairman, and ARC Advisory Group (used with permission)

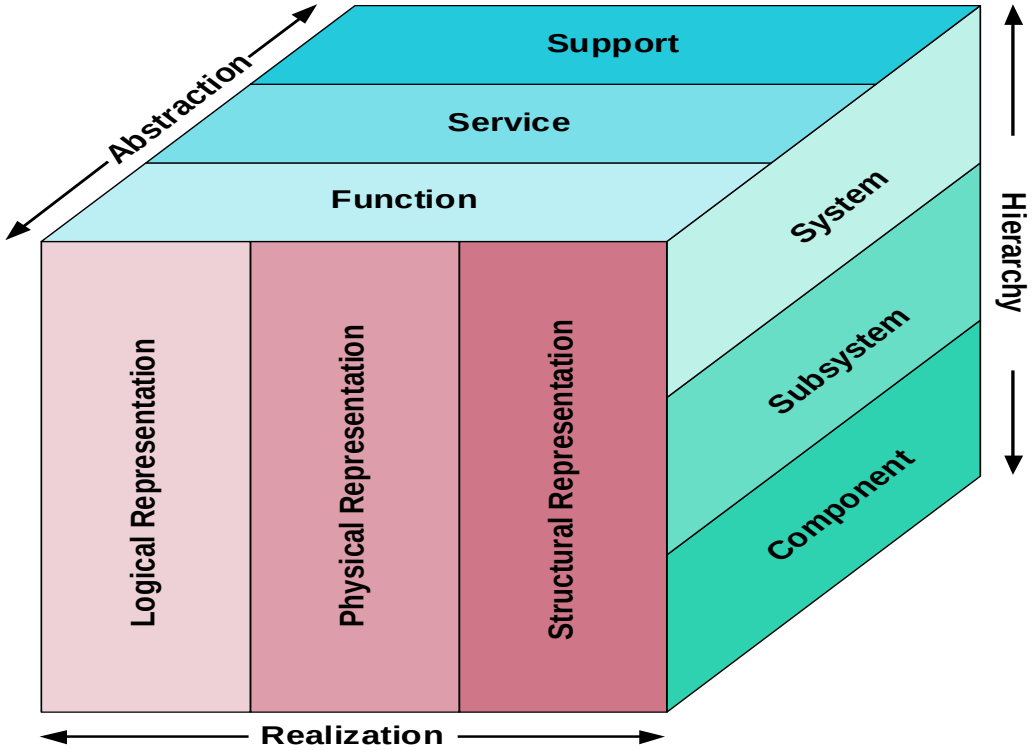
Rockwell Collins



Architectural Perspectives

Perspective Filters:

- Development & Verification
- Manufacturing & Production
- Storage & Transportation
- Installation & Deployment
- Simulation & Training
- Operational
- Maintenance & Support
- Disposal
- Project
- Safety
- Functional
- Physical
- Information/ Data Flow



6 Dec 2001
Ver 2.5

37

Source: R. W. Jorgensen, Architectural Abstractions, INCOSE 2002. Copyright © 2001 Rockwell Collins, Inc. All Rights Reserved.

Permission is hereby granted to use this copyrighted material for any lawful purpose.

Purdue Enterprise Reference Architecture

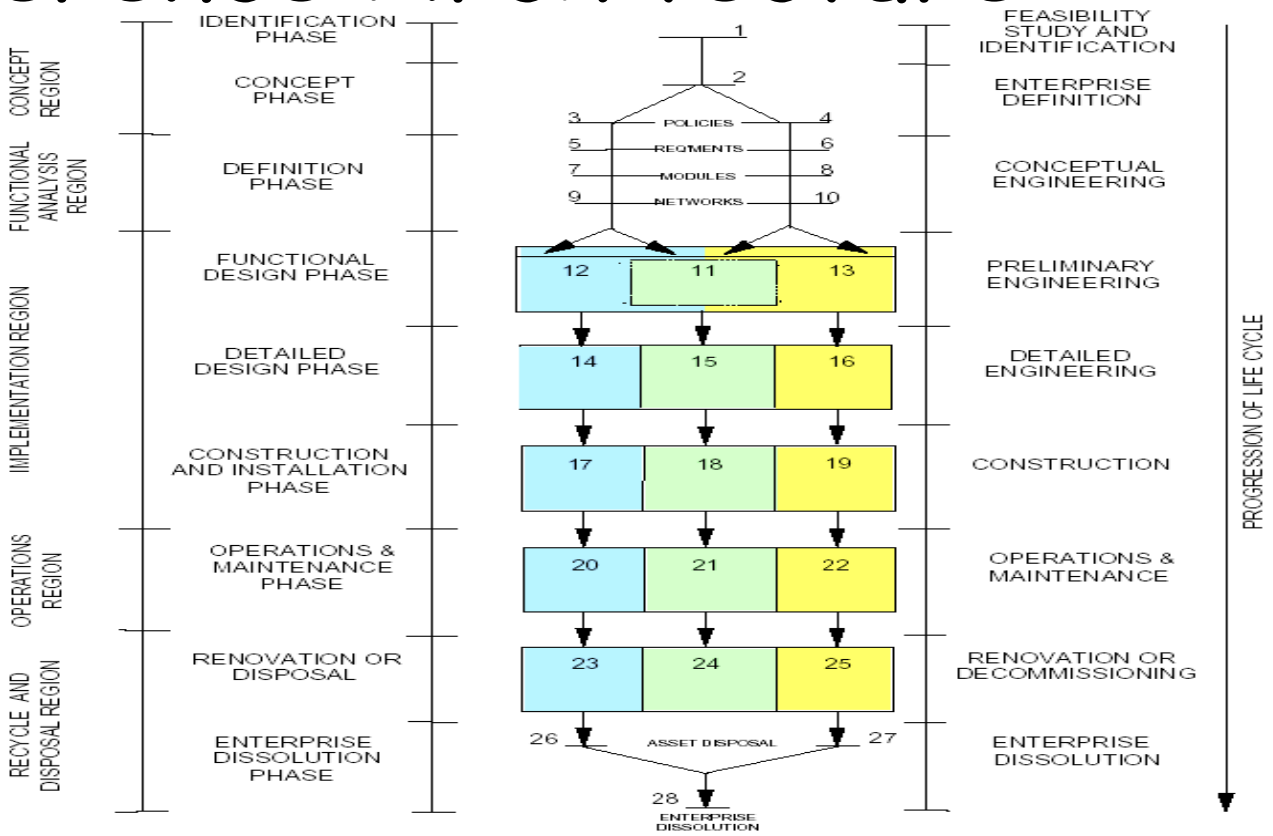
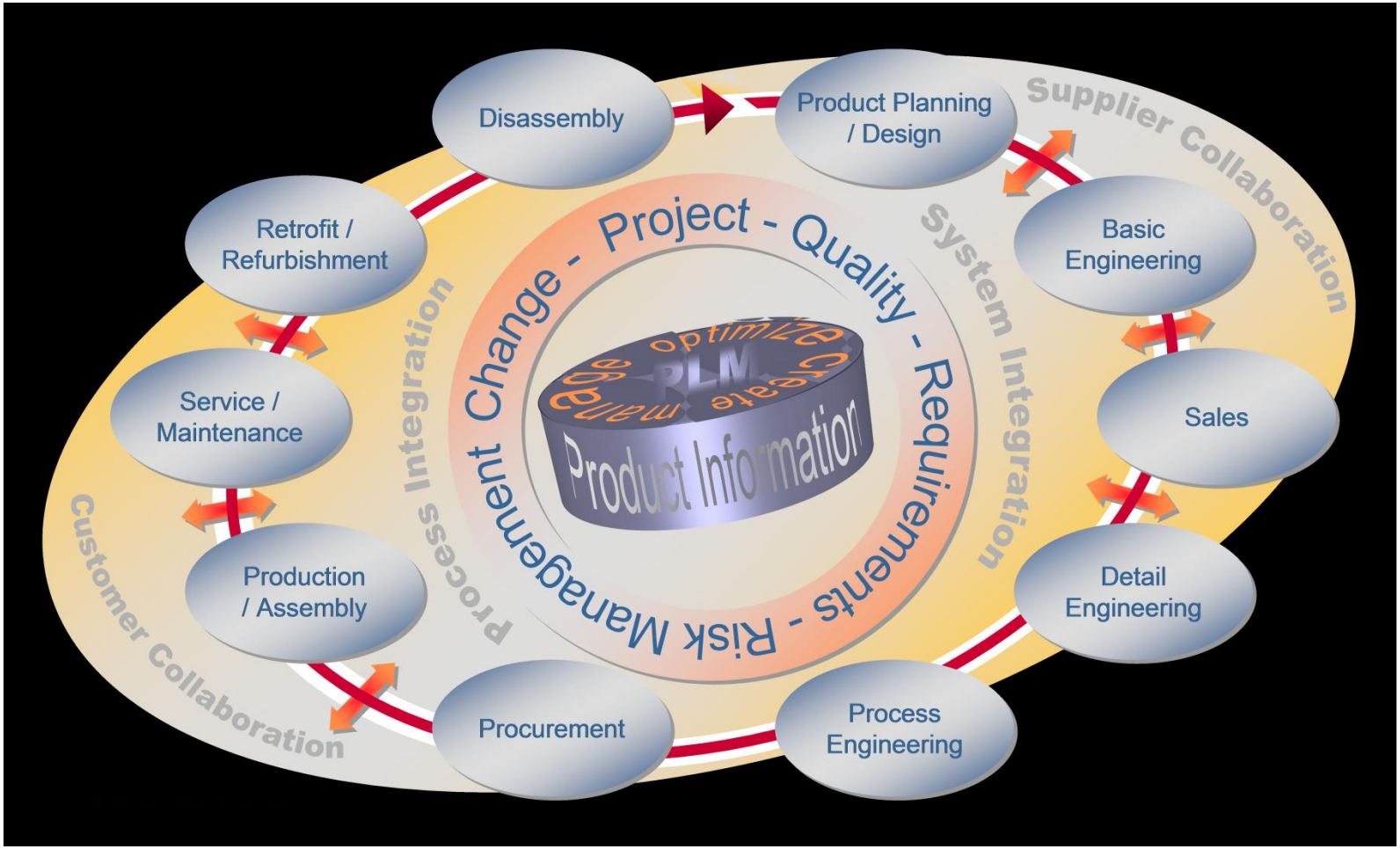


FIGURE 2 - OVERALL FORM OF THE PURDUE ENTERPRISE REFERENCE ARCHITECTURE DIAGRAM SHOWING VARIOUS FORMS OF THE LIFE CYCLE

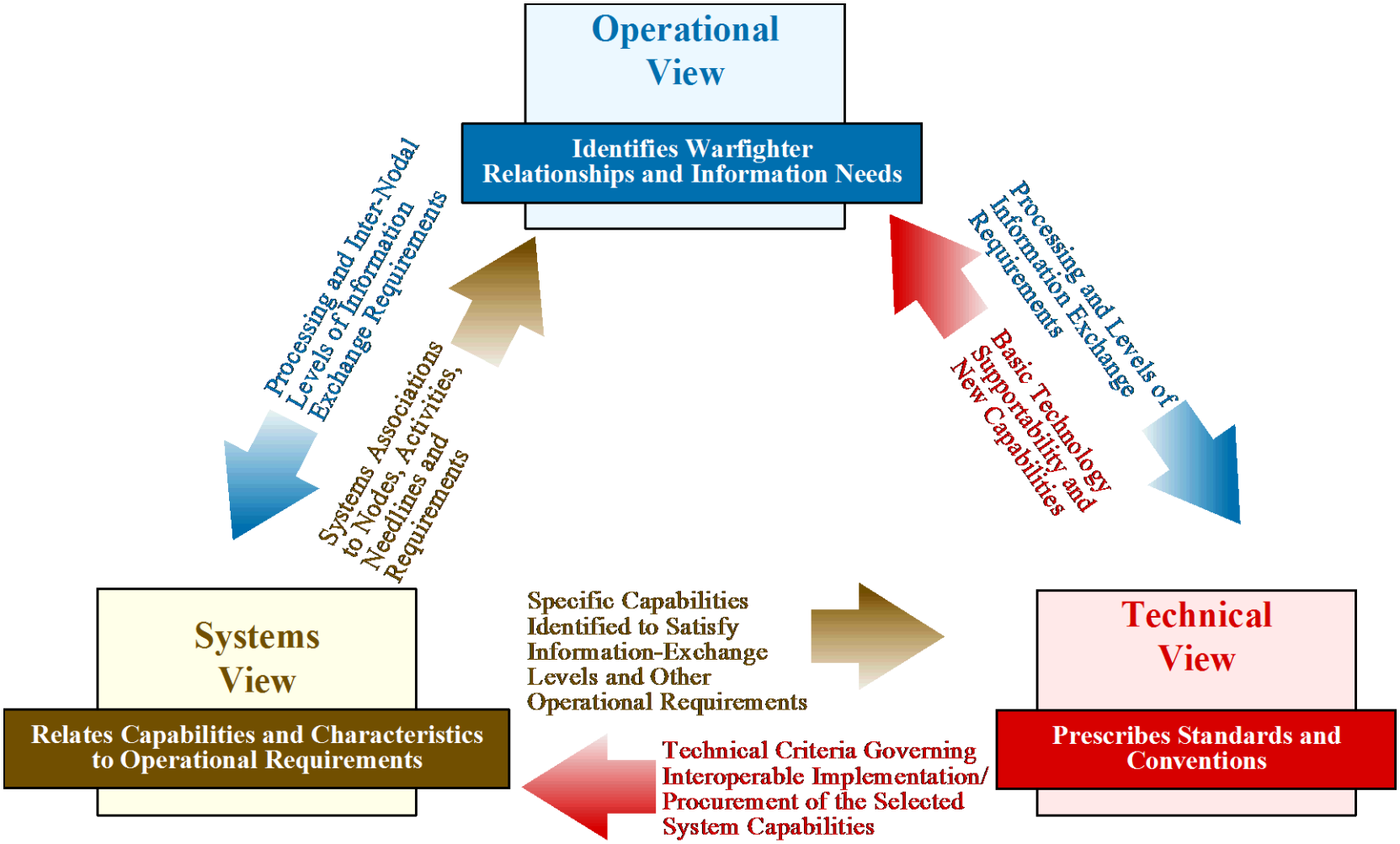
Source: T. J. Williams, A Handbook on Master Planning and Implementation for Enterprise Integration Programs, Institute for Interdisiplinary Engineering Studies, Purdue Univ.

CSC_PLM



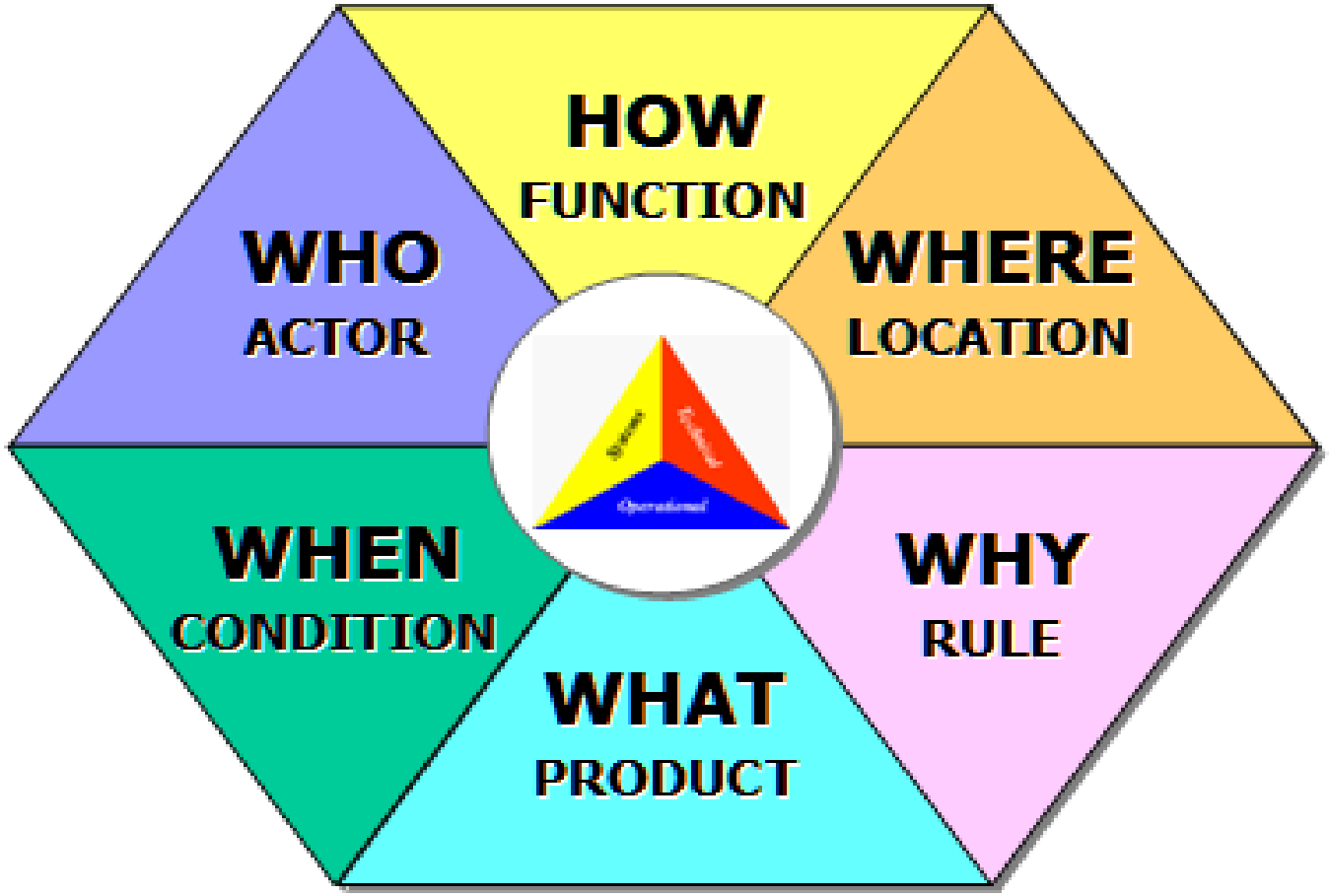
Verband Deutscher Maschinen- und Anlagenbau - German Engineering Federation

DoDAF



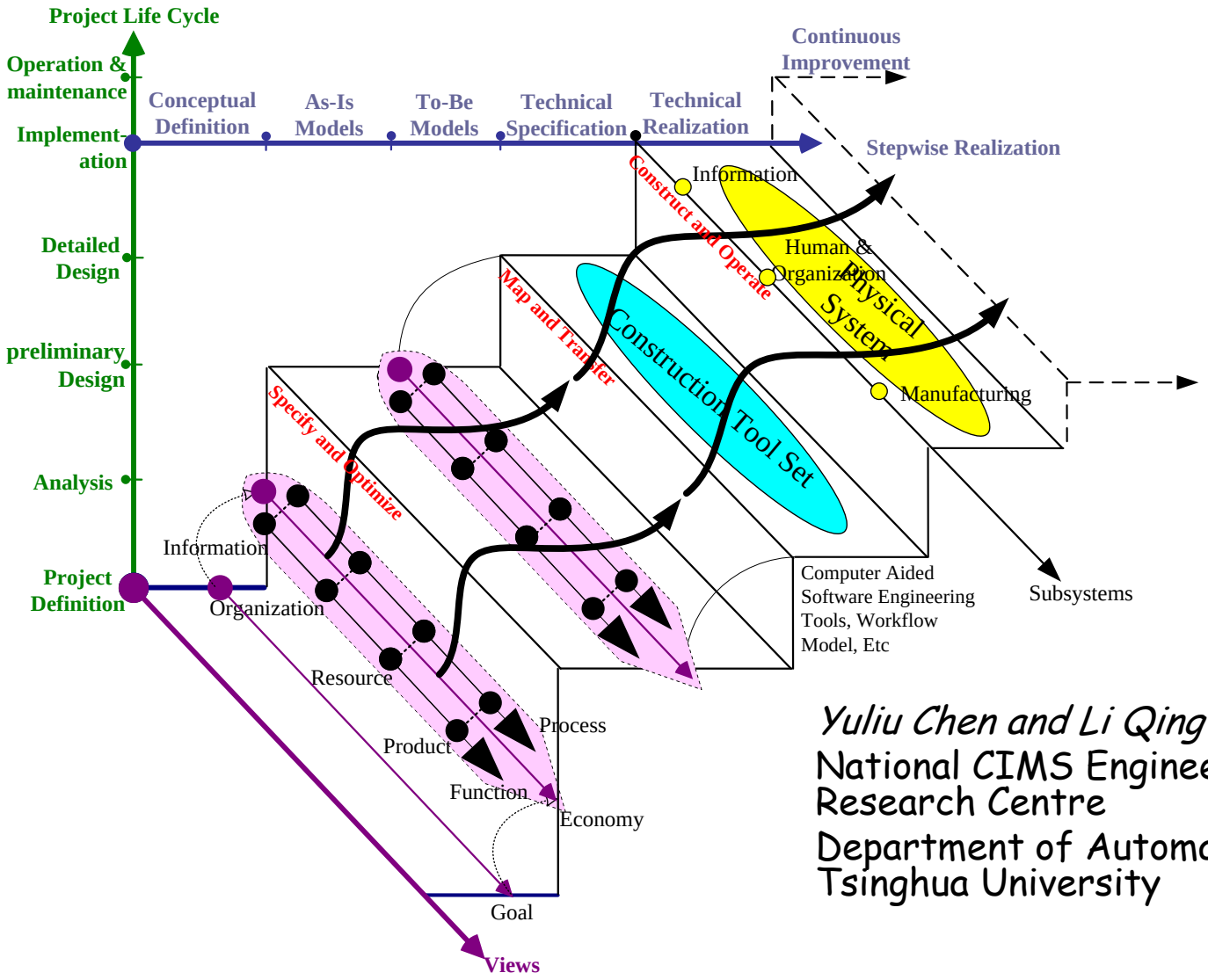
Source: Architecture Working Group, C4ISR Architecture Framework Version 2.0, 1997

DoDAF redux



DoDAF v2.0 Architecture Conceptual Data Model

Stair-like Architecture

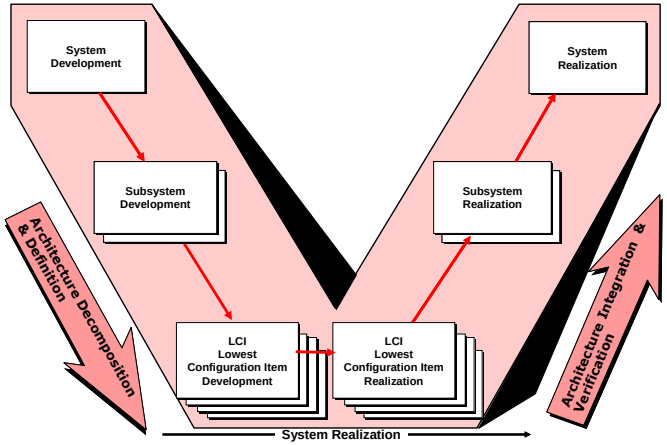


Yuliu Chen and Li Qing
National CIMS Engineering and Research Centre
Department of Automation, Tsinghua University

Mooz and Forsberg

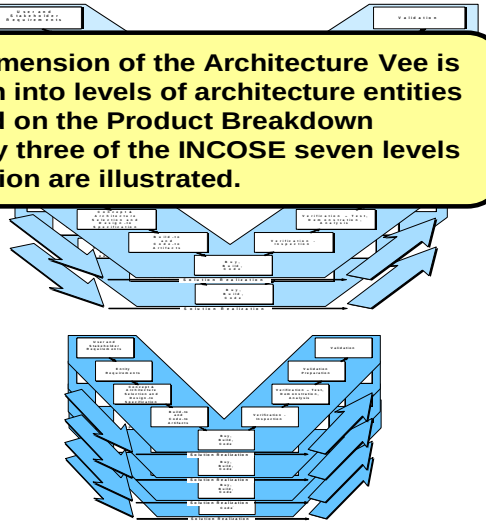
Dual Vee Model

Architecture Vee
for architecture management



Depicts architecture baseline evolution. Vertical dimension is architecture decomposition. Horizontal dimension is system realization. Third and normal dimension is quantity of entities and their interfaces.

Entity Vee
for entity management

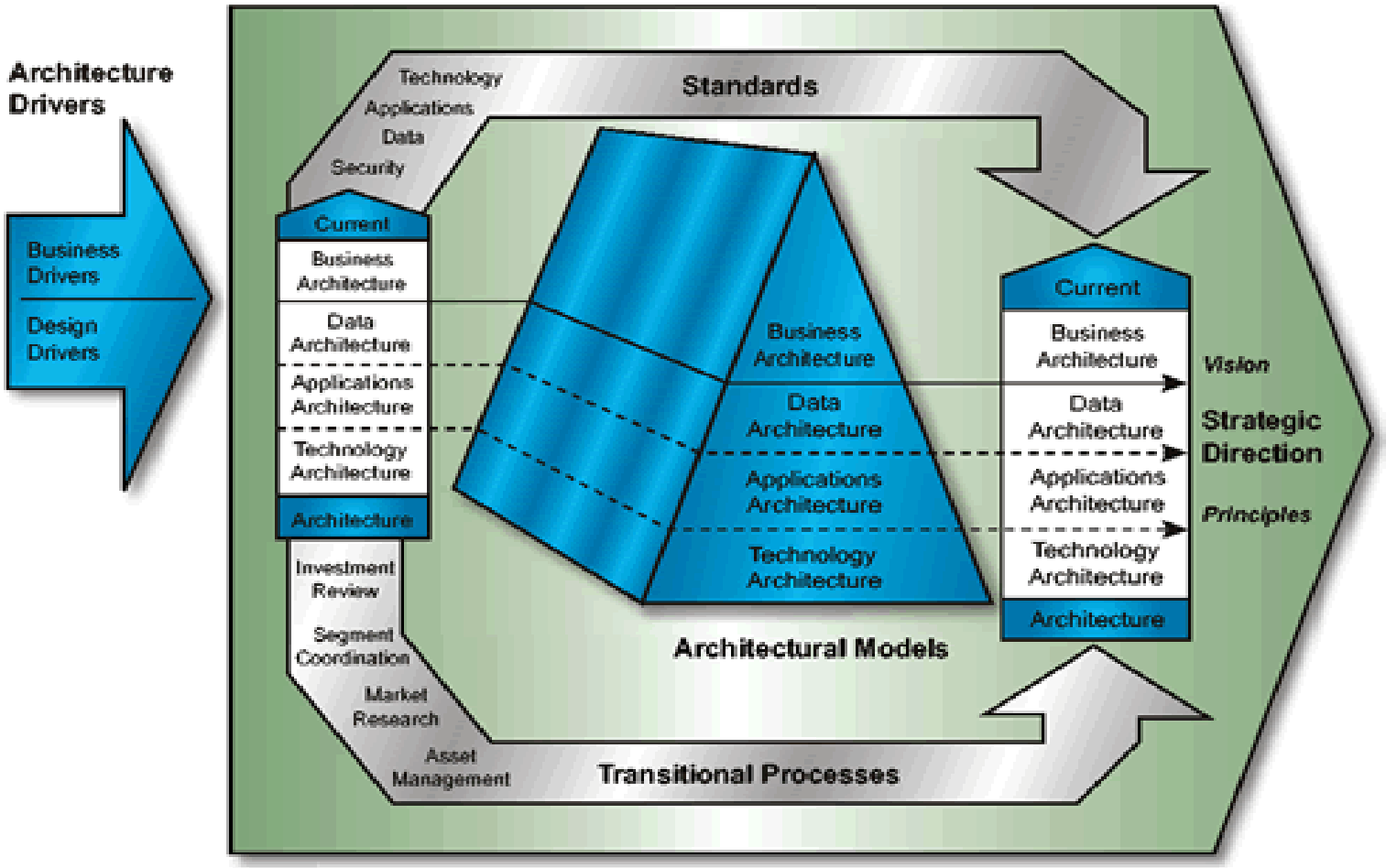


The vertical dimension of the Architecture Vee is decomposition into levels of architecture entities which is based on the Product Breakdown Structure. Only three of the INCOSE seven levels of decomposition are illustrated.

The vertical dimension of the Entity Vee is extent of elaboration detail at the decomposition level of interest such as Subsystem or Lowest Configuration Item. The elaboration includes the baselines of concept, architecture, design-to, build-to, code-to, as-built, etc.

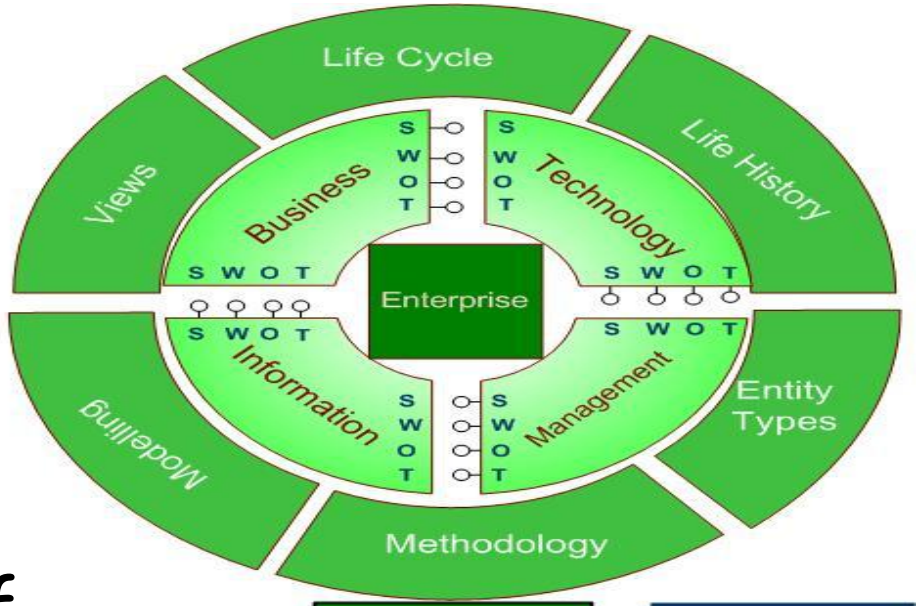
CSM THE CENTER FOR SYSTEMS MANAGEMENT

FEAF

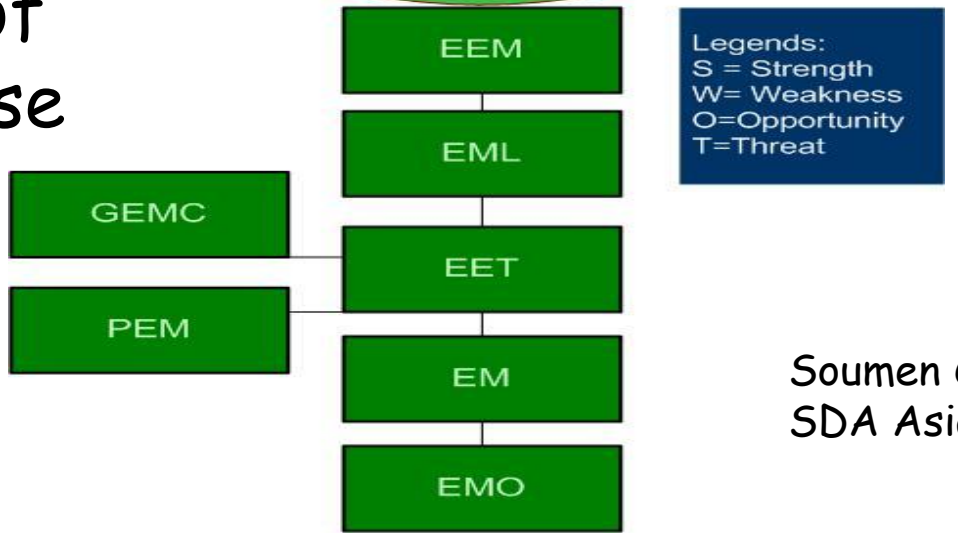


Federal Enterprise Architecture Framework

Key to the Puzzle

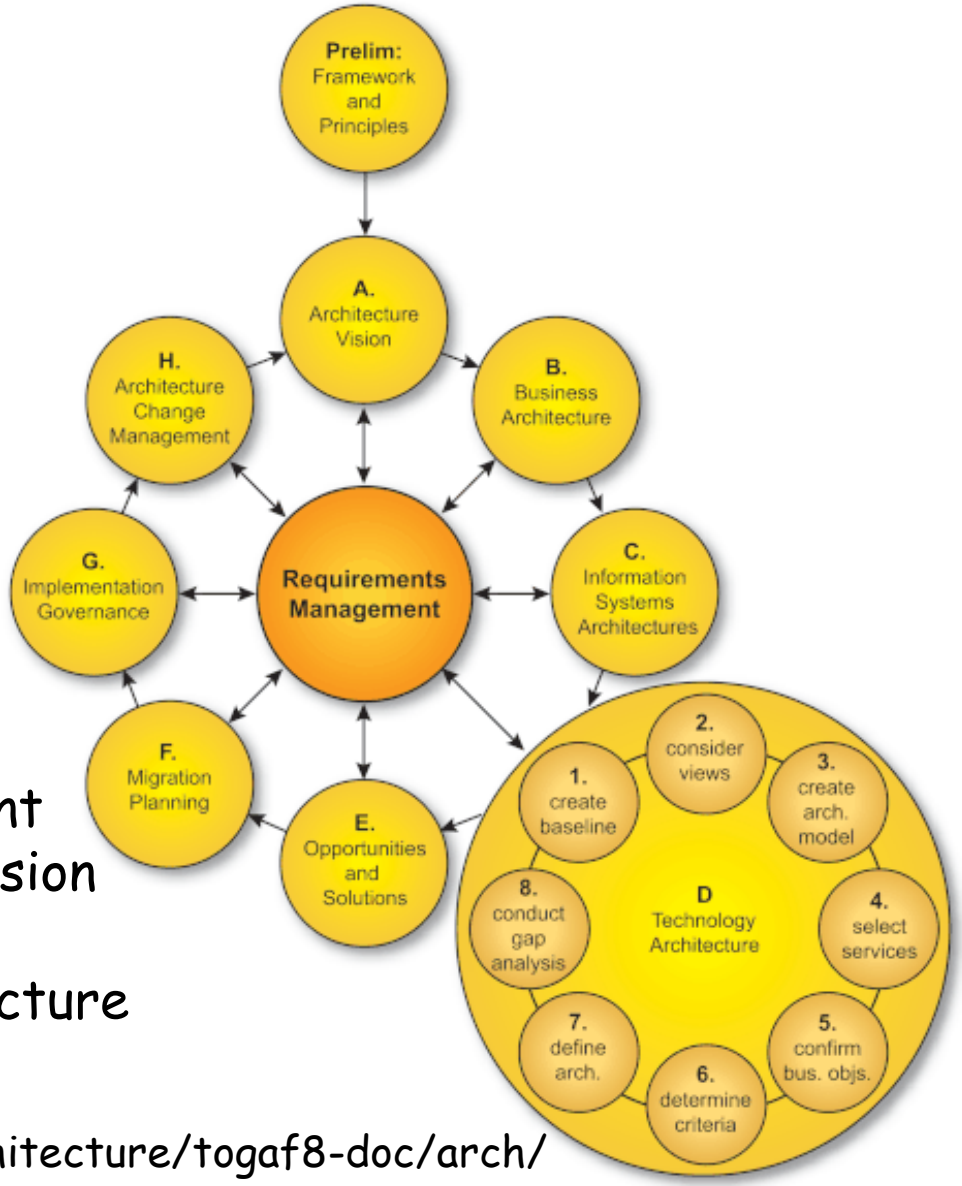


Master key of the Enterprise



Soumen Chatterjee, SDA Asia 2006

TOGAF

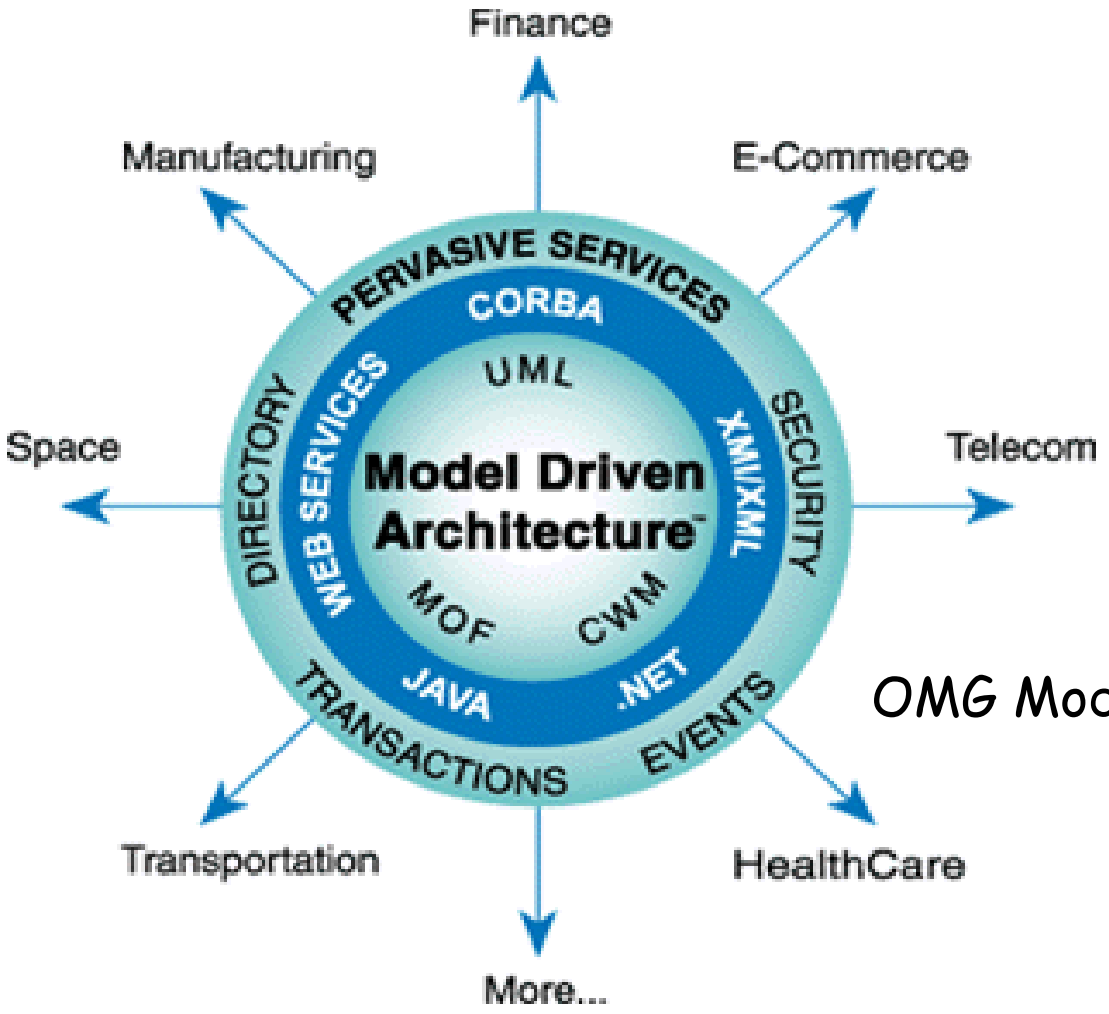


Architecture Development Method cycle with expansion

The Open Group Architecture Framework v8.1.1

http://www.opengroup.org/architecture/togaf8-doc/arch/

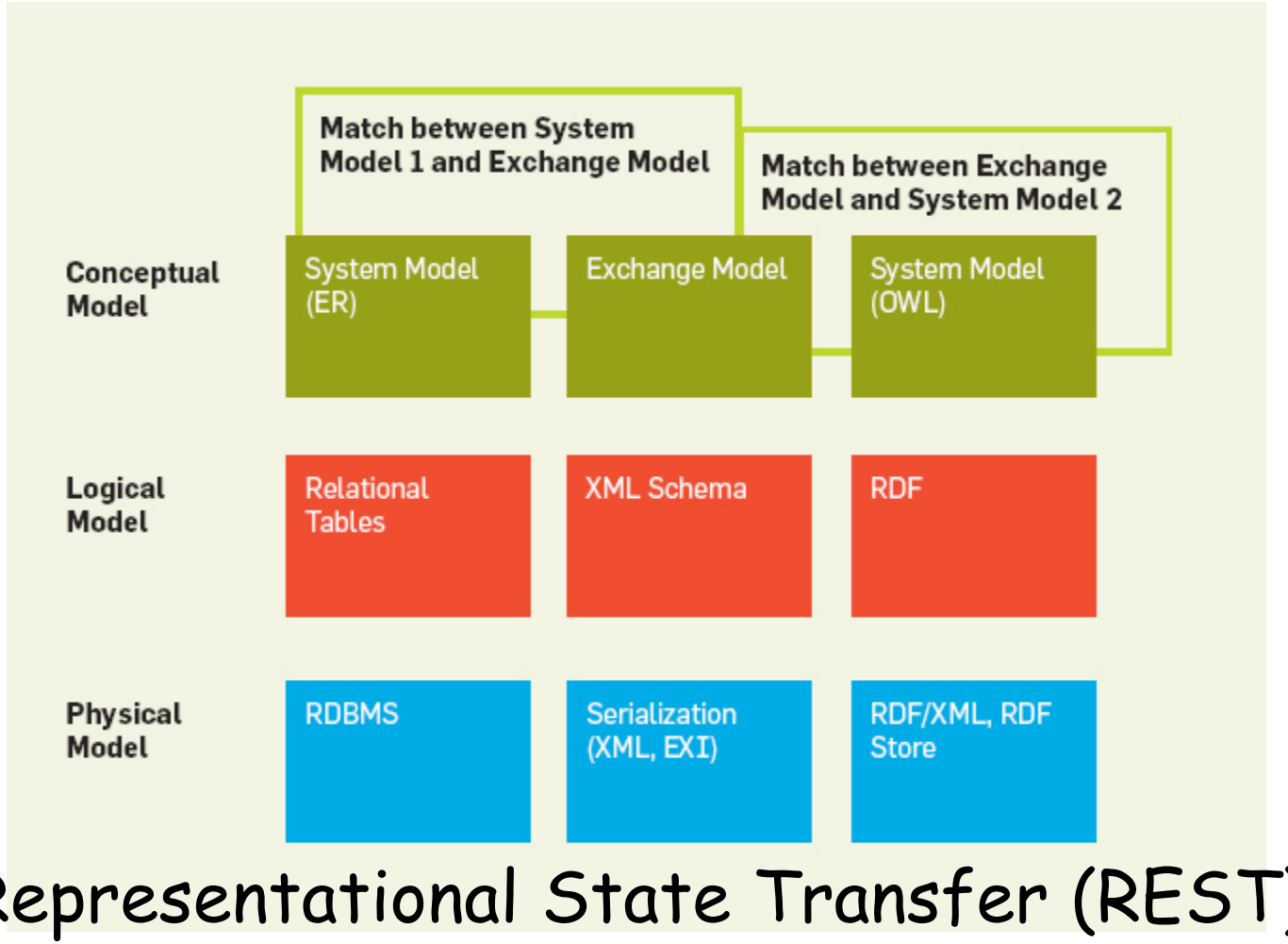
MDA



OMG Model Driven Architecture

http://www.omg.org/mda/

REST



Representational State Transfer (REST) architecture modeling layers

Document Design Matters, Wilde,E., Glushko, R. J., Communications of the ACM, Vol. 51, No. 10, Oct. 2008

Synergetic efforts

Liaison

– a channel for communication between groups

Observation - a critical pattern within the architecture of international standards

Liaisons

Liaisons help to expand the reach for input and validation

Internal (within ISO, like SC4 & JTC1)

External (beyond ISO national bodies)

- IEC, CEN, etc.
- INCOSE, OMG, ISA, OASIS
- Experts from industry and academia

A place for anyone who wants to participate

INCOSE liaison

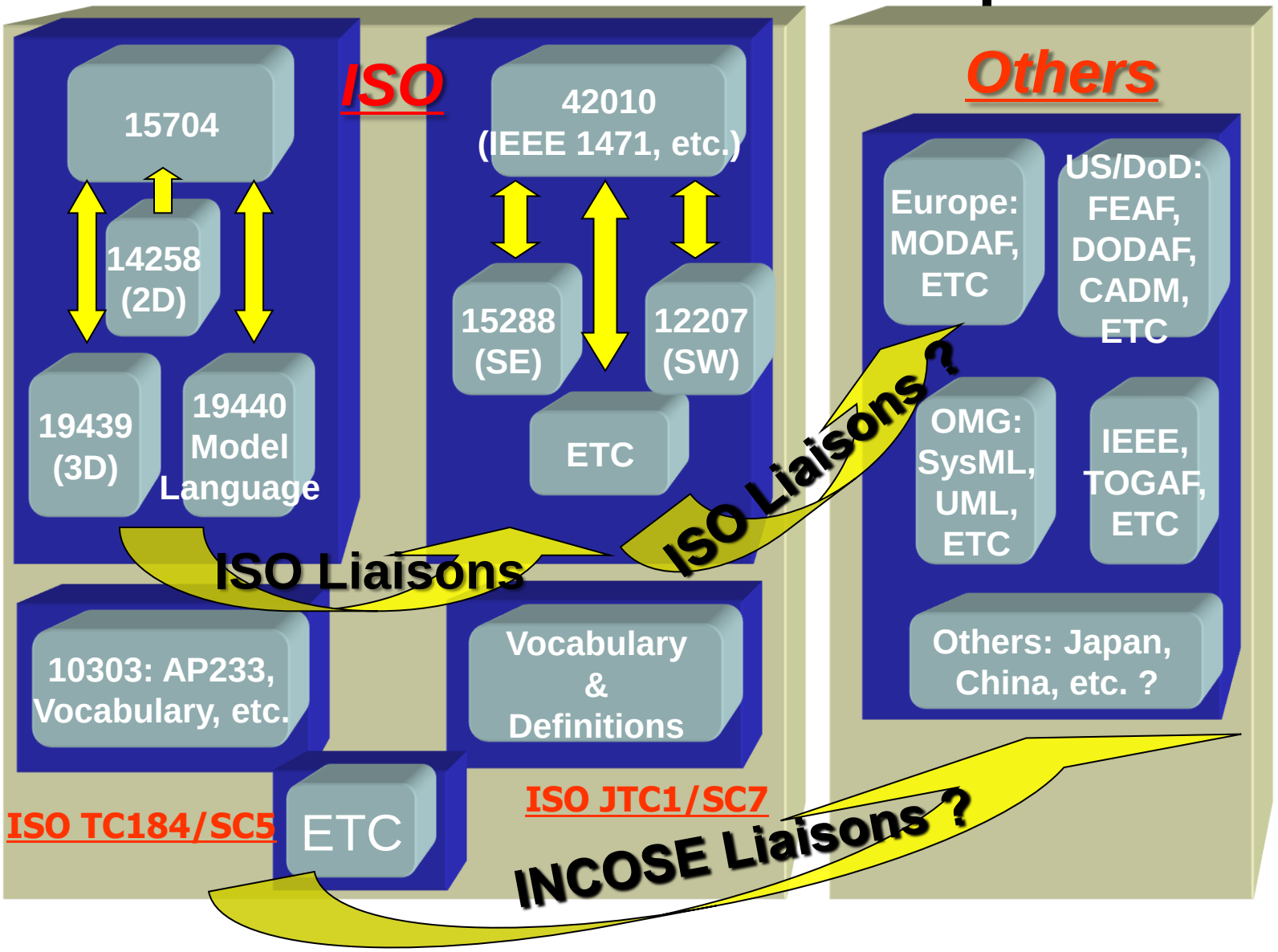
INCOSE Standards Tech Committee (STC)
- Coordinate INCOSE WG involvement, tasks, etc.
- STC Liaisons: JTC1/SC7 & TC184/SC5

Other INCOSE WGs involved:
- Architecture - Charles Dickerson
- Integration & Interoperability - John Nallon
- Model Based SE- Phil Spiby

INCOSE Connect webpage established

INCOSE INSIGHT articles (Jan & Apr 07)

INCOSE Relationships



Break check

- What's the situation now?

Principles for architecture and models of architecture

Mindful of Principles

General Principles

1. Models are formal artifacts developed and used by people.
2. A complexity tradeoff exists between modeling medium and model instances in that medium.
3. Naming serves as the bridge between the formal and the human.
4. Do not confuse meta-levels - separate model and instance decompositions.
5. Dependency is not chronology
6. Don't hide architecture in methodology.

More Principles

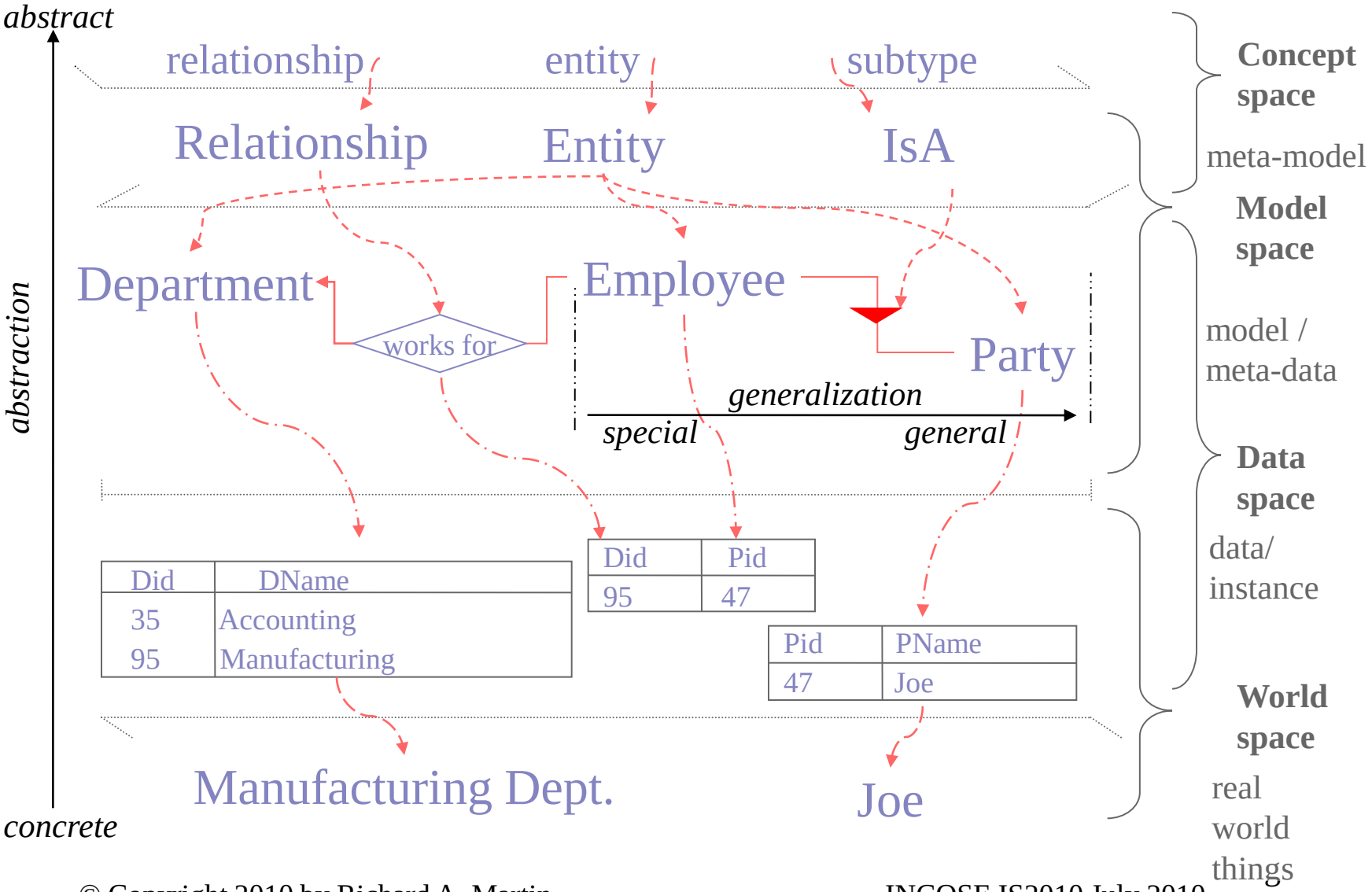
Framework Principles

7. Frameworks organize artifacts to facilitate understanding.
8. To improve quality, distinguish structure from connectivity.
9. Separate policy from mechanism.
10. Both grid (ordinant) and tree (decomposition) structures appear in models.
11. Scale dimensions include:
    abstractness (abstract to concrete),
    refinement (coarse to fine) and
    scope (general to special)

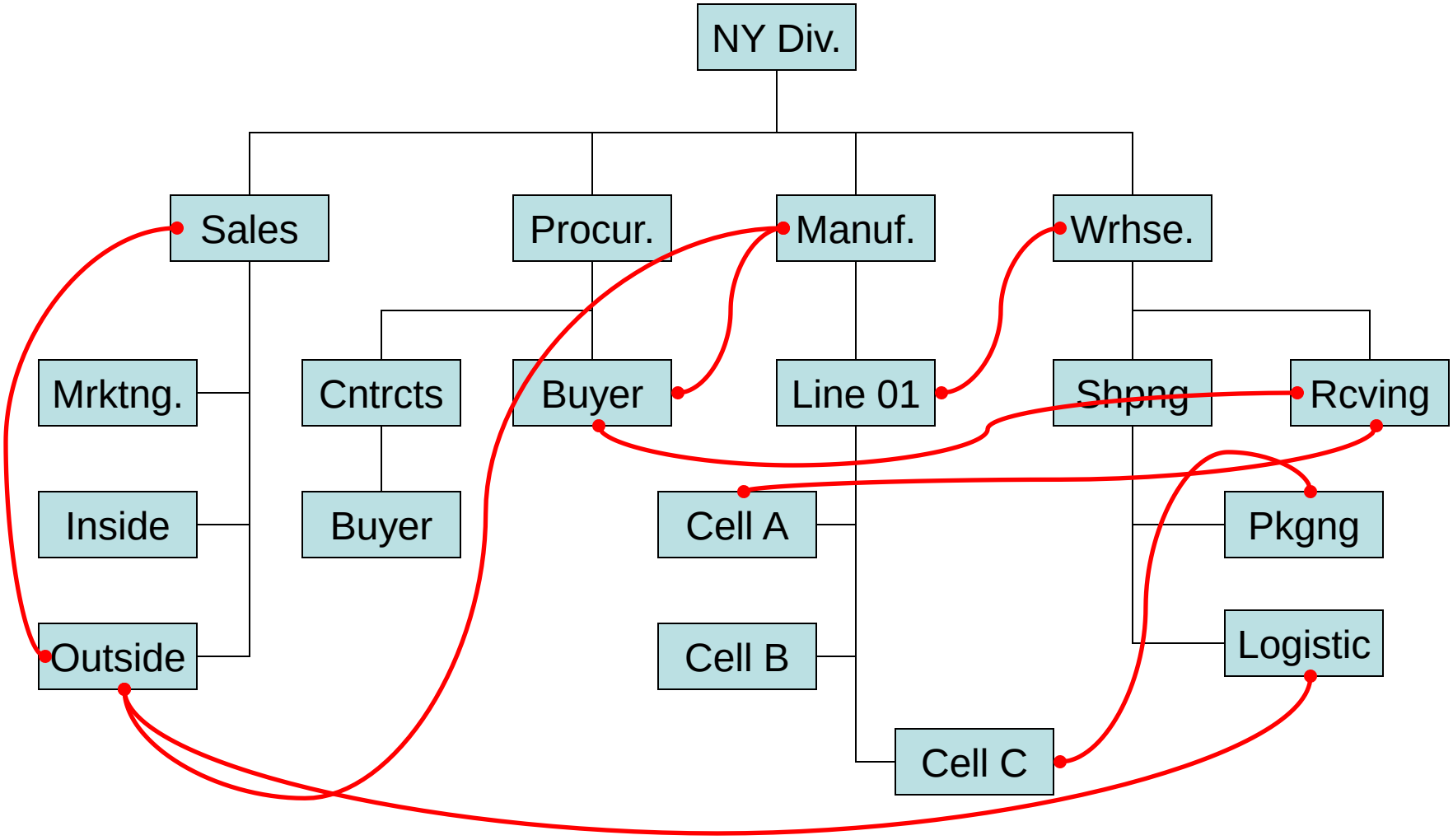
More Principles - 2

12. Within a framework, use of components are driven along one ordered dimension.
13. Along this ordered dimension, all prior context is relevant.
14. Refinement is recursive using iteration.
15. Connections can be of arbitrary arity.
16. Views are important in standards and methodologies.
17. Views are used both to "see" contents and to "create" contents.
18. Separate model and instance constraints.

Scope & Meta-confusion

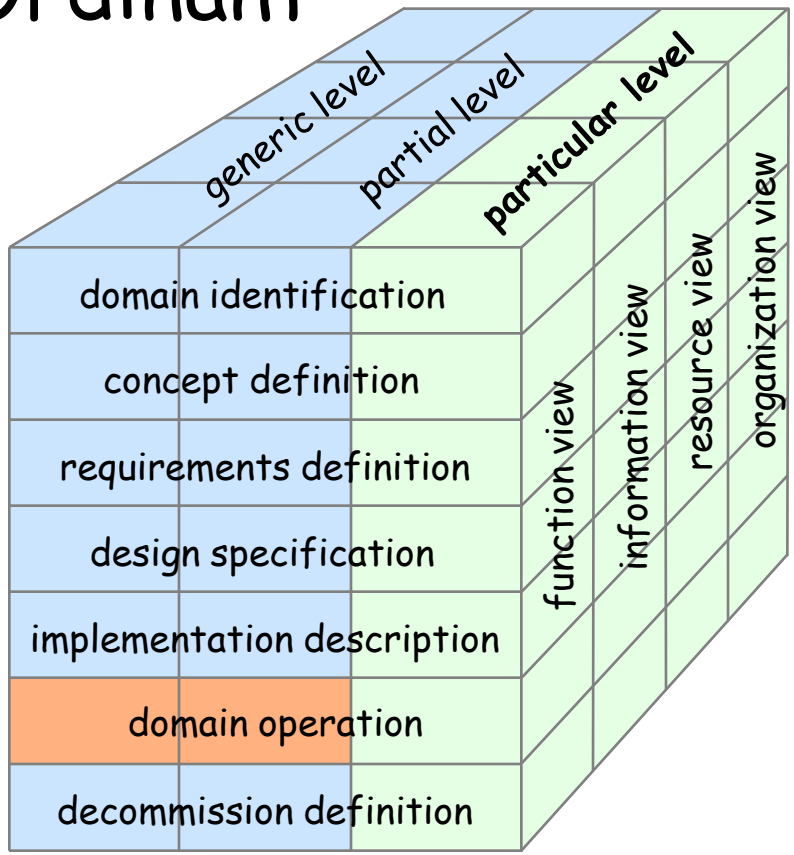


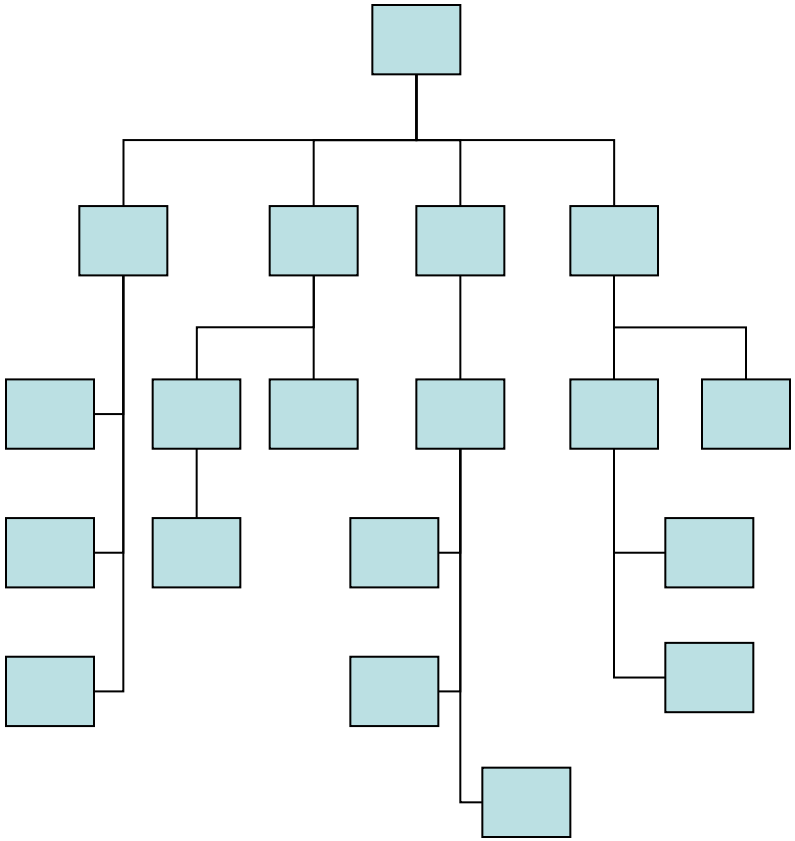
Distinguish structure from connectivity



Two structural aspects

Ordinant





Decomposition

Purposeful dimensions

Zachman: **Role**

{Context, Owner, Designer, Builder, Out-of-context}

ISO 19439: **Model Phase**

{Domain, Concepts, Requirements, Design, Implementation, Operation, Decommission}

ISO 15288: **Process Group**

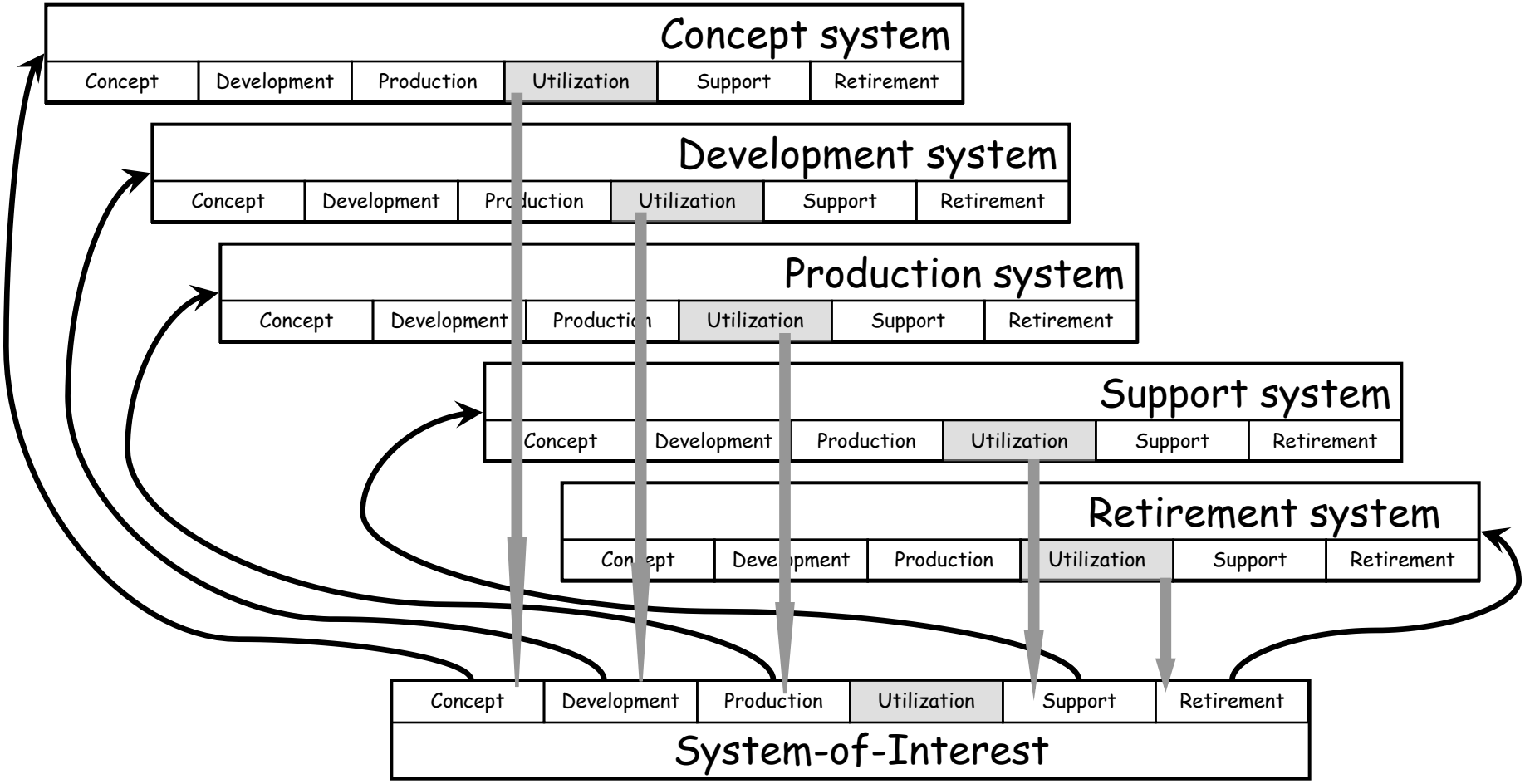
{Agreement, Enterprise, Project, Technical}

C4ISR/DoDAF: **Guidance**

{Focus, Scope, Characterize, Determine, Build, Use}

Recursive refinement

cf. ISO 15288:2002



Views

Views are for communication and analysis.

A static collection of views is insufficient.

Views exist at all meta-levels.

View of structure is meta with respect to view of data.

View update often crosses meta-levels.

Standards sometimes specify a view using a viewpoint.

Observation points

Distinguish a physical viewpoint from a conceptual viewpoint

Terminology wars

No one speaks the way you do

- Different training
- Different disciplines
- Different customs
- Different translations

A large global upper ontology is a myth

Local taxonomies and meaningful phrases are achievable standards

Precision is more important than recall

Extent of architecture

- No agreement on what is an architectural concern and what is not – it is situation and purpose dependent
- Certainly more than simple allocation of form and function – recall *venustatis* - and thus the focus on stakeholder concerns as *fit for intended use*
- Architecture is the trade-off space for requirements while engineering is the trade-off space for implementation

(Don't confuse titles with tasks – its all design)

Limits of architecture

The Intension/Locality Thesis (Eden & Kazman 2003)

Architecture	*Intentional*	*Non-Local*
Design	*Intentional*	*Local*
Implementation	*Extensional*	*Local*
Trouble	*Extensional*	*Non-Local*

Architecture formally

A specification **S** is intentional iff there are infinitely-many possible instances thereof. Conversely, all other expressions are extensional

A specification **S** is local iff the following condition holds:

If **S** is satisfied in some design model **m** then it is satisfied by every design model that subsumes **m**.

Architecture or not?

Consider two views of a system, S: with hardware view, HW(S), & software view, SC(S).

Assure every application has a platform and every platform has an application:

$$ExecutesOn \subseteq HW(S) \times SC(S) \text{ s.t.}$$
$$\Pi_1(ExecutesOn) = HW(S) \text{ and}$$
$$\Pi_2(ExecutesOn) = SW(S)$$

If SC(S) has elements, e1, … e6, and HW(S) has platforms, p1, … p4, express a model for which software elements execute on which platforms:

ExecutesOn (R1) = { (e1, p1), (e1, p4), (e2, p2), (e2, p3), (e3, p3), (e4, p4) }

MOSES Boundary

Hybertson – Model Oriented Systems Engineering Science, June 2009, CRC Press

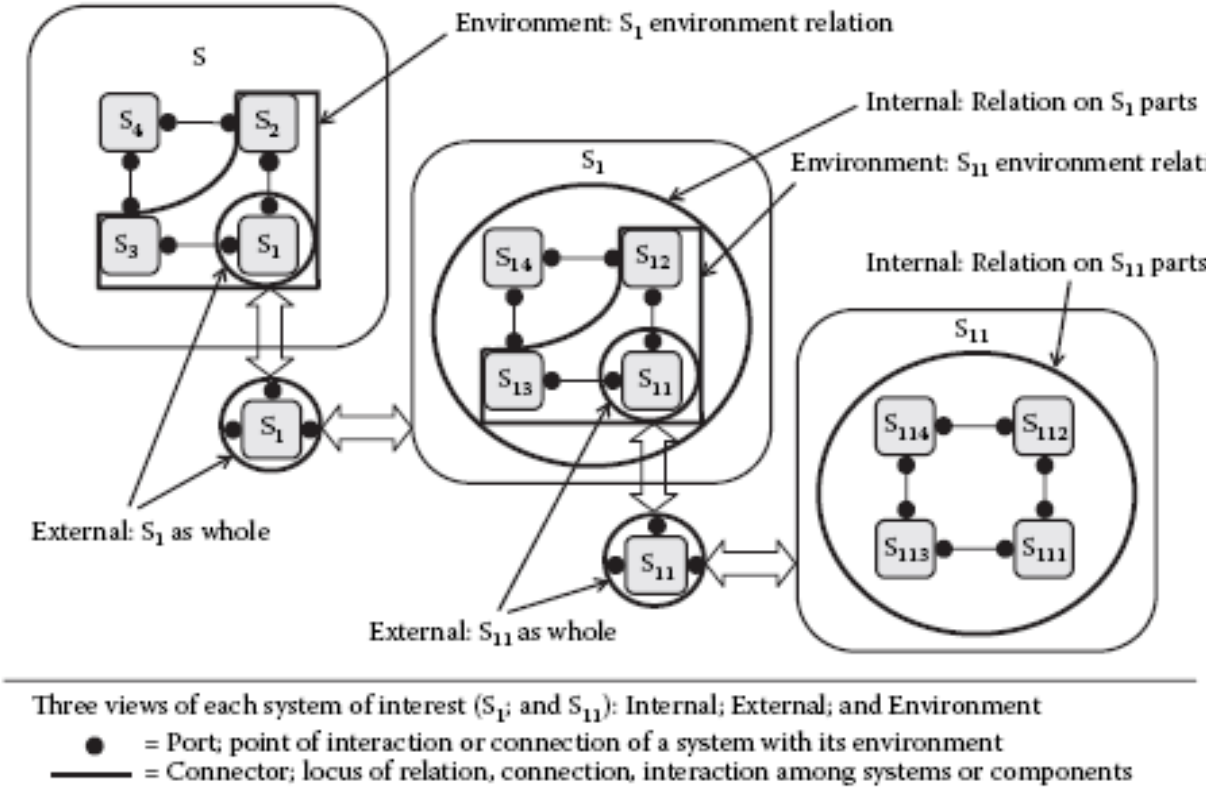


Figure 8.1 SE context for multiple domains, languages, and ontologies.

Morning Break

What have I said about architecture, systems, and enterprises that does not fit with your understanding?

In the previous MOSES slide, do the boundaries shown correspond to those of the boundary object and uses identified earlier? (see slide 9)

ISO/IEC 42010:2007 Architecture description

ISO 42010:2007 (formerly IEEE 1471)

- Developed by the IEEE's Architecture Working Group under the sponsorship of the Software Engineering Standards Committee of IEEE
- Effort began in 1995 with large working group and reviewer group
- Focus on best practices and a vocabulary for architecture concepts

Whence cometh 42010

- First approved by IEEE Standards Board in 2000 followed by ANSI as an American National Standard in 2001
- Fast track approval by ISO/IEC in 2007 as 25961 and relabeled 42010 at time of publication
- Under revision to harmonize with JTC1 standards and other ISO standards

ISO 42010 Scope

- Expression of the system and its evolution
- Communication among the system's stakeholders
- Planning, managing, and executing the activities of system development
- Planning, managing, and effective utilization of a system's elements and resources throughout its life cycle*

* Added to scope for revision draft

More 42010 Scope

- Evaluation and comparison of system architectures in a consistent manner
- Expression of the persistent characteristics and governing principles of a system to guide acceptable change
- Verification of a system's implementation for compliance with an architectural description
- Recording contributions to the body of knowledge of systems and software architecture.

Big ideas from 1471

1. Architecture exist to satisfy known concerns from stakeholders
   - Ensures architecture and its description are relevant
   - Stakeholder concerns, often non-functional, drive the architecture
2. Architecture Descriptions are inherently multi-view
   - No single view addresses all concerns
   - A view should cover the entire system

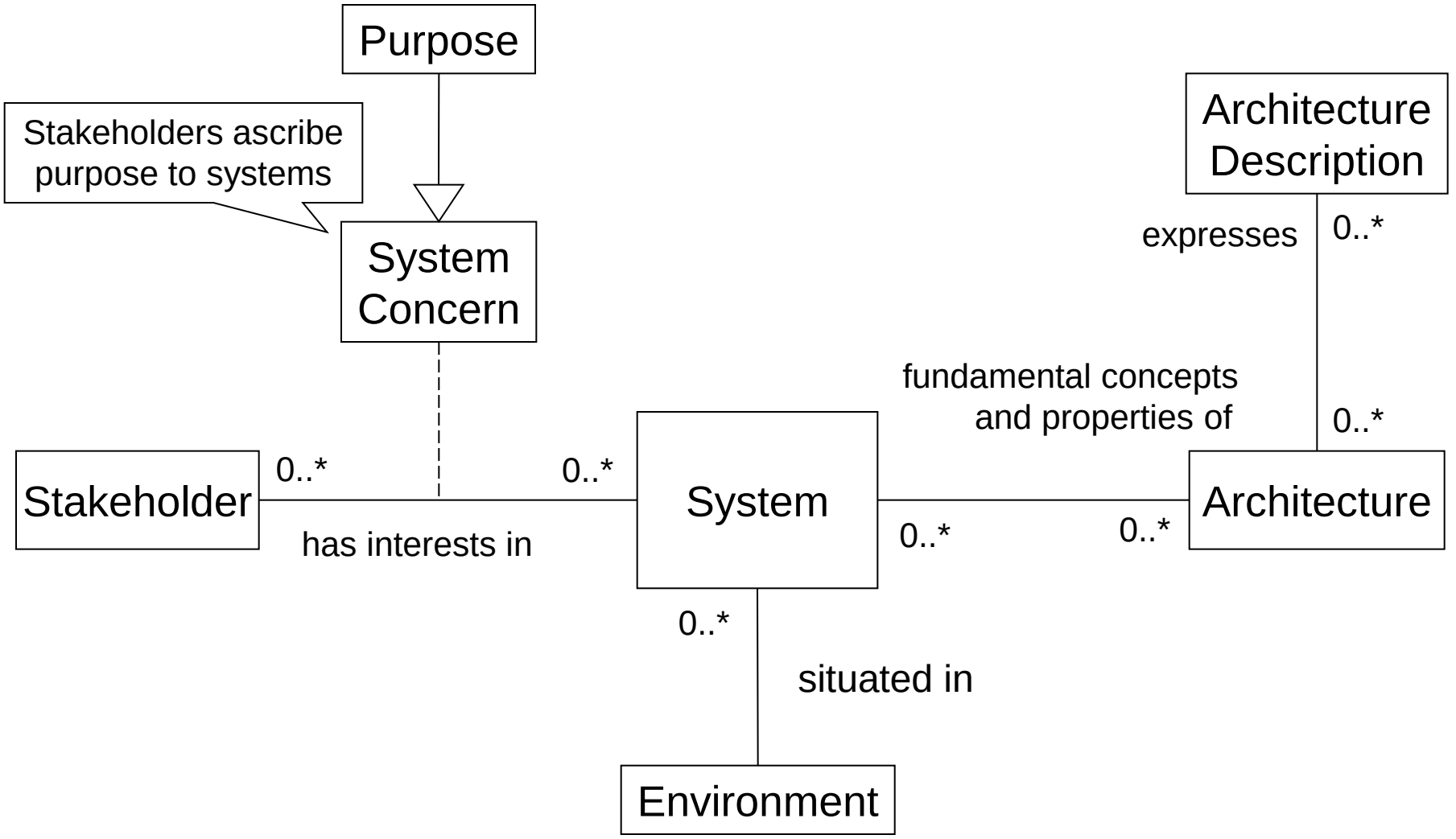
Third 1471 idea

3. Viewpoints ('what to describe') are separate from Views ('this description')
   - Represents current practice with 'viewpoint sets'
   - Ensures consistency and repeatability, particularly when evaluating alternative architectures
   - Supports development of architecture tools, techniques and methods

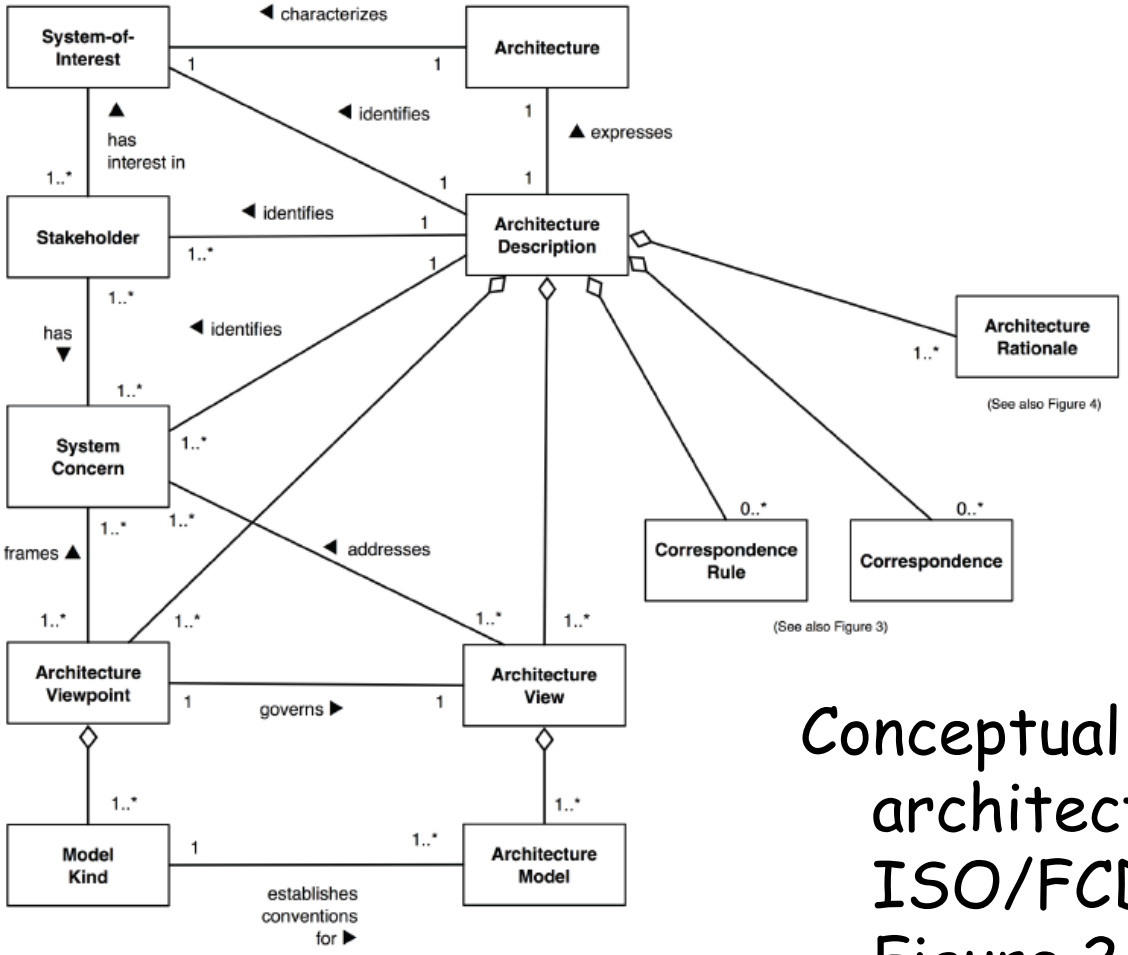
Limits of 42010

- All about (don't extend expectations)
  - a single architecture description of
  - a single architecture of
  - a single system
- No specification of notation, format or media
- No required content of an architecture description reflecting current practice and consensus
- Conformance to the standard is with respect to a point in time

42010 Conceptual space



42010 FCD AD model



Conceptual model of architecture description ISO/FCD ISO 42010 Figure 2

42010 requirements

- Stakeholders (relevant to architecture)
  - Architectural description must explicitly identify the system's stakeholders
  - Two key stakeholder roles are acquirer and architect
- Concerns (relevant to architecture)
  - Interest of stakeholder in system development, operation or other aspect
  - Include system consideration such as performance, reliability, security, distribution, and evolvability
  - Drive viewpoint selection

42010 viewpoints

- Viewpoints are first-class, i.e., they are 'declared' before use
- May originate in an AD or elsewhere
- Establishes the conventions by which a view is created, depicted and analyzed
- Determines languages and associated modeling methods
- AD includes rationale for a viewpoint
- Each stakeholder & concern is addressed by at least one viewpoint

People may expect a different notion of "viewpoint"

42010 models

- Viewpoint specifies model elements
- Architectural models are developed using the methods established by associated viewpoint(s)
- Models have many forms and manifestations – physical, logical, etc.
- Architectural models are the constituents of architectural views
- May participate in more than one view

Importance elevated in revision version

42010 views

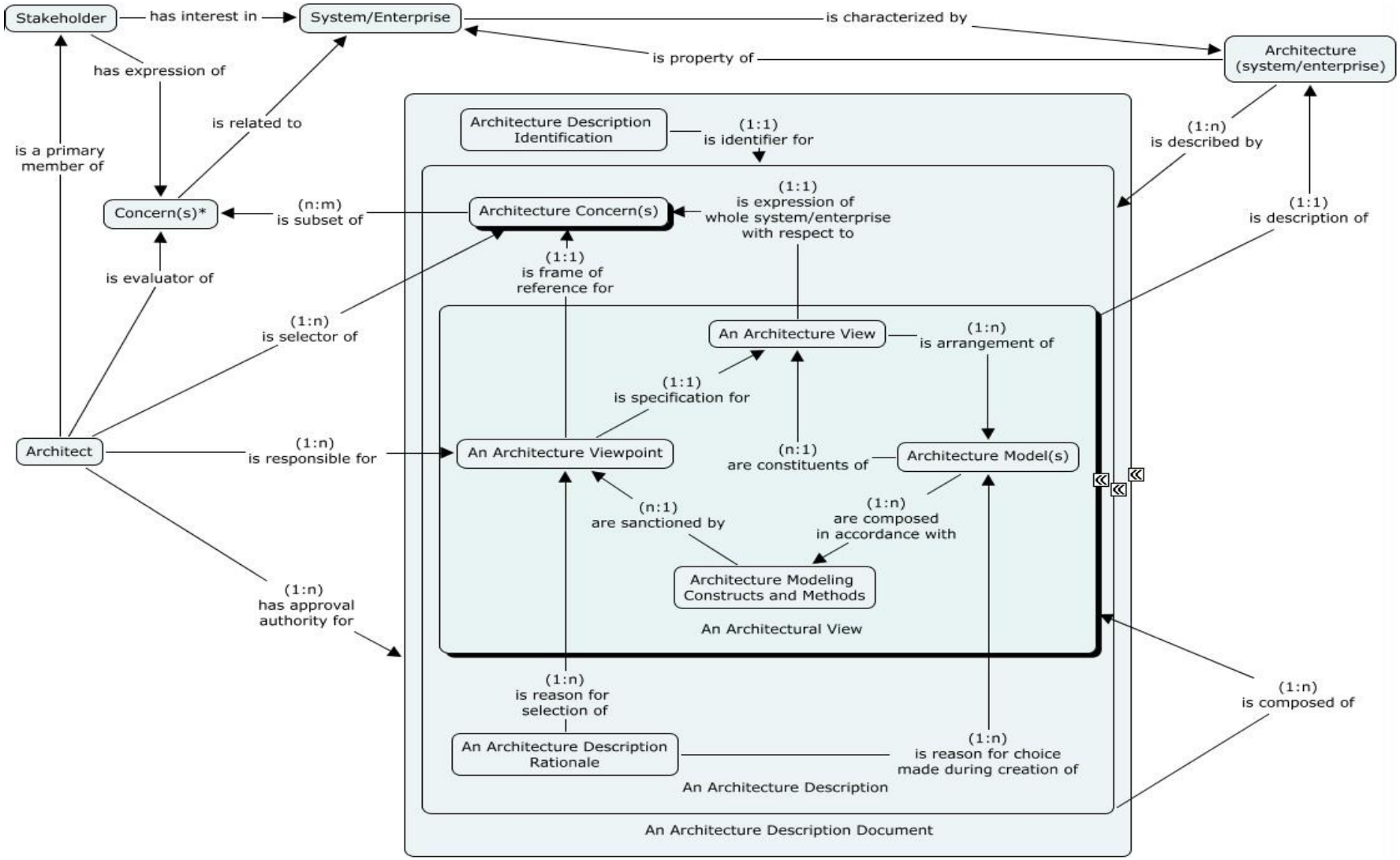
- The stakeholder perspective of architecture (like Zachman)
- Each architectural view spans the whole system of interest with respect to one or more concerns
- May consist of one or more kinds of architectural model
- The AD is composed of architectural views and supporting information, particularly rationale

42010 correspondence

- Each view expresses exactly one viewpoint
- The set of views in an architecture description corresponds to a complete allocation of concerns
- AD should contain analysis of consistency across all of its architectural views

Revision version adds notion of correspondence and correspondence rules between AD model entities

Digging Deeper



42010 frameworks

- Architecture framework identifies architecture-related concerns, stakeholders holding those concerns, and one or more architectural viewpoints that frame those concerns
- May define viewpoint correspondence rules to relate its viewpoints
- Any architecture framework meta-model shall reflect the 42010 Core Model
- AD conforms to AF if and only if AD views correspond to AF viewpoints

ISO 42010 Conformance

Conformance requires that:

- For each architecture description, all views, viewpoints, concerns and stakeholders are properly identified
- Each concern is addressed by at least one viewpoint
- Each view corresponds to exactly one viewpoint
- Models consistent with the viewpoint compose the view associated with that viewpoint
- Inconsistencies between views are explicit
- Rationale for architectural choices is provided

Frameworks for enterprise architecture and architecture modeling

Comparing frameworks

One set of viewpoints

Framework Principles

- Structure, Connections, Views, Constraints

Usage Observations

- Prototypes, Time, Purpose

Archetypes

- Zachman, ISO 15704, ISO/CEN 19439, ISO/IEC 15288

Complements

- Prototypes, Purpose, Artifacts, Change

What is a Framework?

A containment structure

- context for model artifacts
- interconnections between models
- access to model components
- model fidelity and consistency

NOT a programming framework,
but maybe a 42010 framework.

Structure

A space of one or more dimensions

meta-model:

Arrangement

- Ordinant (label) - Ordered, Unordered
- Decomposing (path)

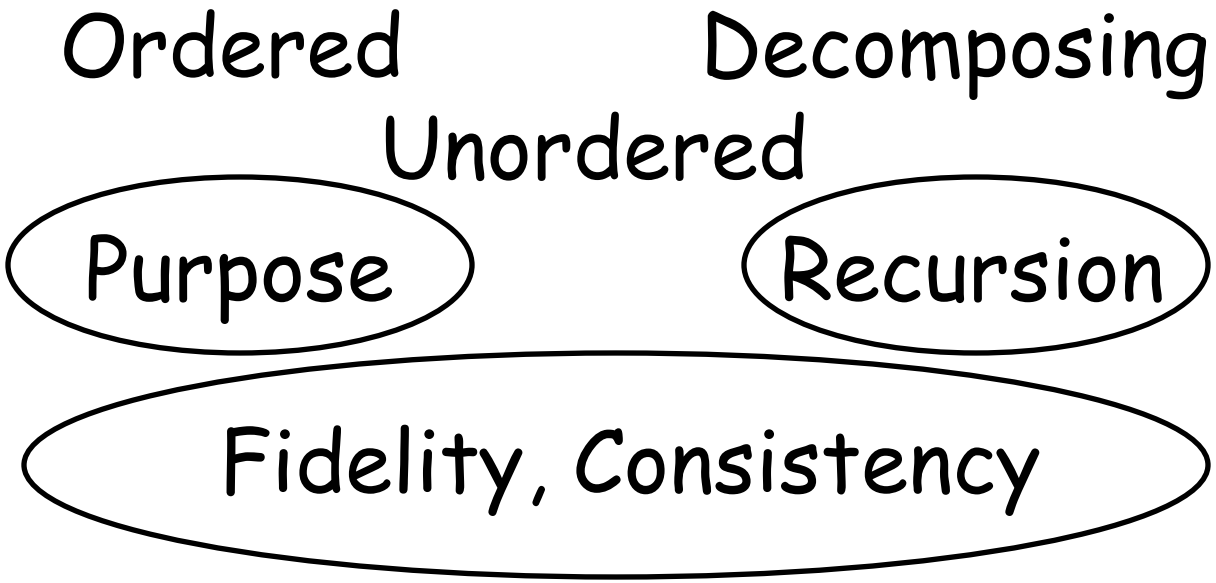
Scale

- Scope (general to specific)
- Abstract (abstract to concrete)
- Detail (coarse to fine)

Connections

Structural linkage along and among dimensions

meta-model:



- Dependence
- Equivalence
- Transitivity

Views

Different ways of looking at artifacts

meta-model:

- Filter along a dimension
- From one dimension to another
- Rearrange a framework – derive a view
- Use selection and projection

Formal meta-model harder than mechanism

Constraints

Evaluate conformance to a standard

meta-model:

- Structure – a place for everything of interest
- Connection – within and between dimensions, typically binary
- View – something must be placed to be seen, often used to define constraints
- Distinguish model from instance constraints
- Formal mechanisms within one model

Artifact Prototypes

- Frameworks are conceived with prototype artifacts in mind
- Framework artifacts are models we build both formally and informally
- Frameworks partition artifacts along conceptual categories (dimensions) with coordinates and paths
- Prototypes range over all enterprise aspects – automated, mechanical, human
- Framework expression is the realized model instances derived from prototype artifacts

Entities in Time

The characterization of a framework with respect to time informs us about the nature of change in the framework's context.

- **Continuant** - identity continues to be recognizable over some extended interval of time
- **Occurrent** – identity is not stable during any interval of time.

(see SOWA)

Continuants / Occurrents

- **Continuants** are wholly present (i.e., all their parts are present) at any time they are present.
- **Occurrents** just extend in time by accumulating different temporal parts, so that, at any time they are present,they are only partially present.
- **Continuants** are entities that are in time. Lacking temporal parts all their parts flow with them.
- **Occurrents** are entities that happen in time. Their temporal parts are fixed in time.

(see Masolo, Borgo, Guarino, et. al.)

Why does it matter?

- We expect descriptions of architectures, frameworks, and models to be reasonably stable over time – like continuants.
- But systems, and especially enterprises, are never stable for very long – they are occurrents.
- Therefore we must recognize the limits of descriptions to aide in managing a changing world.

Enterprise Description

- Enterprise as product is continuant
- Enterprise as process is occurrent

- Purpose emerges from an ordered dependency
- Dependency is not necessarily chronology

- Purpose can be found in both continuant and occurrent enterprise descriptions
- Frameworks address continuant and occurrent purposes in enterprise description – but a single framework cannot do both!

Architecting in time

- Architecture evolves over time
  - new product development
  - revision to meet a market condition
- Architecture at all levels evolve
  - better way found to articulate architecture
  - ad-hoc gives way to disciplined approach
- Architecting is an enterprise

Layers of Architecting

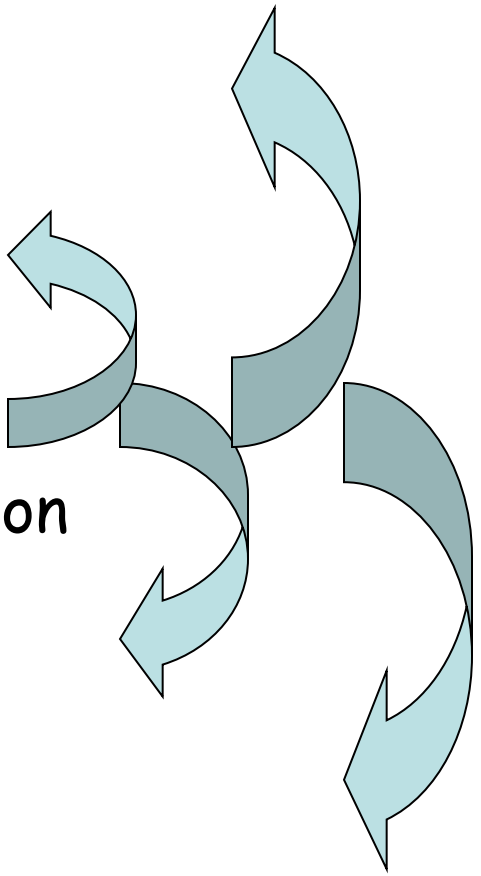
- 1st – applying a meta-architecture (e.g. a framework) to create and use an architecture description for the Enterprise
- 2nd – creating the meta-architecture for use in 1st layer activities
- 3rd – a meta-meta-level that describes evolution of 'architecture' and includes changes to the 2nd level meta-architecture

1st layer evolution of utility

- As architecture description (**AD**) evolves it serves two distinct stakeholder communities
  - One delivering more abstract concepts that address their concerns
  - One expected to take the elaboration further toward a less abstract, more elaborated specification
- Elaboration evolves architecture from vague concept to formal descriptions for use by designers of enterprise details

Stakeholder community grows

- Line of business manager
  - Market Opportunity Assessment
- Enterprise concept team
  - Business concept of operations
- Enterprise architect
  - Enterprise architecture description
- Business design team
  - Enterprise specification
- Enterprise production implementation team
  - ….



Other architecting layers

- Components of detail need specifying and this too results in more rounds of architecting at the component level
- Life cycle, detail, and genericity dimensions all involve hand-off from one set of stakeholders to another as architecture evolves

---

- A framework informs about expected stakeholders as the elaboration space increases

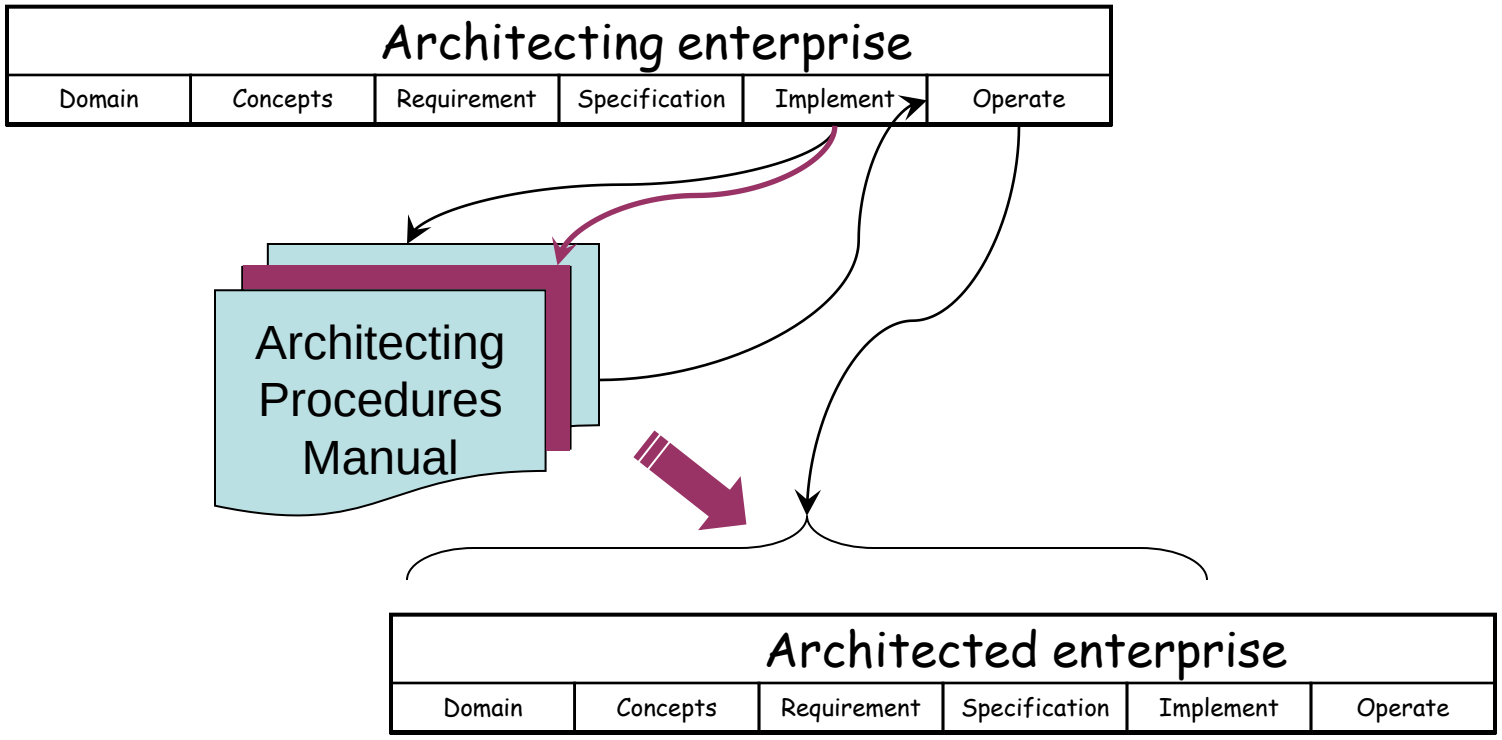
Elaboration Hierarchy

- Decomposing results in new architecture opportunity/specification raising issues of consistency and coherence between levels:
  - Enterprise, family of systems, system, segment, element, subsystem, component, subassembly, parts
- Transformations occur as context shifts focus in use of both meta-architecture for each level and creation of enterprise architecture

Life cycle evolution

- Within a life cycle phase, **AD** is an artifact of previous phases and serves as a guide for subsequent phases
- Systems and enterprises exhibit common life cycle patterns, not the same life cycle
- Instability is caused by overlap in life cycle phases across meta-levels
- Stability is enhanced by overlap in artifacts across meta-levels

Stability and instability



Meta- vs. time

- Meta-architecture specifies system life cycle processes that occur over time
- Tend to think of complete architectures as static but meta-level architecture use changes over the course of a project
- Different meta-levels have different time spectra; lower-level activity is continuous with respect to higher-level activity that is perceived as discrete, i.e. it has a more granular clock

Stakeholder utility

- Architecture accommodates succession of stakeholders
- Utility of **AD** is response to intentional concerns of input stakeholders
- Utility of **AD** is request for extending concerns of new stakeholders
- Utility of architecture is realized by service to stakeholders

Architectures to compare

- Zachman
- ISO 15704 Annex A GERAM
- EN/ISO 19439
- ISO/IEC 15228

Zachman Framework for Enterprise Architecture

ENTERPRISE ARCHITECTURE - A FRAMEWORK ™

	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL) *Planner*	List of Things Important to the Business ENTITY = Class of Business Thing	List of Processes the Business Performs Function = Class of Business Process	List of Locations in which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organizations	List of Events Significant to the Business Time = Major Business Event	List of Business Goals/Strat Ends/Means=Major Bus. Goal/ Critical Success Factor	SCOPE (CONTEXTUAL) *Planner*
ENTERPRISE MODEL (CONCEPTUAL) *Owner*	e.g. Semantic Model Ent = Business Entity Reln = Business Relationship	e.g. Business Process Model Proc. = Business Process I/O = Business Resources	e.g. Logistics Network Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organization Unit Work = Work Product	e.g. Master Schedule Time = Business Event Cycle = Business Cycle	e.g. Business Plan End = Business Objective Means = Business Strategy	ENTERPRISE MODEL (CONCEPTUAL) *Owner*
SYSTEM MODEL (LOGICAL) *Designer*	e.g. Logical Data Model Ent = Data Entity Reln = Data Relationship	e.g. "Application Architecture" Proc .= Application Function I/O = User Views	e.g. "Distributed System Architecture" Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	e.g. Human Interface Architecture People = Role Work = Deliverable	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means =Action Assertion	SYSTEM MODEL (LOGICAL) *Designer*
TECHNOLOGY MODEL (PHYSICAL) *Builder*	e.g. Physical Data Model Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	e.g. "System Design" Proc.= Computer Function I/O = Screen/Device Formats	e.g. "System Architecture" Node = Hardware/System Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Format	e.g. Control Structure Time = Execute Cycle = Component Cycle	e.g. Rule Design End = Condition Means = Action	TECHNOLOGY CONSTRAINED MODEL (PHYSICAL) *Builder*
DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*	e.g. Data Definition Ent = Field Reln = Address	e.g. "Program" Proc.= Language Stmt I/O = Control Block	e.g. "Network Architecture" Node = Addresses Link = Protocols	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification End = Sub-condition Means = Step	DETAILED REPRESEN-TATIONS (OUT-OF-CONTEXT) *Sub-Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

Zachman Institute for Framework Advancement - (810) 231-0531 Copyright - John A. Zachman, Zachman International

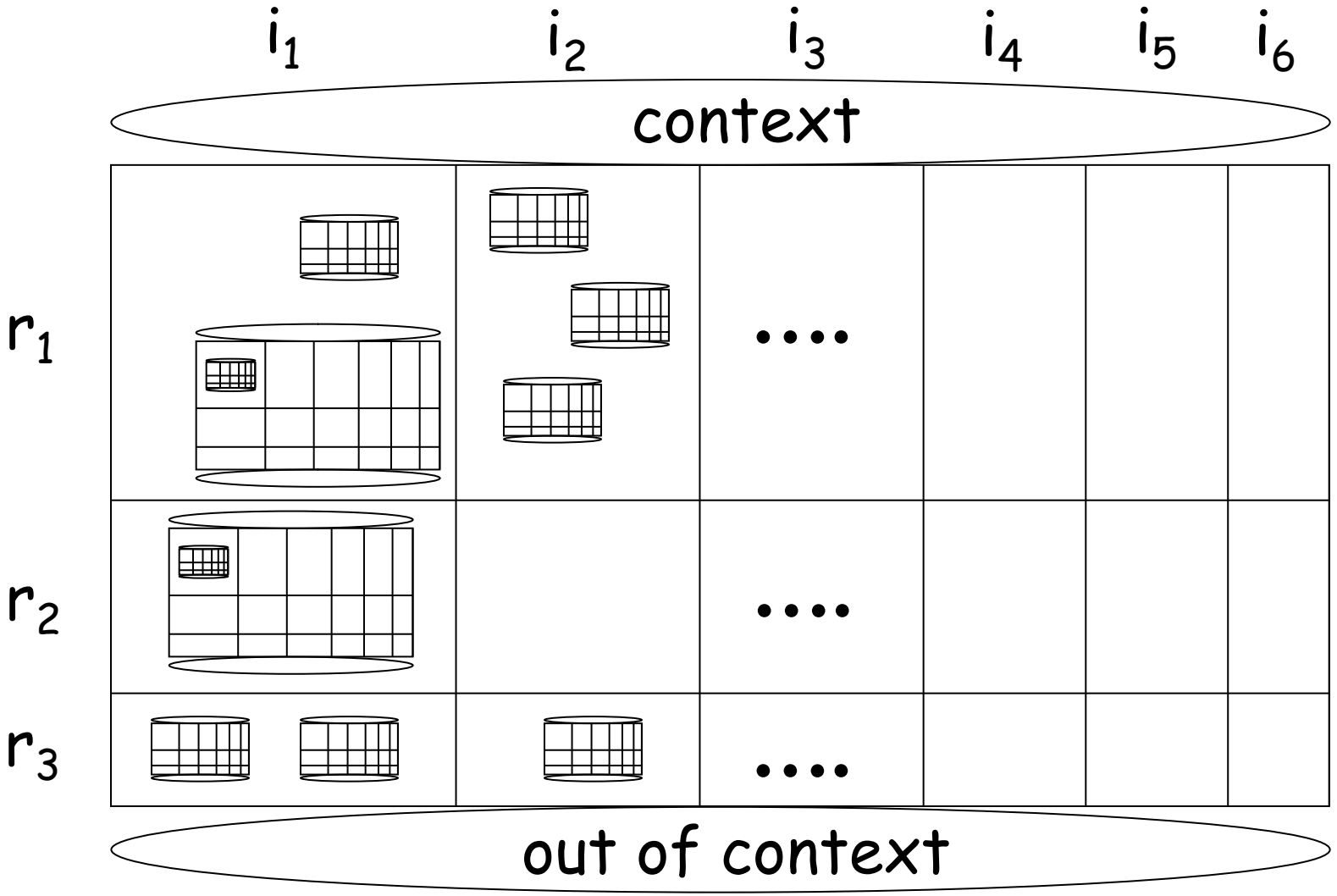
(used with permission)

Zachman Framework for Enterprise Architecture

(Information System version)

R \ I	What	How	Where	Who	When	Why
	Entity - Relation	I/O - Process	Node - Link	People - Work	Time- Cycle	Ends - Means
Context	Important things	Proceses performed	Operating locations	People and groups	Events and cycles	Goals and strategies
Owner	Semantic model	B-process model	Logistics network	Work flow model	Master schedule	Business plan
Designer	Logical data model	Application model	Distributed system	Human interface	Processing structure	Business rule model
Builder	Physical data model	System design	System arch.	Presenta- tion arch.	Control structure	Rule design
Out of context	Data definition	Program code	Network arch.	Security arch	Timing definition	Rule speci- fication

Zachman Recursion

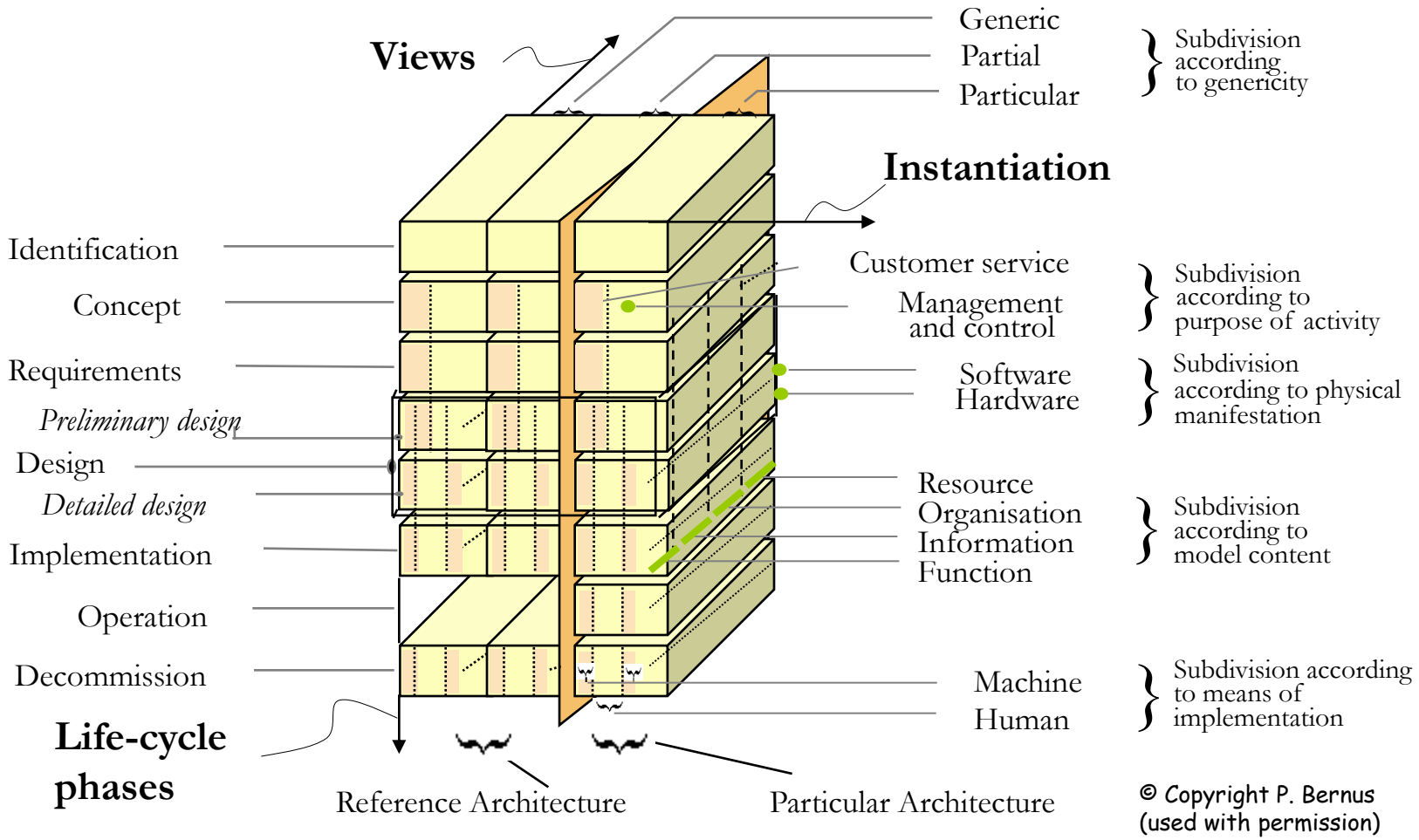


Zachman Properties

- **Role** dimension is ordinant, ordered, and purposive
- **Purposive** dimension is timeless
- **Interrogative** dimension is ordinant and unordered
- **Primitive model** contents facilitate complex model composition
- **Recursive** decomposition (frameworks nested in frameworks)

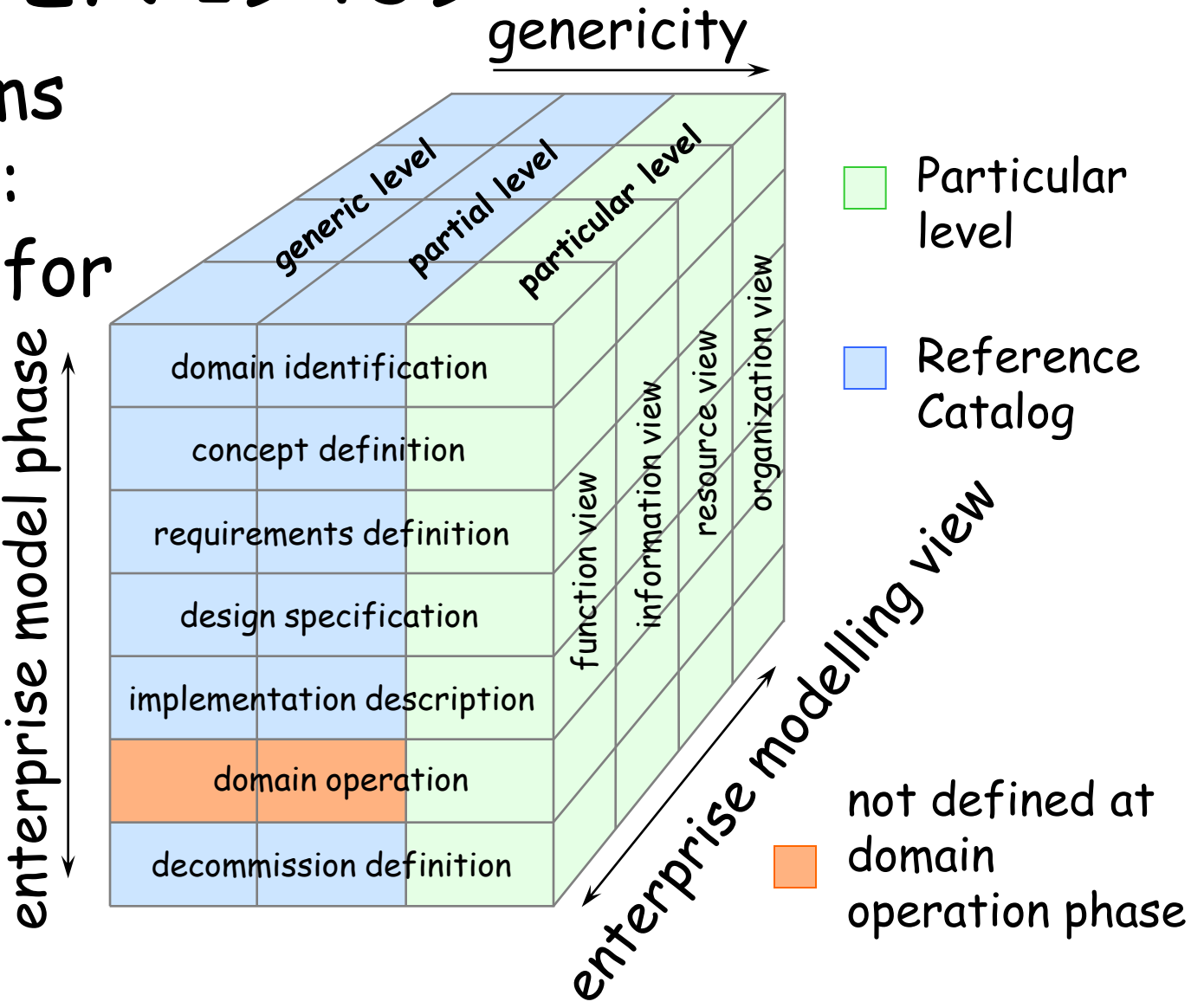
15704: Annex A - GERAM

Generalised Enterprise Reference Architecture and Methodology



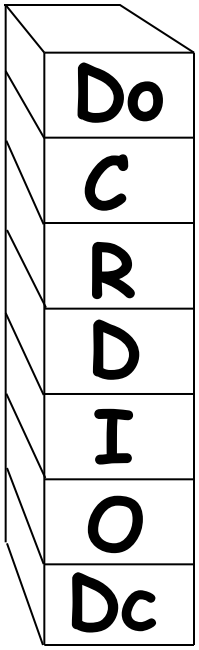
ISO/CEN 19439

CIM Systems Integration: Framework for Enterprise Modelling



19439 – Model Dimension

Model – the purposive ordinant dimension ordered by coordinates corresponding to the phases of the enterprise model life-cycle.



Enterprise **model phase**:

- **Domain** identification — *Identify*
- **Concept** definition
- **Requirements** definition
- **Design** specification
- **Implementation** description

Elaborate

- domain **Operation** — *Use*
- **Decommission** definition — *Dispose*

Emphasize model development process for process oriented modeling.

19439 - View Dimension

View – an unordered ordinant dimension with pre-defined or user selected coordinates that partition facts in the integrated model relevant to particular interests and context.

Enterprise **modelling view**:

Function - the system behavior, mutual dependencies, and influence of elements during function execution

Information - the material and information used and produced in the course of operations

Resource - capabilities of people and technological components

Organization- authority and decision-making responsibility during operations

ISO 19439 – Genericity

Genericity – an ordered ordinant dimension that reflects 19439 as a "standard" framework.

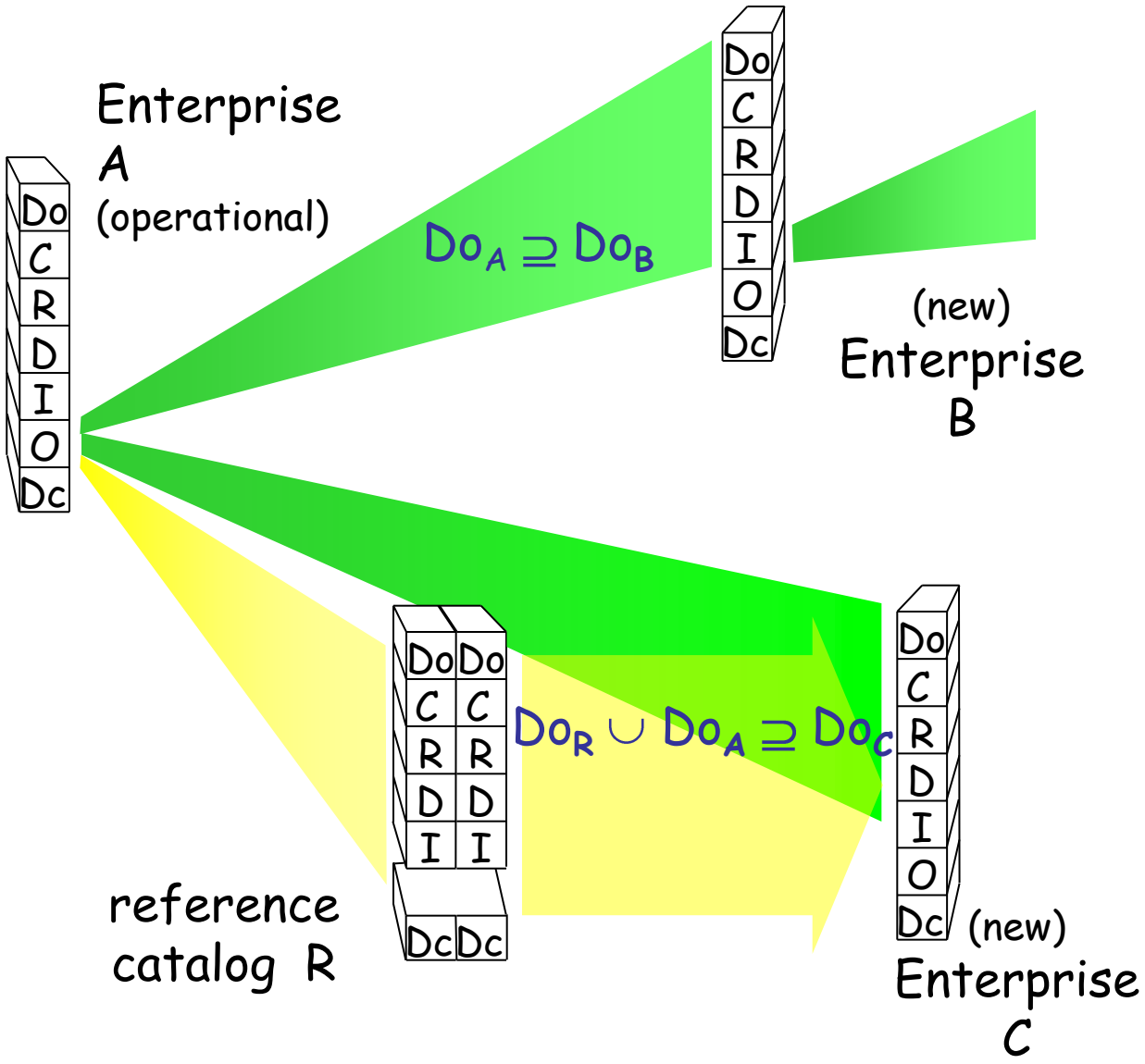
Enterprise **genericity level**:

- **Generic** - reusable modeling language constructs
- **Partial** - prototype models of industry segment or industrial activity
- **Particular** - models of a particular enterprise domain

Reference catalog

19439 - Recursion

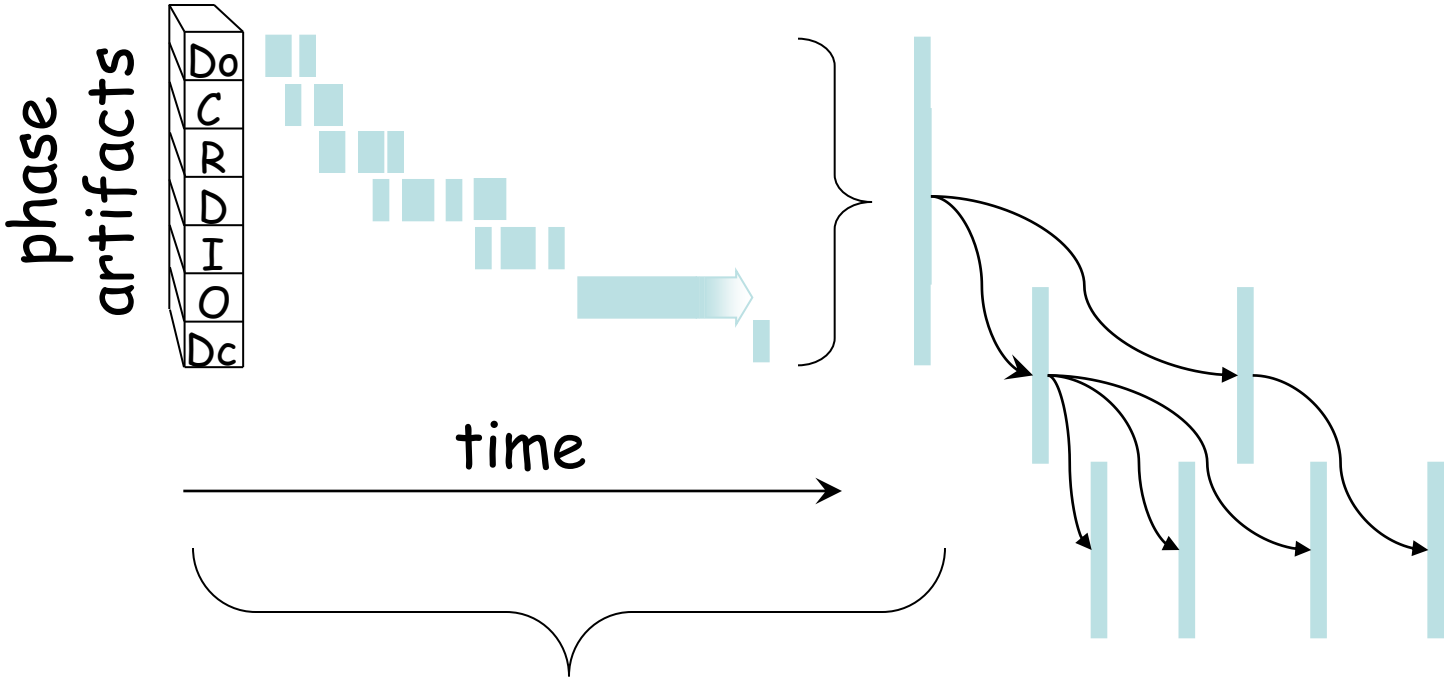
Enterprise operations can model new enterprises either from its own particular models or using reference constructs and partial models.



Phase vs. Stage

- Phase refers to the various modeling aspects of the life cycle
- Phase is link to dependency that can be identified with respect to a particular phase
- Stage refers to the various intervals of an enterprise life history
- Stages are firmly linked to sequential time

19439 – Life History



a complete life-cycle

a life history pictogram of related life-cycles (point-in-time solution set)

Adapted from P. Bernus, Griffith University, Australia

ISO/IEC 15288 Systems engineering – System life cycle processes

- Common process framework covering life cycle of man-made systems…spans conception of ideas through to retirement
- For acquiring and supplying systems
- Assess and improve life cycle processes
- Comprehensive set from which an organization can construct system life cycle models
- Can be applied at any level of system structure and throughout life cycle

15288 - Structure

A degenerative case where framework structure is trivial but has many constraints that govern instances, e.g.,

Modularity – maximal cohesiveness of the functions of a process and minimal coupling among processes.

Ownership – a process is associated with a responsibility.

Properties – the purposes, outcomes and activities for a process

15288 – Dimensions

Process Group – a hierarchic arrangement where enterprise processes manage project processes composed of technology processes all mediated by agreement processes

Life cycle – minimal normative requirement

"A life cycle model that is comprised of stages shall be established…The purpose and outcomes shall be defined for each stage of the life cycle."

15288 - Process Groups

- Agreement – define activities that establish agreement between internal/external entities
- Enterprise – manage capability to acquire and supply through project initiation, support and control
- Project – establish and evoke project plans, assess achievement, control execution
- Technical – define the activities that enable functions to optimize benefits and reduce risks of technical decisions and actions

15288 - Process Hierarchy

Project Portfolio Mgmt Quality Mgmt Life Cycle Model Mgmt

Infrastructure Mgmt Human Resource Mgmt < 11p, 24o, 47a >

Project Planning Project Assessment & Control Risk Mgmt Config Mgmt

Information Mgmt Measurement Decision-making < 15p, 42o, 93a >

Stakeholder Requirements Definition

Requirements Analysis

Architectural Design

Implementation

Integration

Verification

Transition

Validation

Operation

Maintenance

Disposal

< 11p, 52o, 98a >

15288:2002 - Process Hierarchy

Enterprise Environment Mgmt

System Life Cycle Processes Mgmt

Investment Mgmt

Resource Mgmt

Quality Mgmt

< 11p, 21o, 34a >

Project Planning

Project Assessment

Project Control

Decision-making

Risk Mgmt

Configuration Mgmt

Information Mgmt

< 16p, 35o, 61a >

Stakeholder Requirements Definition

Validation

Operation

Transition

Maintenance

Requirements Analysis

Verification

Architectural Design

Integration

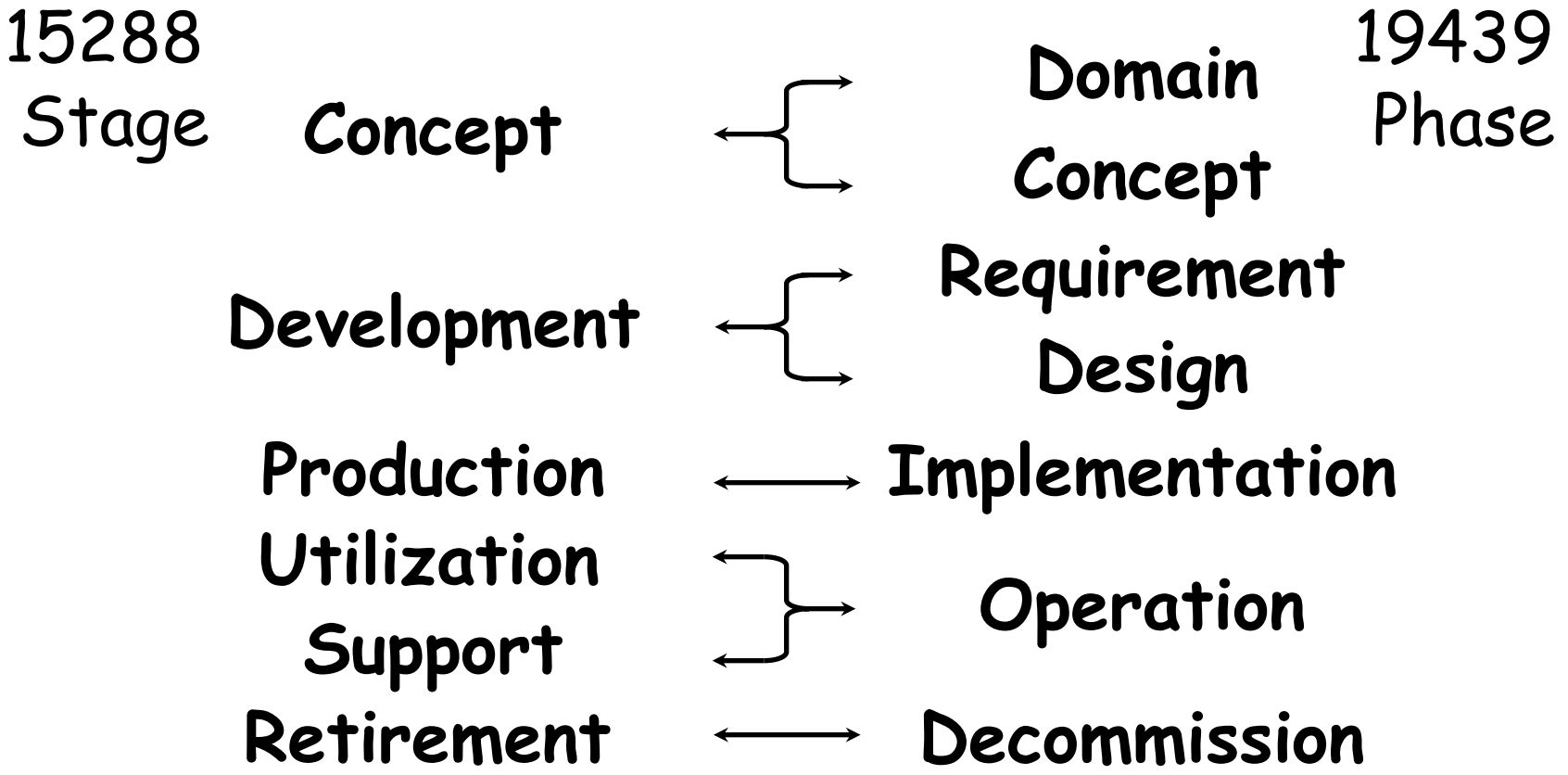
Disposal

Implementation

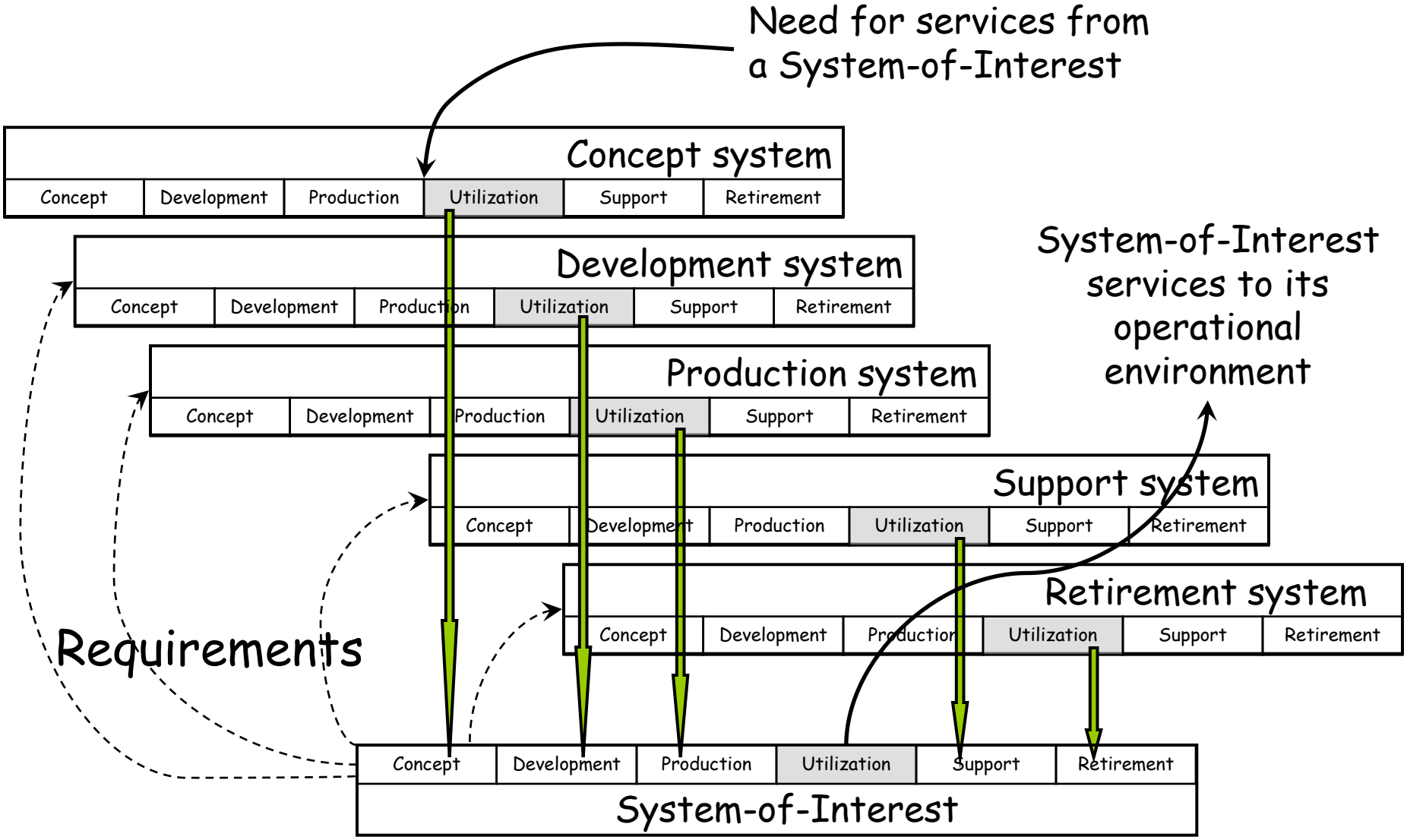
< 34p, 53o, 96a >

15288 - Life Cycle

Informative guidance for life cycle stages



15288 - Recursion



Archetype Summary

Zachman –
Role {Context, Owner, Designer, Builder, Out-of-context}
Interrogative {What, How, Where, Who, When, Why}

ISO\CEN FDIS 19439 –
Model {Domain, Concepts, Requirements, Design,
Implementation, Operation, Decommission}
View {Function, Information, Resource, Organization}
Genericity {Generic, Partial, Particular}

ISO 15288 –
Process Group {Agreement, Enterprise, Project, Technical}

Prototype Models

Zachman - **interrogative models** {entity-relationship, input-process-output, node-link, people–work, time–cycle, ends–means}

Zachman - **cell models** {Semantic Model, System Design, Control Structure, Business Plan, etc.}

19439 - **constructs** {domain, business process, enterprise activity, event, enterprise object, resource, capability, decision centre, etc.}

19439 - **partial models** {industry sector, company size, national variation, etc.}

15288 – **process definitions** { 25 processes consisting of 63 purposes, 123 outcomes, and 208 activities (in 33 pages of text)

Purposive Dimension

Zachman has a continuant purposive dimension (Role) and therefore serves well in an **analytic resource and reference mode**. It is always all there – either explicitly or implicitly.

19439 has an occurrent purposive dimension (Model Phase) and therefore serves well in a **realization and operational mode**. It provides the point-in-time solutions we use.

15288 has a decompositional purposive dimension (Process Group) with descriptive process artifacts **suitable for use in Zachman or 19439**.

Comparative Summary

Zachman is the most comprehnsive of the three presented.

Zachman holds primitive models while 19439 extracts those primitives and composes views.

Zachman provides a conceptual partitioning as a major focus whereas the other two focus on support for methodological approaches.

Lunch Break

- Is comparing frameworks by looking at viewpoints useful?
- Do the International Standards seem to hit a target at which you have been aiming?

More depth into the standards next!

ISO 15704 – Framework for enterprise architecture and models

ISO 14258:1998

Adopted after 14 year effort

- Over 300 documents reviewed
- Focus on manufacturing systems architecture
- Lead by National Institute of Standards and Technology - USA

A consensus predicate to on-going reference model effort

10 basic definitions (some still contentious)

Systems theory aspects

structural – elements have multiple interdependencies leading to emergent qualities

behavioral – identification of variable and functional relationships

hierarchical – systems within systems and levels of abstraction embodied in emergent qualities

Levels of abstraction

Lower levels reveal detail and the means to achieve purpose – more concrete

Higher levels reveal the role of system within environment – more abstract

Each level has structure and behavior

Observation - enterprise architecture (EA) is manifest in enterprise models (EM) as a pattern. Constraints on the EM are the EA.

Need for life-cycle

EM shall address what happens to the factors of production (such as people, capital, material, information, energy, and tools) during the phases of the enterprise or product life-cycle.

- Products, processes, projects, and enterprises are systems.
- Systems have a life cycle that can be partitioned into phases such as plan/build, use/operate, and recycle/ dispose.

Broad model scope

EM shall define relevant aspects of the enterprise necessary to

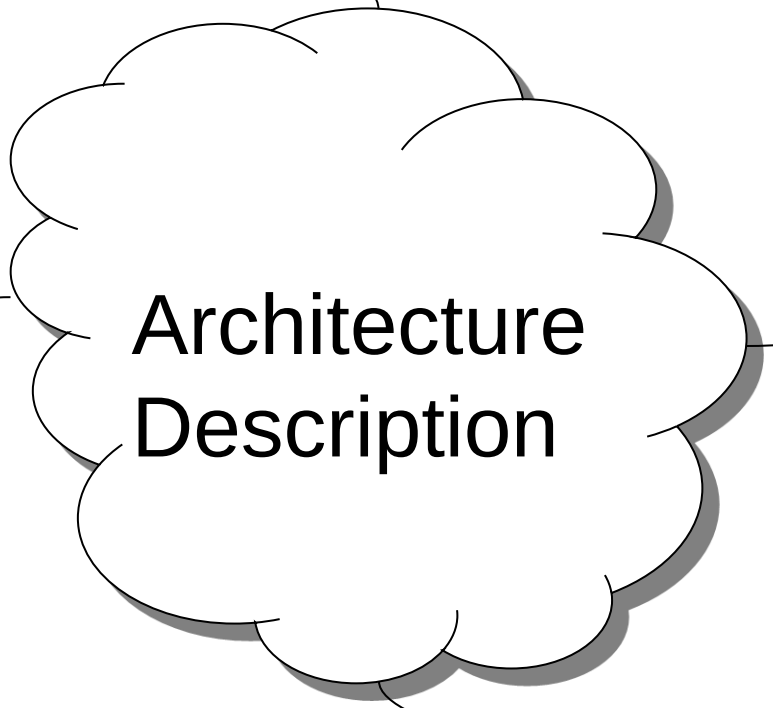
- conceive, design, procure for, and construct an enterprise consisting of any set of related chosen processes
- manage and operate an enterprise so that it can meet its objectives
- support an enterprise to modify, redesign, dismantle and rebuild it

Must be accessible

As architectural representations of enterprises, models shall exhibit syntax and semantics so that contents of the model are understandable to human users.

- The syntax of a model refers to the permissible kinds of relations.
- The semantics of a model encompass the meanings of the elements and relations with respect to enterprise-model concepts.

AD as boundary object



Literature:
Documentation for current and future generations of users and developers

Blueprint:
Specification of the system to be implemented

Language:
Medium of communication for achieving common understanding

Decision:
Choices about the system to be implemented and rationale

A bit too 'formal'

```
(forall (?occ)
(iff (occurrence_of ?occ make_harness_wire)
     (exists (?occ1 ?occ2 ?occ3)
        (and (occurrence_of ?occ1 extrude)
             (occurrence_of ?occ2 twist)
             (occurrence_of ?occ3 jacket)
             (min_precedes ?occ1 ?occ2
                  make_harness_wire)
             (min_precedes ?occ2 ?occ3
                  make_harness_wire))))
```

[Source: ISO/CD 18629-44 Annex B]

ISO 14258:1998 Figure 1

Phase \ Issue-solving activities	"What" Activities	"How" Activities	"Do" Activities
Plan and Build Phase (e.g., before sell/buy title transfer)	•Develop goals •Define strategy •Define product needs	•Develop Requirements •Define concept •Design product •Plan to produce product •Plan to support product	•Procure parts •Produce product •Test product •Ship product
Use and Operate Phase (e.g., after sell/buy title transfer)	•Define support needs •Define Use	•Define Use Requirements •Define Support Requirements	•Use the product •Support product
Dispose and Recycle Phase (e.g., after product is No longer useful)	•Define recycle/dispose needs	•Define recycle/dispose requirements	•Recycle product •Dispose product

Another way to view it

ISO 14258:1998 Figure 1 Transposed

Issue-solving activities \ Phase		Plan and Build Phase (e.g., before sell/buy title transfer)	Use and Operate Phase (e.g., after sell/but title transfer)	Dispose and Recycle Phase (e.g., after product is no longer useful)
		w| How |w|w|w|w	w| How |w|w|w|w	w| How |w|w|w|w
Specify	"What" Activities	•Develop goals •Define strategy •Define product needs	•Define support needs •Define Use	•Define recycle/dispose needs
Design	"How" Activities	•Develop Requirements •Define concept •Design product •Plan to produce product •Plan to support product	•Define Use Requirements •Define Support Requirements	•Define recycle/dispose requirements
Build Operate	"Do" Activities	•Procure parts •Produce product •Test product •Ship product	•Use the product •Support product	•Recycle product •Dispose product

Life-cycles of systems

Different life-cycle phases may have different models.

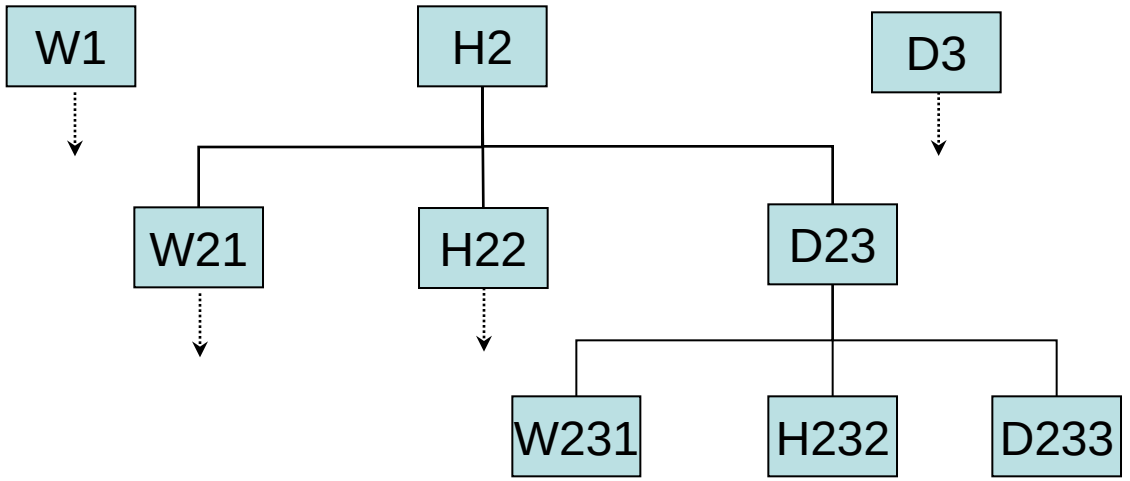
These models shall have the capability to interoperate where it has been determined that processes need to communicate with each other.

- Feeding modeled information forward and backward in life-cycle activities enables value-added iteration of enterprise processes that improves product quality.

Recursion (structural)

The What, How, and Do activities are recursive and decomposable.

Activities can be divided into sub-activities, and these sub-activities will consist of another set of W, H, and D activities.



Sub-activities interact

Sub-activities may be represented by different types of models.

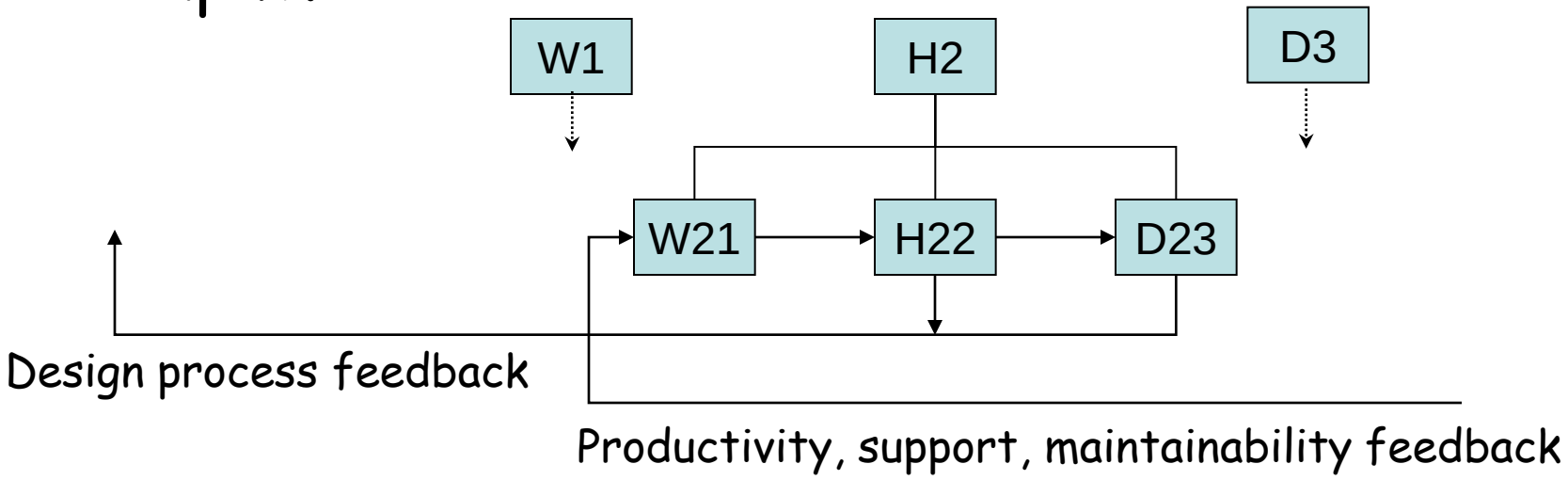
These models shall be able to interoperate where it has been determined that these sub-activities need to communicate with each other.

Observation – recall the distinction between structure and connection where both are relationships

Iteration (behavioral)

The W, H, and D activities are iterative.

- There is no fixed sequence of activities.
- It is possible to return to previous activities to repeat them with updated input.



Manage change

Each performance of each model-making activity may result in a different model.

Every one of these different models shall be subject to both change and version management.

Enterprise models shall be designed in such a way as to allow their constituent parts to be managed by an automated configuration-management system.

Hierarchy types

Classification (Kind-of) hierarchies shall be used within models to classify building blocks for entities to be modeled.

Composition (Part-of) hierarchies shall be used to link models of different scope and detailing granularity of decomposition.

Observation – not all structures are so regularly composed

Structure concept types

1. Activities correspond to elements and objects correspond to relations.
   - E.g., a value-adding process where the output objects (considered as relations) of one activity (considered as an element) are the input objects of another action (considered as an element).
2. Activities correspond to relations and objects correspond to elements.
   - E.g., the structure of a process plan where two objects (considered as elements) are linked by an activity (considered as a relation).

Compatible structures

The type of structuring shall be unambiguous to whatever facility is interpreting the models, either human or machine.

The enterprise modeler shall ensure that models obtained by the two structuring approaches are able to interoperate.

Observation – two views of the same underlying conceptual or actual model

Behavioral concepts

An enterprise is a social hybrid system, determined by properties of humans and machines.

Humans (modeled as objects or resources) in the enterprise have a different behavior (e.g. learning and problem solving) from machines (e.g. acting and reacting) and sometimes need a different kind of information.

Representing behavior

The set of models for an enterprise shall have the capability to describe behavior with respect to

- sequentiality, events, actions, condition
- states, state changes, start states, end states,
- sequencing relationship between actions,
- description of transformation functions.

Time concepts

Time is relative to the observer.

- Static representations are devoid of time
- Dynamic representations express time sensitive properties and dependencies
- Change is immediate (short-term) or continual (long-term)

Describing behavior requires sequentiality.

- Sequential cycles are similar states being traversed at different times.
- Measuring sequence in time enables discrimination between similar cycles progressing at different rates.

Representing time

To trace individual elements, sequence properties shall be modeled to describe short-term changes.

EM shall be able to represent time duration, dynamic performance of processes, and sequential phenomena after specific units of time.

EM used to analyze enterprise performance or to simulate processes shall be able to represent effects of sequential phenomena and the time duration of each sequence step.

Modeling purpose

Models describe essential and relevant parts of an area of concern.

Models do not duplicate reality but are limited approximations of the subset of reality under consideration.

Extent of model detail is relative to its purpose.

Full model description includes purpose, assumptions, and constraints.

Observers ≈ Viewpoints

- Observers perceive and analyze with attribution of meaning.
- Observer filter is continually modified by experience, personality, politics, society, and situation.
- Enterprise modeler is observer whose purpose is to create an enterprise model.
- The modeler shall define unambiguously the purpose for the model.
  - Model user is observer with task in area of concern addressed by the model.

Views

Views enable observation using assumptions and constraints. (viewpoints)

Views are used to verify completeness, consistency, and integratability of EM.

Two views are of primary importance in representing the structure and behavior of a real world system.

- The information view reveals structure.
- The function view reveals behavior

Model description

A full, integratable description of any model shall include statements and descriptions of its purpose, assumptions, and constraints.

This shall be done by including a minimum set of modeler views that ensure adequate completeness and consistency, and provide the potential for integrating multiple models of the same enterprise.

Many models, one solution

- There are as many ways to represent in models as there are reasons to model.
- Users want to reuse models across applications and not be dependent on specific application and tool configuration.
- Users want the many forms to appear as and operate as one solution for their enterprise modeling needs.
- Many models must interoperate to be one solution.

ISO 15704:2000

- Adopted in 2000 after 15 year effort to consolidate existing knowledge and practice in the area of industrial automation
- Lightweight standard, only 8 pages, and middleweight annex A of 31 pages with heavyweight content
- References normative content from ISO 14258

ISO 15704:2000 (cont.)

Primary input is annex A produced by an IFIP-IFAC task force

Annex A, known as GERAM, articulates a compliant approach and expands context

Amended in 2005 with 2 new user views

- Economic view that introduces a new dimension of detail
- Decision view that introduces a way to articulate operational decisions in time

Informative introduction

Two primary concerns for enterprise integration base of reference

- Model the whole life history of an enterprise-integration project
- Encompass the people, processes and equipment involved in achieving the enterprise mission

EA is about enterprise project structure.

SA is about system that is part of EA.

Eight key principles

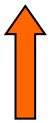
1. Issues of enterprise integration are ubiquitous and apply to any enterprise, regardless of its size and mission or other attributes.
   - Integration goes beyond information and control systems to encompass culture and mission.
   - Manufacturing is a customer service.
2. Enterprise identification and mission definition are essential.

Key principle 3

3. Separate mission-fulfillment functions from mission-control functions
   - Fulfillment includes process operation to produce the product or service
   - Control includes the use of information to manage processes and maintain operations
   - Interconnection between fulfillment and control is operational data to control and operational commands to fulfillment.

Key principles 4 & 5

4. Operations consists of the many transformations of material, energy, and information performed by processes.
   - Processes executed concurrently or sequentially
   - Combination of processes define functionality of the enterprise
5. Evolutionary integration of modules is essential.

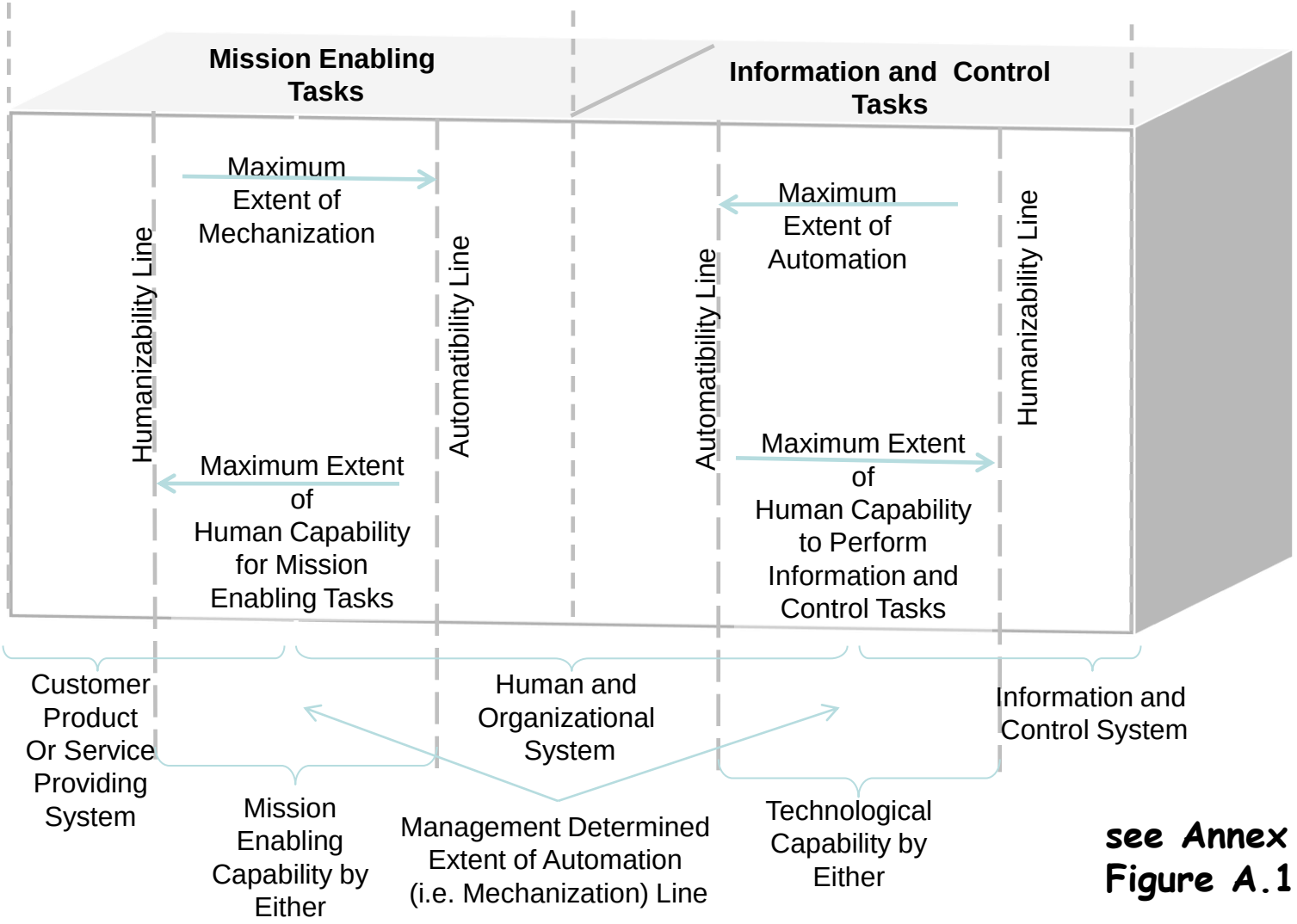


evolving

Key principle 6

6. Three kinds of processes
   - Information and control activities that can be automated by control devices
   - Mission activities that can be automated by mission-fulfillment equipment
   - Activities carried out by humans, whether for information and control or mission-fulfillment
     - Desire a simple way of showing where and how humans fit in the enterprise and how the distribution of functions between humans and machines is accomplished

Function distribution



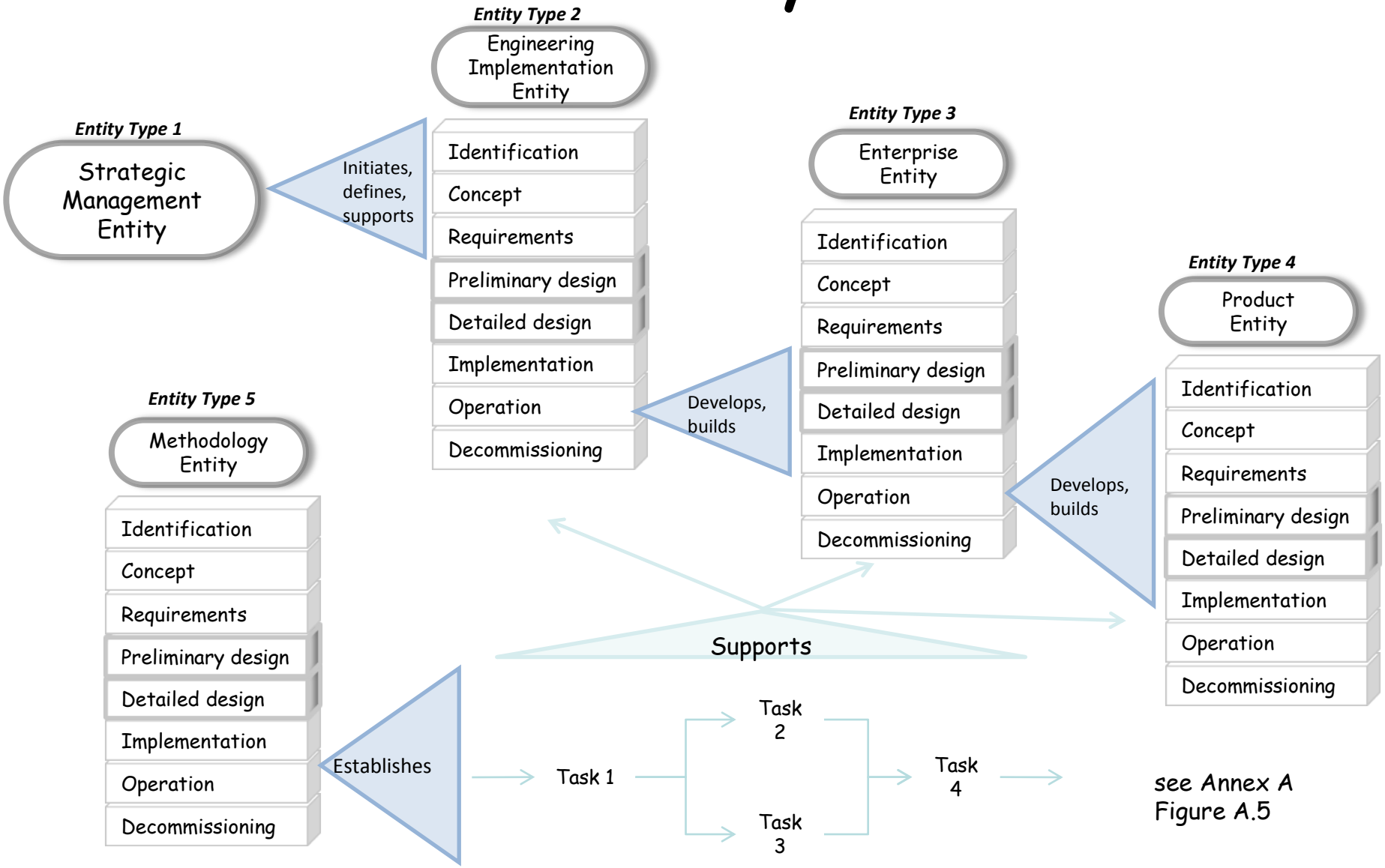
see Annex A, Figure A.11

Adapted from: T. J. Williams, Institute for Interdisiplinary Engineering Studies, Purdue University

Key principles 7 & 8

7. Every enterprise and product has a life-cycle.
   - A life-cycle can be partitioned
     - Partitioning is specific to purpose
     - Partitioning does not imply strict sequence
   - One enterprise can be embedded within the life-cycle of another's operational phase.
8. Modularity should be enforced whenever possible.

Embedded life-cycles



Stakeholder concerns

The standard enables an enterprise-integration-planning team to:

- Describe the tasks required
- Define the necessary quantity and quality of information
- Specify relationships among humans, processes, and equipment in the integration considered
- Address management needs

More concerns

- Address relevant economic, cultural, and technological factors
- Detail the extent of computer-support required
- Support process-oriented modeling that can model the whole life history of an enterprise
- Checking for completeness with respect to ERAM current and future purpose

ISO 15704 Scope

Enterprise-reference architectures and methodologies (ERAM) covers those constituents deemed necessary to carry out all projects required by the enterprise throughout the whole life of the enterprise, including:

- enterprise creation
- major enterprise restructuring efforts
- incremental changes affecting only parts of the enterprise-life cycle

Requirements of 15704

Enterprise-reference architectures and methodologies (ERAM) shall be capable of assisting and structuring the description, development, operation, and organization of any conceivable enterprise entity, system, organization, product, process, and their supporting technology.

Areas covered by ERAM shall be clearly identified.

General usage

The methodology associated with a reference architecture shall provide the necessary guidelines and management techniques for the initiation and pursuit of a project or program of development and operation of an enterprise or entity.

Such a methodology may or may not be model-based, i.e., the enterprise engineering process may or may not result in a specific enterprise model.

Many potential ways

ERAM need not be based on any one single methodology and its accompanying architecture or framework. Many different methodologies and/or frameworks may be used for it.

The primary consideration shall be applicability and capability in relation to these requirements.

Design and operation

ERAM shall identify

- concepts and components
- activities to manage, conceive/define, describe, design, implement, maintain, and decommission any enterprise entity
- activities to use the results of enterprise engineering in the operation itself
  - Such use may include model-based decision support and model-driven operation monitoring and control.

Conceptually broad

Throughout the life-cycle of the enterprise, the ERAM shall address the

- role of humans
- description of processes (function and behavior)
- representation of supporting technologies

Human oriented

ERAM shall exhibit the capability to represent human aspects, such as:

- Organizational and operational roles
- Capabilities, skills, know-how, competencies
- Responsibilities, authorization
- Relations to the organization.

Process oriented

ERAM shall exhibit the capability to represent the enterprise operation.

Such representations shall cover both the functionality and behavior of the operation.

The representations shall recognize the life cycle and life-history concepts of enterprise-entity types and shall support process-oriented operations.

Technology oriented

ERAM shall exhibit the capability to represent all technologies employed in the enterprise operation.

Such representation shall provide for the use of integration-technology infrastructures to support

- enterprise engineering and operation of business processes, models of enterprise resources, facility layout models, information-system models, communication-system models and logistics models.

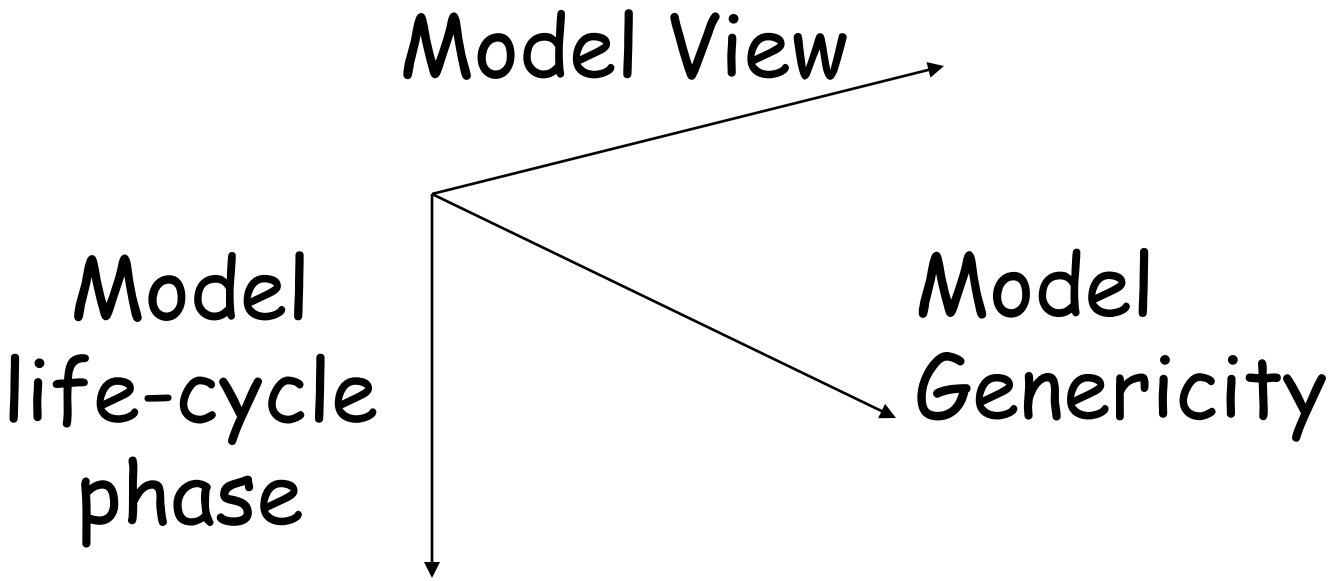
Mission oriented

ERAM shall exhibit the capability to represent any process and its constituent activities involved in

- performing the established mission of the enterprise
- accomplishing management and control of the established mission of the enterprise according to the criteria established by enterprise management

Framework for modeling

ERAM that are model-based shall exhibit the capability to model entities within the conceptual space defined by three dimensions.



Genericity

ERAM that are model-based shall provide the capability for representing

- Generic-enterprise elements that are reusable modeling language constructs
- Partial-enterprise models that are prototype models of industry segment or industrial activity
- Particular enterprise models that are models of a particular enterprise domain

Life-cycle & life-history

ERAM shall identify and represent the life-cycle phases pertinent during the life of any enterprise entity.

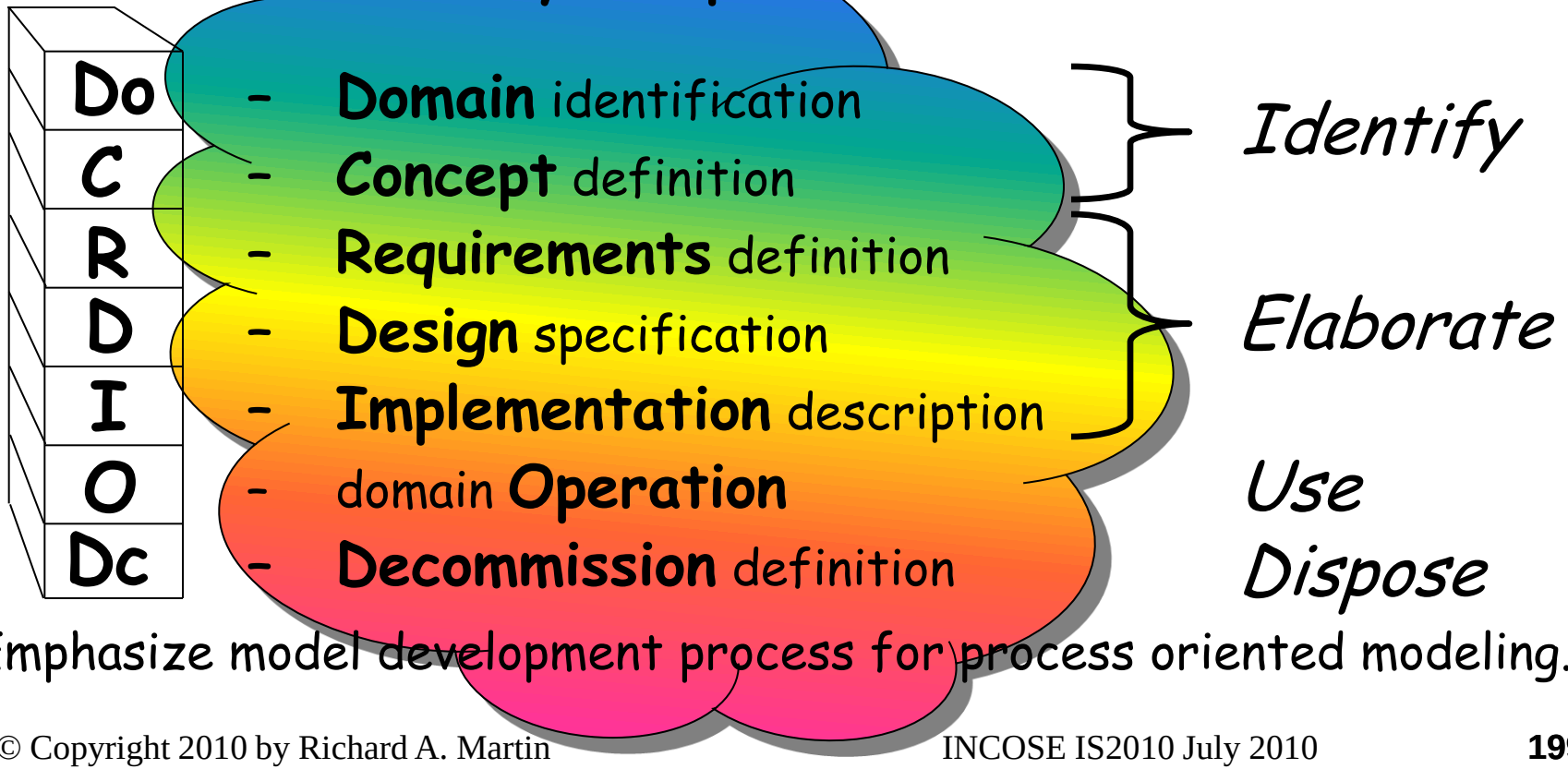
(Dependency links phases)

ERAM shall be capable of representing the life history of any enterprise entity, i.e., the representation in time of activities carried out on any enterprise entity (traceability)

(Chronology links history)

ISO 19439 Model phase

The purposive ordinant dimension ordered by coordinates corresponding to the life-cycle phases



Emphasize model development process for process oriented modeling.

Early phases

Domain identification

- Business objectives, functions, capabilities

Concept definition

- Enablers of objectives & operations
- Means for achievement of functions & capabilities

Requirements definition

- Functional, behavioral, informational, capability for service, manufacturing, management and control

Design specification

- Processes with all components necessary to satisfy requirements

Post-design phases

Implementation description

- All information needed for all tasks of operational system

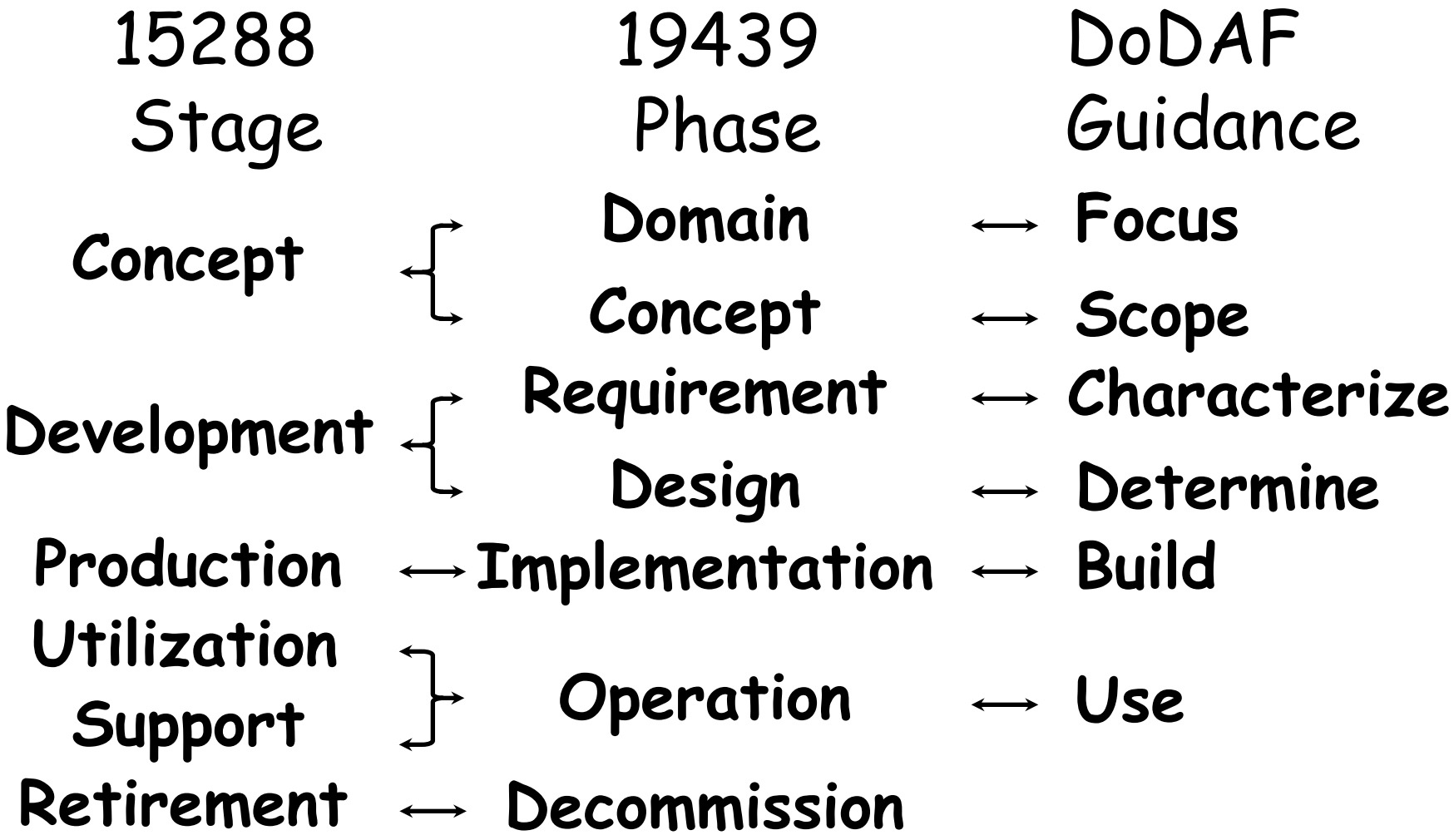
Domain operation

- Operational usage of model released from implementation

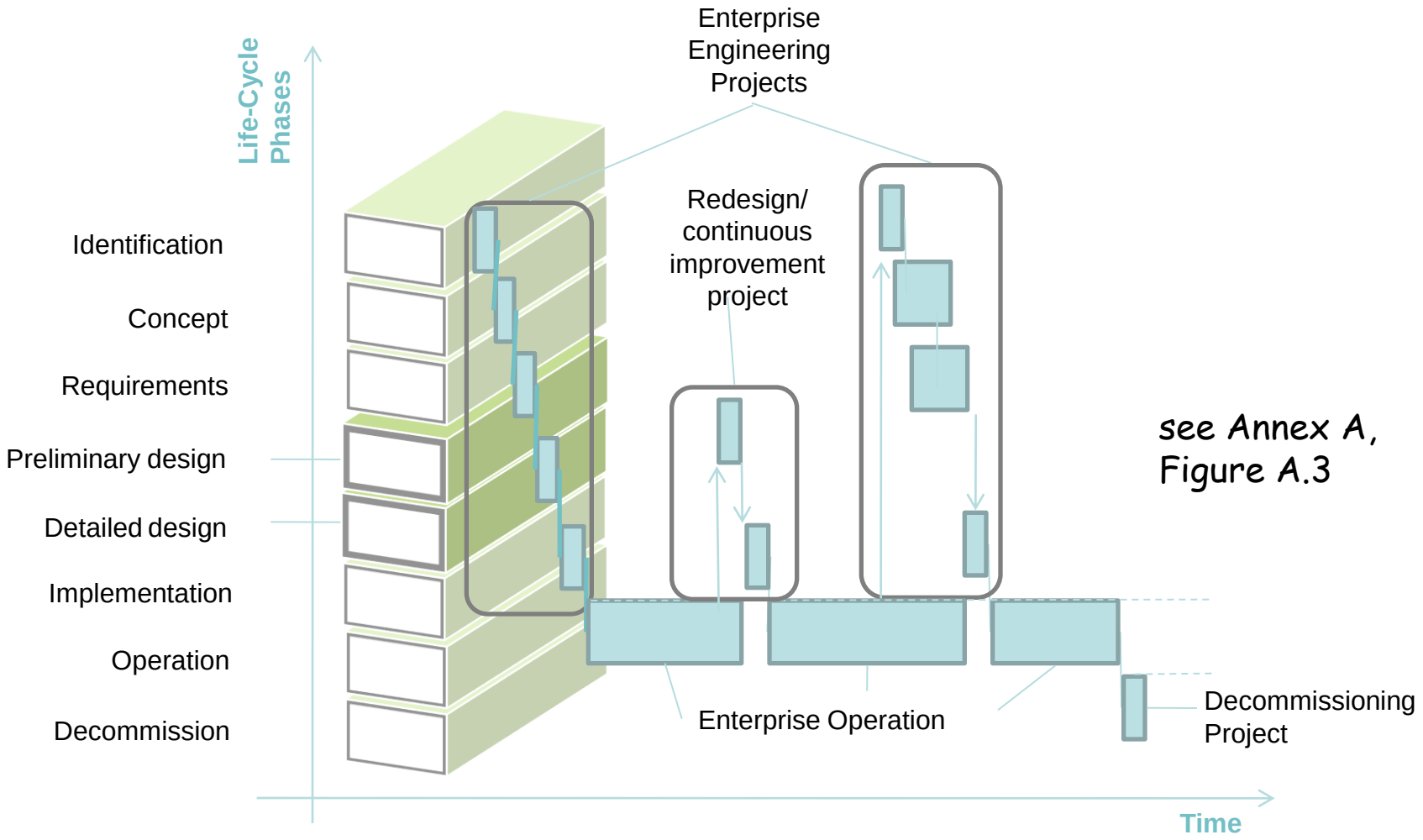
Decommission definition

- Tasks and resources for retraining, redesign, recycling, preservation, transfer, disbanding, disassembly, disposal

Many possible coordinates



Life history



see Annex A, Figure A.3

Model "view"

ERAM that are model-based shall provide concepts for representing views of a model to allow it to be described as an integrated model but presented to users in different subsets.

- Views contain subsets of facts present in the integrated model
- Concentrate on relevant questions respective stakeholders may wish to consider.
- The concept of view is applicable to models of all entity types across their life cycle.

Four views required

Different views may be made available highlighting certain aspects of the model and hiding others.

ERAM that are model based shall include four model-content views:

Function

Information

Resource

Organization

Function view of 19439

Function

- Enables representation and modification of the processes of the enterprise, their functionalities, behaviors, inputs and outputs
- Emphasis on system behavior, mutual dependencies, and influence of elements during function execution
- Includes decisional, transformational and support activities
- Identifies all entities (material, information, resources and control) required for function execution

Other views of 19439

Information

- The material and information related objects used and produced in the course of operations

Resource

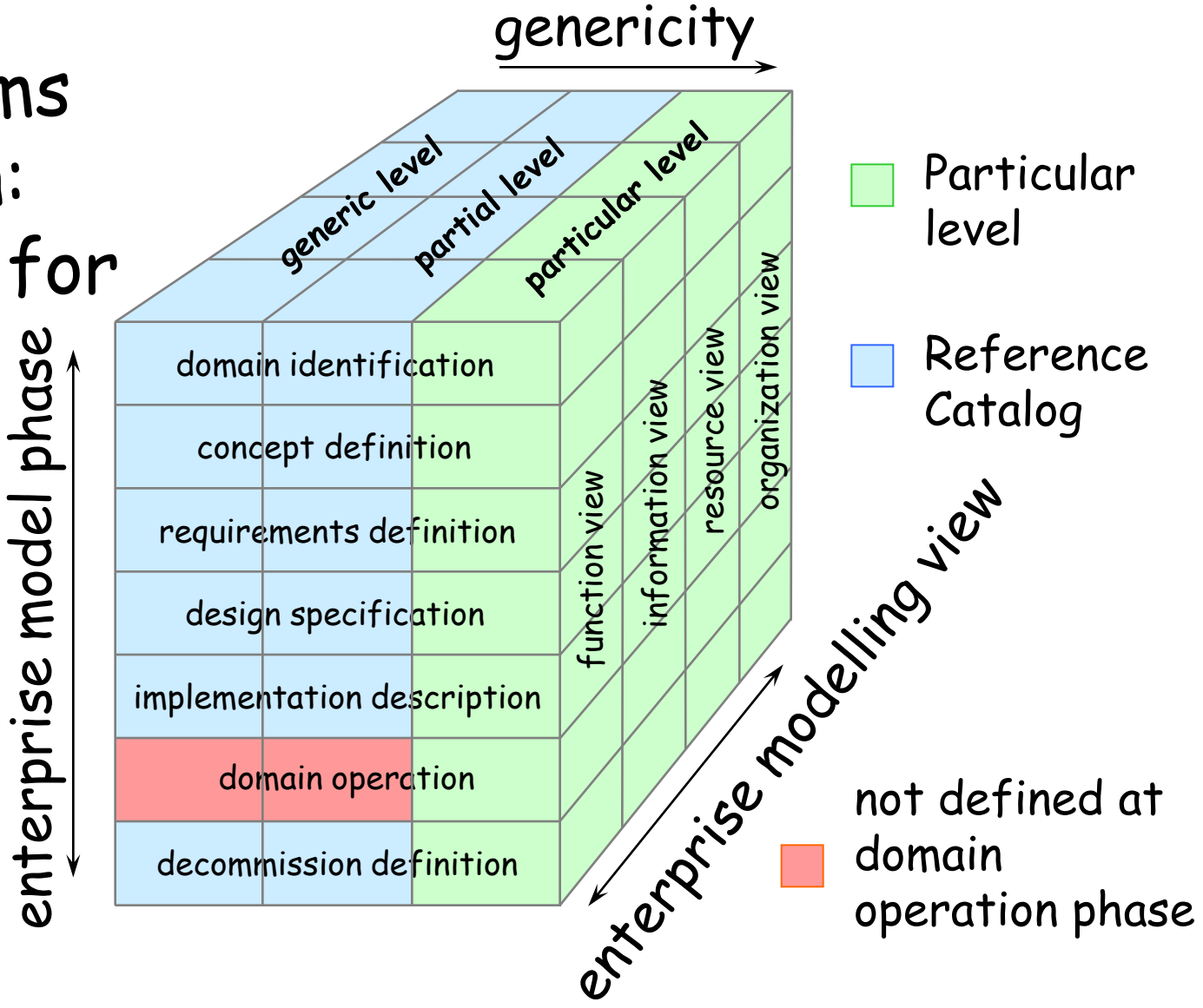
- Capabilities of people and technological component assets

Organization

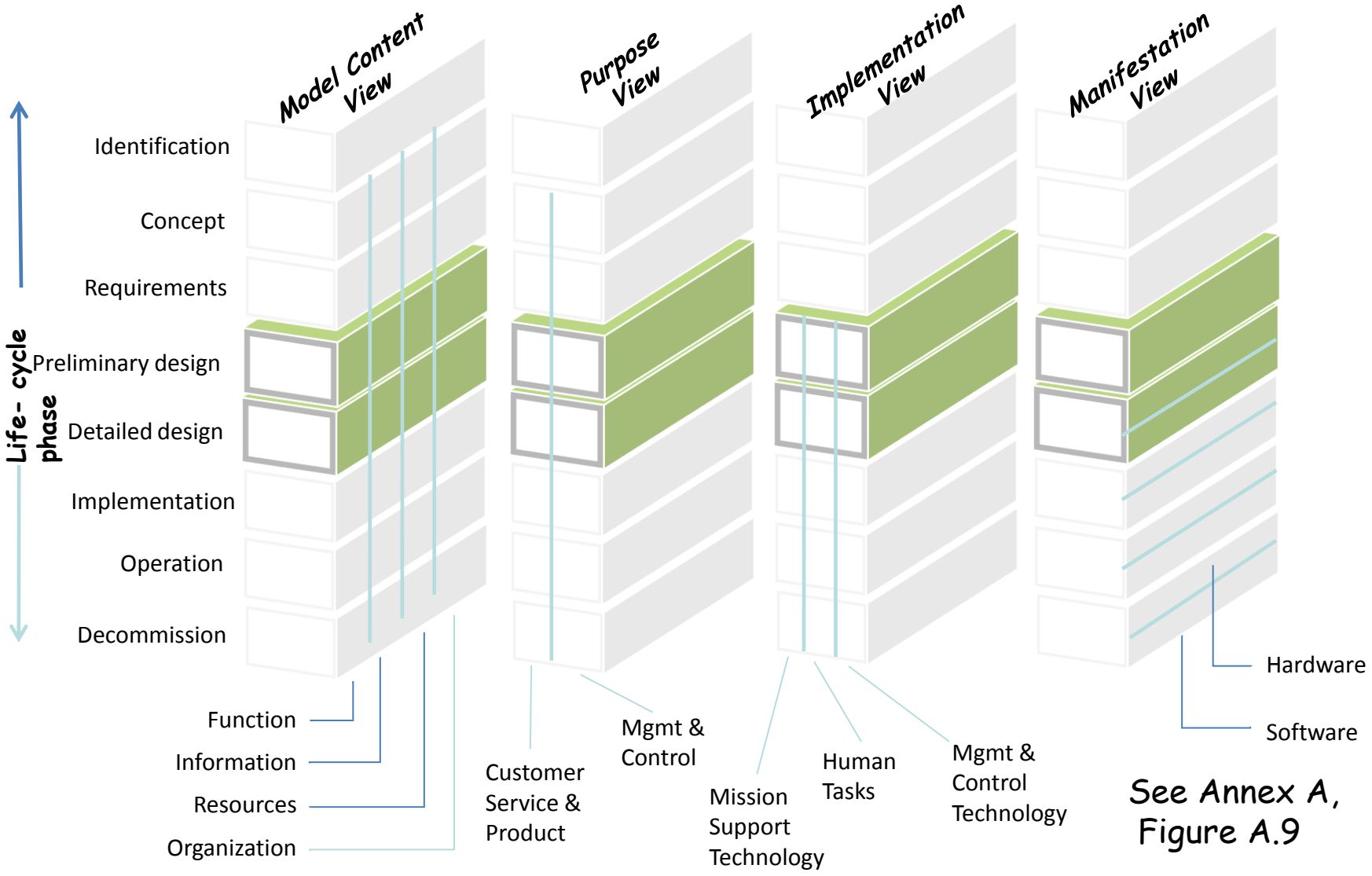
- Authority and responsibility during operations
- Expresses decision support structure

Graphic 19439 dimensions

CIM Systems Integration: Framework for Enterprise Modelling



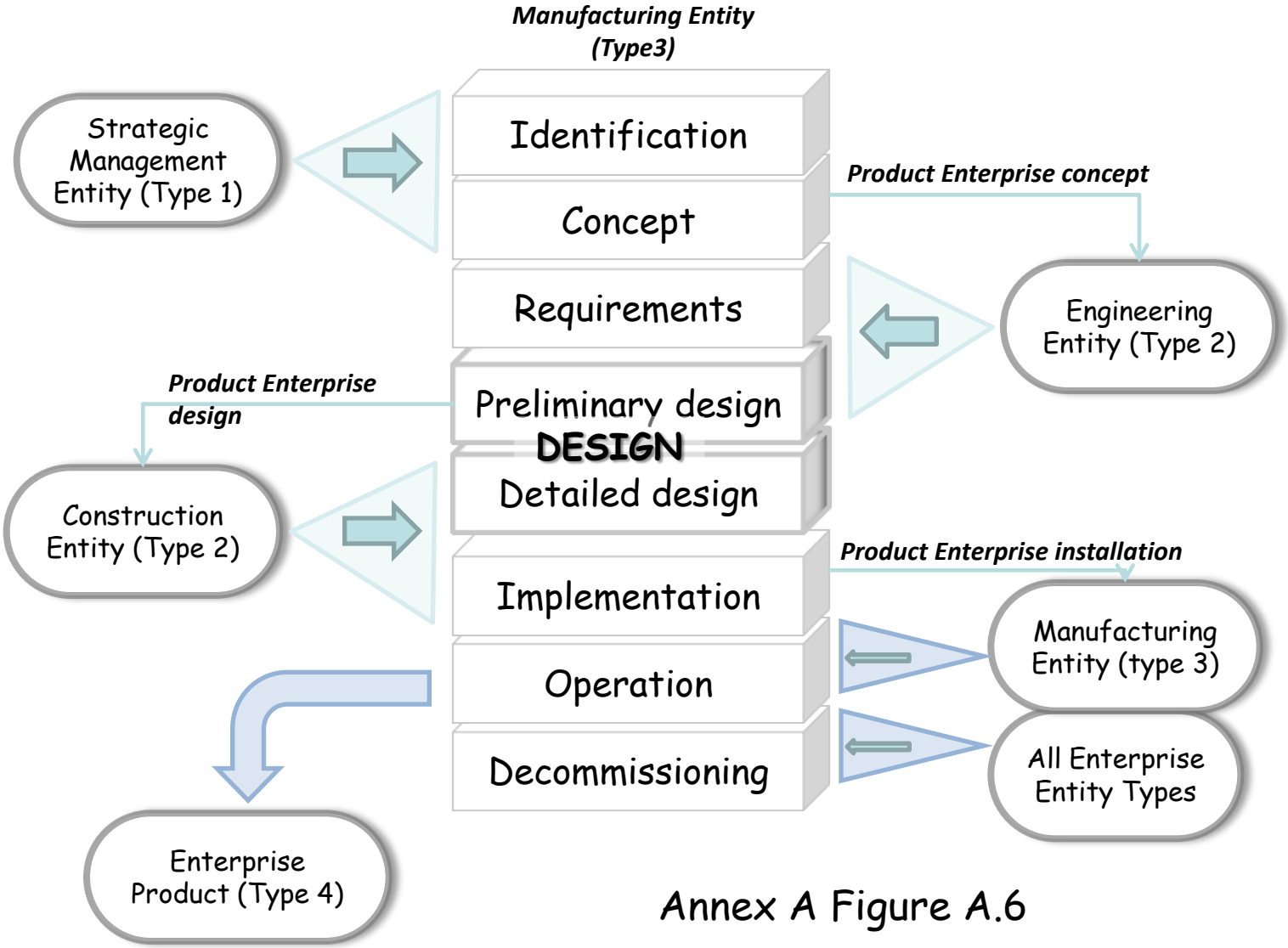
Many views possible



Architecture components

- Enterprise-engineering methodologies (EEM) for every type of life-cycle activity for any enterprise-entity type.
- Enterprise modeling languages (EML) or modeling constructs that allow the enterprise operation to be described.
  - Constructs shall allow users to represent the different elements and thereby improve both efficiency and understanding

GERA entity types



Annex A Figure A.6

Languages for people

The form of modeling constructs shall be adapted to the needs of people creating and using enterprise models.

EML shall be expressive enough to model human roles, operational processes and their functional contents and support

EML semantics can be described in terms of ontological theories but the definition of the formal semantics shall be supported by natural language explanations of the concepts.

Constructs of 19440

Domain

Business Process

Enterprise Activity

Event

Resource

Functional Entity

Capability

Decision Centre

Enterprise Object

Object View

Product

Order

Operational Role

Organizational Unit

Organizational Role

Person Profile

Construct template

Common format

- Header
  - Type label
  - Identifier unique to model
  - Name
  - Authority for design of construct
- Body
  - Descriptives in textual form
  - Relationships specified by reference

Construct descriptives

- Predefined for each construct
- User-defined by extension
- May be qualified (e.g. mandatory or optional)
- Possibly XML schemas, UML or EXPRESS notation
- Attributes
  - Name (meaningful in domain)
  - Data type (simple or complex)
- Complementary Concepts

Complementary concepts

Not fully developed as constructs but have particular significance and semantics for the purpose of enterprise modeling

- Behavior rule
- Constraint
- Declarative rule
- Functional operation
- Integrity rule
- Objective
- Performance indicator

Construct relationships

Model the dynamics between run-time instances

Types of relationship

- Operational authority and responsibility
- Membership in specialization
- Part of an aggregation
- Consist of an aggregation
- Other associations

Possibly reflexive

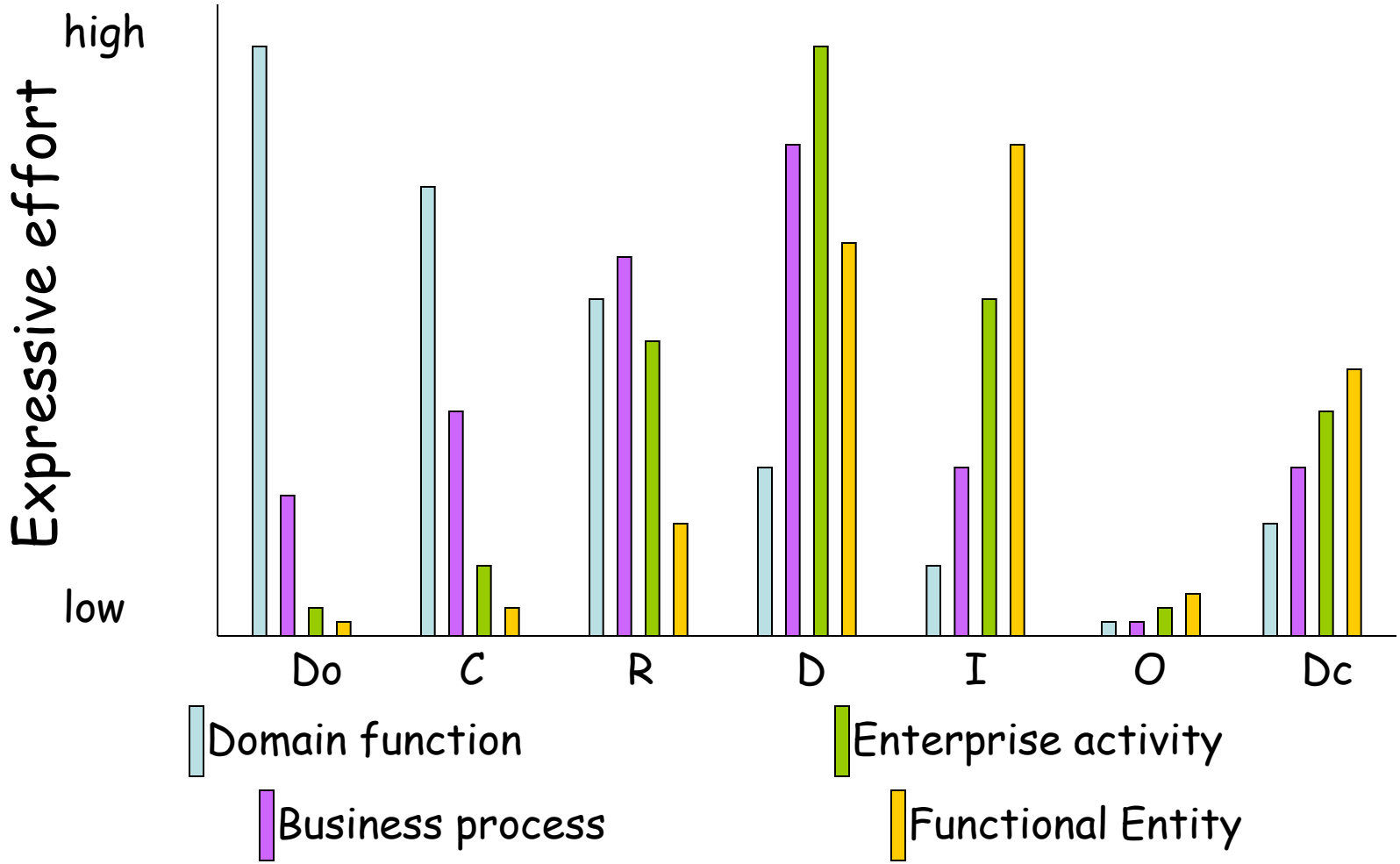
Constructs and roles

- Human organizational role captures assigned responsibilities and required capabilities (skills)
- Human operational role captures the operational capabilities of person assigned to a task
- Machine operational role captures the operating capabilities of machine assigned to a task
- Machine product role captured by attributes that describe input and output of activities to change state

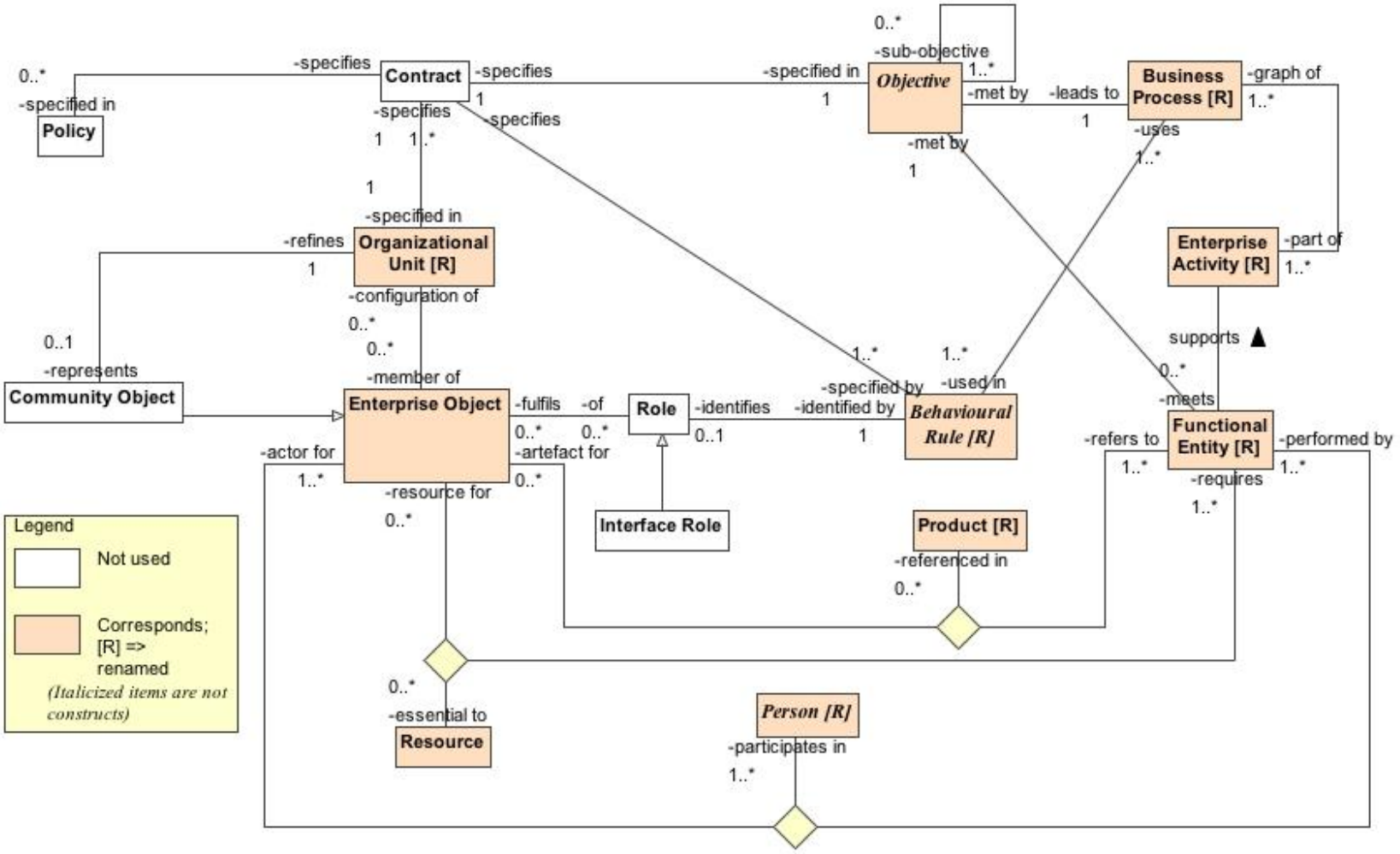
Construct across phases

Construct label	EO (Enterprise Object)
Identifier	<model-unique string>
Name	name of the Enterprise Object instance
Design Authority	[[<identifier> "/" <name>] [NIL | :" <identifier> "/" <name>]] of Organizational Role and Organizational Unit respectively, having authority to design or maintain this particular instance
Body ***A1 Descriptives relevant for all enterprise model phases***	
Description	short textual description
Nature of Object	PHYSICAL | INFORMATION
Properties	[<property_name> = <property_value>]+ – elements representing properties and their values for the entity represented by the Enterprise Object instance
Constraints	[<constraint>]* imposed on selected named attributes of the Enterprise Object instance
A2 Descriptives relevant for different enterprise model phases	
A2.1 applicable at concept definition and later phases	
Not applicable	
A2.2 applicable at requirements definition and later phases	
Integrity Rules	[<integrity rule>]* applicable to attributes of the Enterprise Object instance in the requirements definition phase

Life-cycle expression



ISO 19440 and ODP



- **Figure D.2 — Metamodel of ODP Community and Behaviour Concepts modified to fit this standard**

Process behavior

Shall be described in its behavioural rule set attribute by a set of *behavioural rules*, which control the sequence of constituent Business Processes and Enterprise Activities

- well-structured is completely defined
- semi-structured is known at run-time
- ill-structured is non-deterministic

Behavior rules

Shall enable

- the capturing of all the conditions that control the sequencing and the dynamic behaviour of Business Processes
- their presentation in both human and machine understandable form

Apply only to Business Processes

- Internal behaviour of an Enterprise Activity, the sequencing of its set of *functional operations,* is considered to be an implementation issue.

Formal syntax in BNF

behavioural rule set = behavioural rule {behavioural rule} ;

behavioural rule = WHEN condition DO action ";"

condition = condition term
| condition term {AND condition term}
| condition term {OR condition term} ;

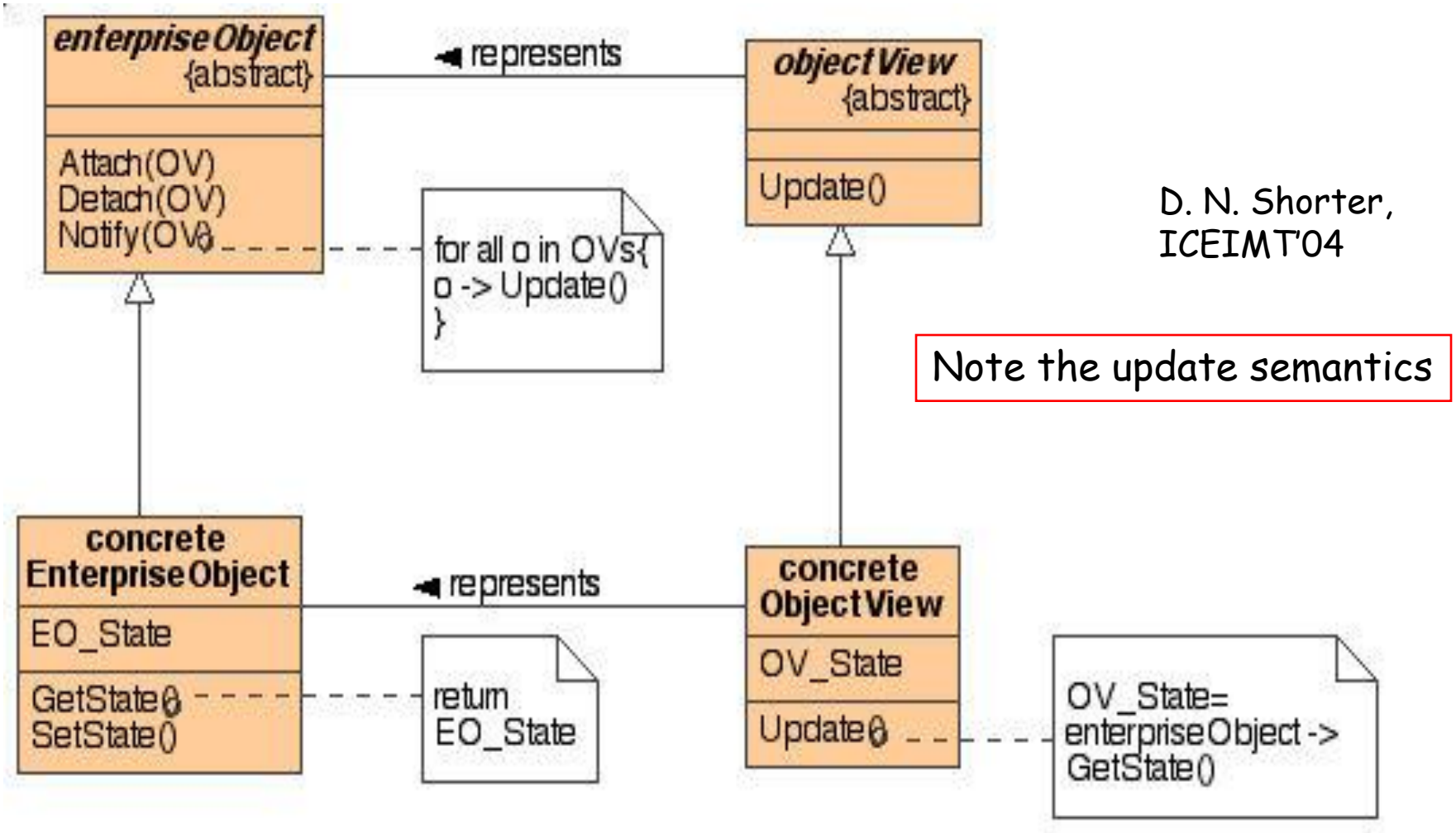
condition term = single condition | (condition) ;

single condition = event occurrence | action completion ;

event occurrence = event | start | exception ;

event = ev ; (*the named Event needs to be included in the containing Business Process's list of Event Inputs *)

Object view as pattern



D. N. Shorter, ICEIMT'04

Note the update semantics

Module representation

ERAM shall be able to represent the concept of enterprise modules, building blocks or systems (products, or families of products) for use as common resources in enterprise engineering and enterprise integration.

- One important set of enterprise modules is the integrating infrastructure or the set of integration-technology services required for enterprise engineering and operation in heterogeneous environments.

Operational system

One result of the enterprise-engineering process shall be a design or model for the enterprise-operational system (EOS).

The enterprise operational system shall consist of the hardware and software needed to fulfill the enterprise objectives and goals.

The content of the operating system is derived from enterprise requirements.

Glossary a must

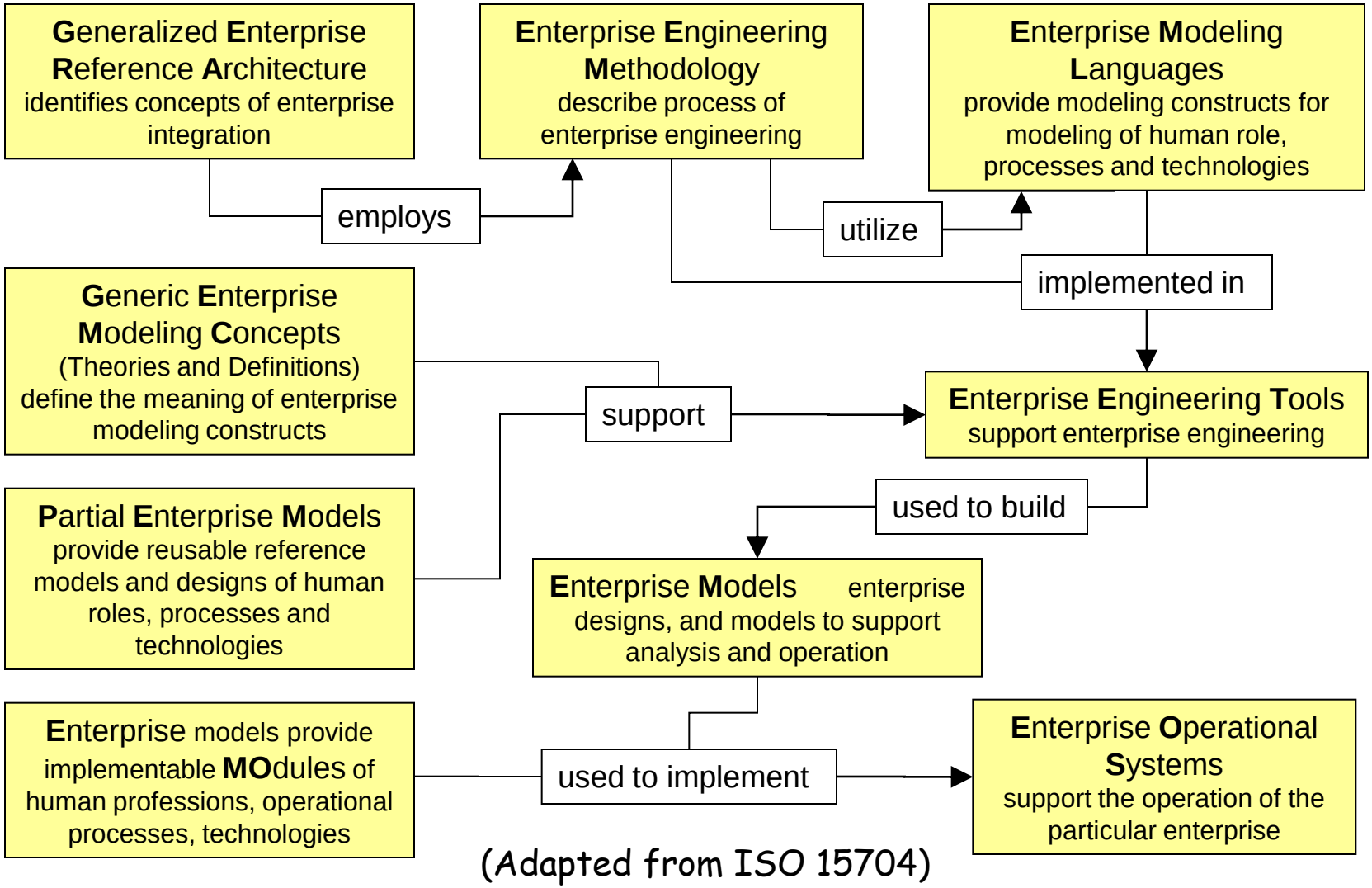
To promote understanding about projects and other co-operative efforts, ERAM shall provide

- consistent glossary and a semantics and syntax for use in enterprise-engineering and integration efforts, or
- reference to other suitable glossaries.

Afternoon break

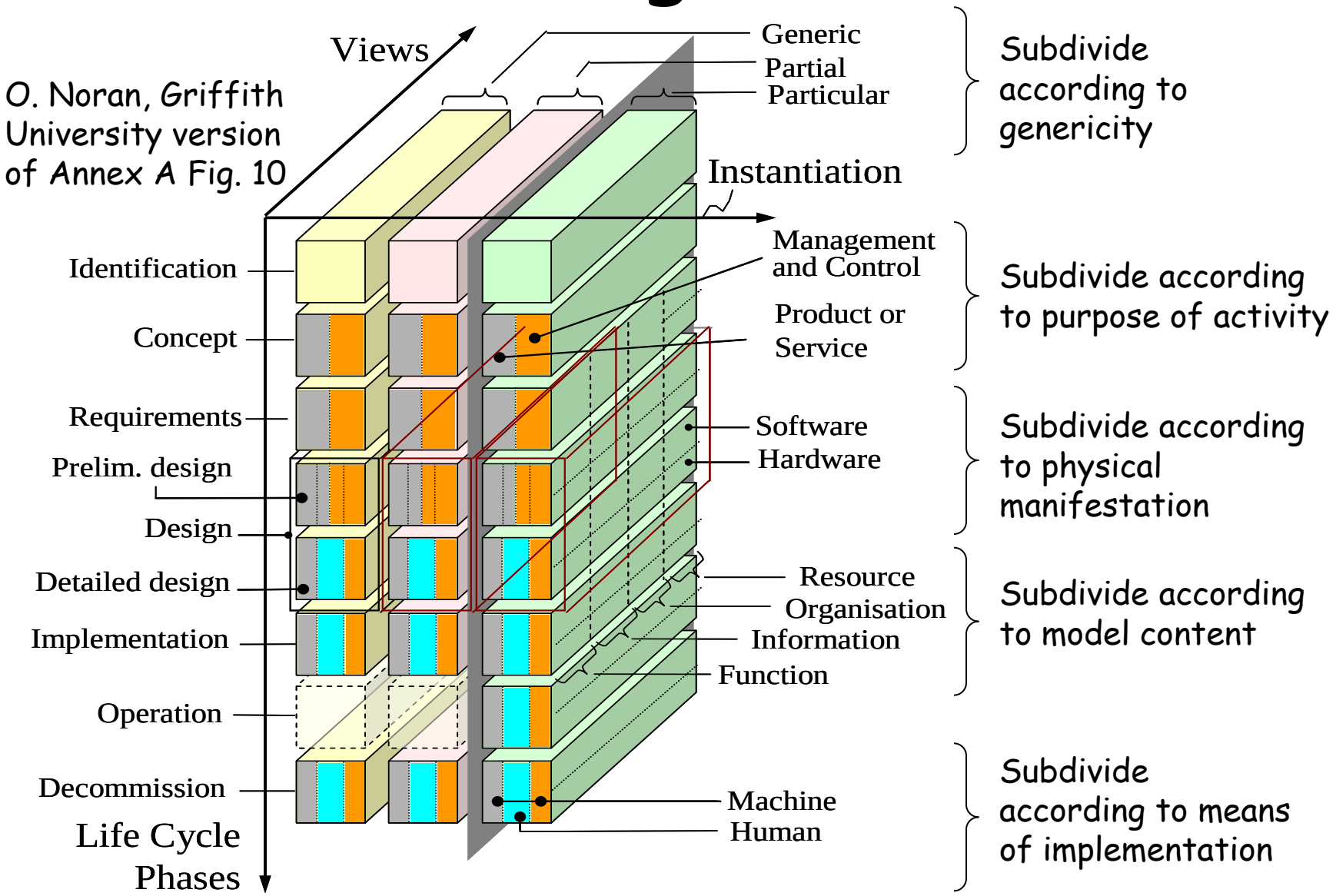
- What can we do to improve the value of International Standards dealing with EA that maintains their general applicability to international commerce?
- What value do enterprise archtiectures have with respect to system archtiectures?
- What is a reference architecture?

Annex A - Scope of GERAM



(Adapted from ISO 15704)

GERA modeling framework



ISO 15704 Compliance

Any assessment of the degree of compliance of a candidate architecture and methodology shall be qualified by the following:

- a preliminary statement as to whether or not they are model based;
- a statement of the degree to which they then conform partially or totally to the appropriate requirements

In the event of partial compliance, areas of non-conformance shall be explicitly identified.

Conformance to 19439

Shall have function and information view

Shall be able to derive resource and organization view

Shall provide distinct model phases

Shall provide for derivation of partial and particular model from generic constructs

Shall propagate model changes to all views

Conformance to 19440

Shall either use the constructs as defined or be able to map to the constructs

Can claim qualified compliance by using a subset of constructs or mapping to a subset of constructs

Can be a valid construction of a compliant modeling language

Shall identify construction and model execution testing levels

So what!

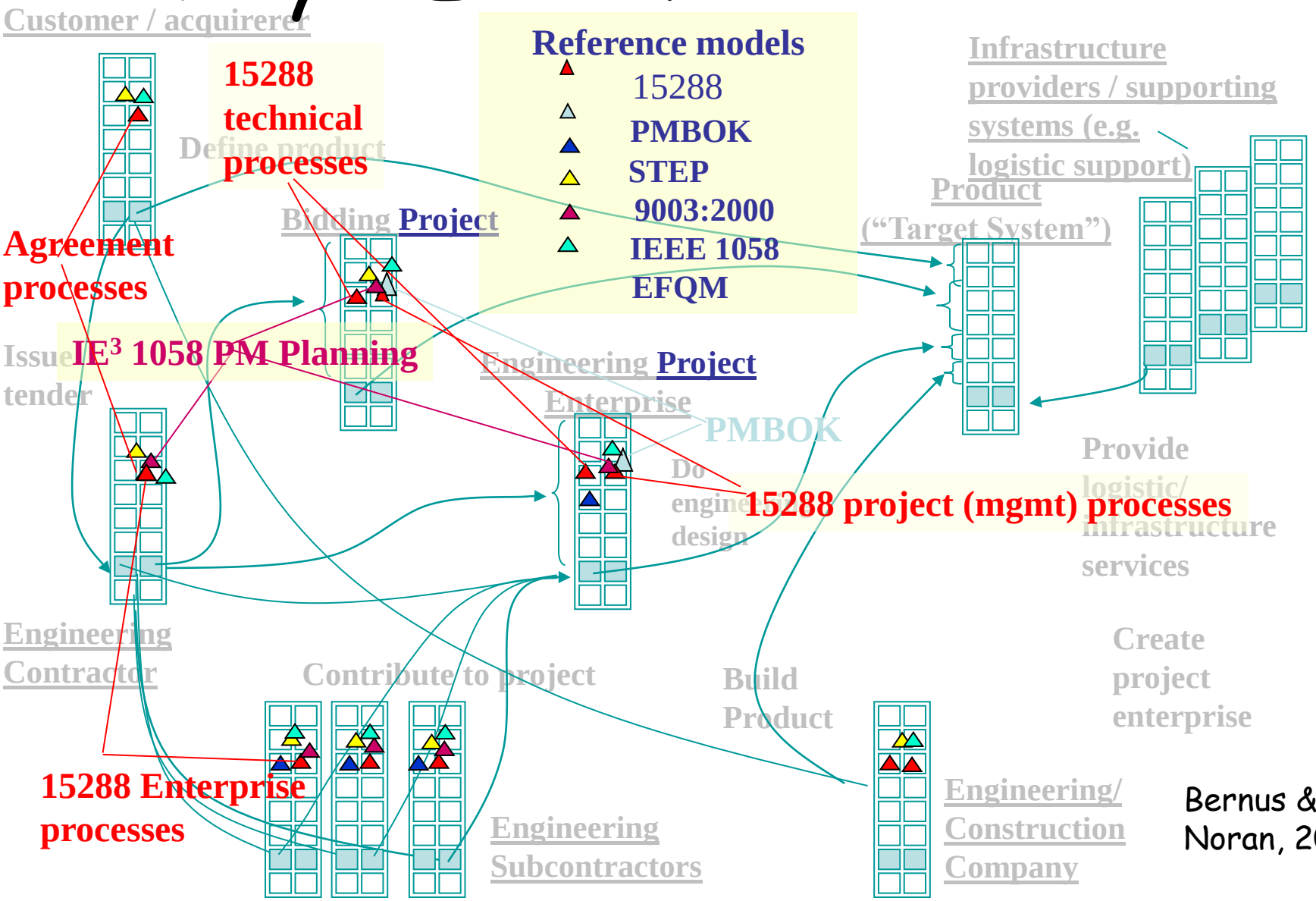
Why the comparisons are important: extracting the architectural value.

Why the International Standards are important: they enable commercial activity to compete in a well-defined market.

Integrating standards

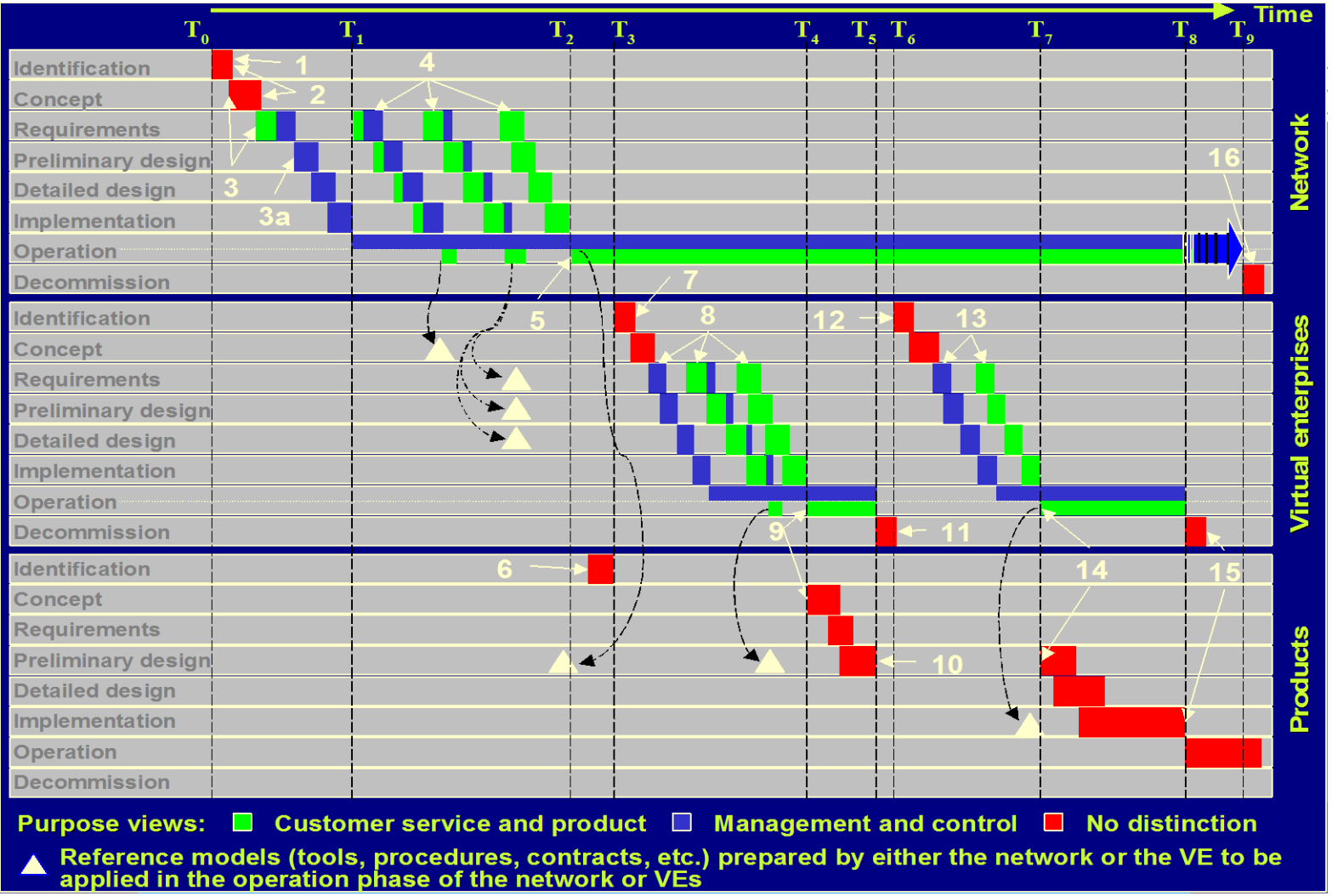
- One standard is never enough!
  - Limited by scope statement
  - Driven by normative content
- Multi-standard challenge
  - Crafted by people with different training and experience
  - Competing for product recognition and acceptance
- Not always easy but is done every day!

Many GERA & standards



Bernus & Noran, 2004

Life history example



3 GERAM instances linked by response to events and reference models

Source: J. Vesterager, P. Bernus, J. Pedersen & M. Tolle, The what and why of a Virtual Enterprise Reference Architecture, in E-work and E-commerce: Novel solutions and practices for global networked economy. B. Stanford-Smith and E. Chiozza (Eds) IOS Press, Amsterdam (2001) Used with permission

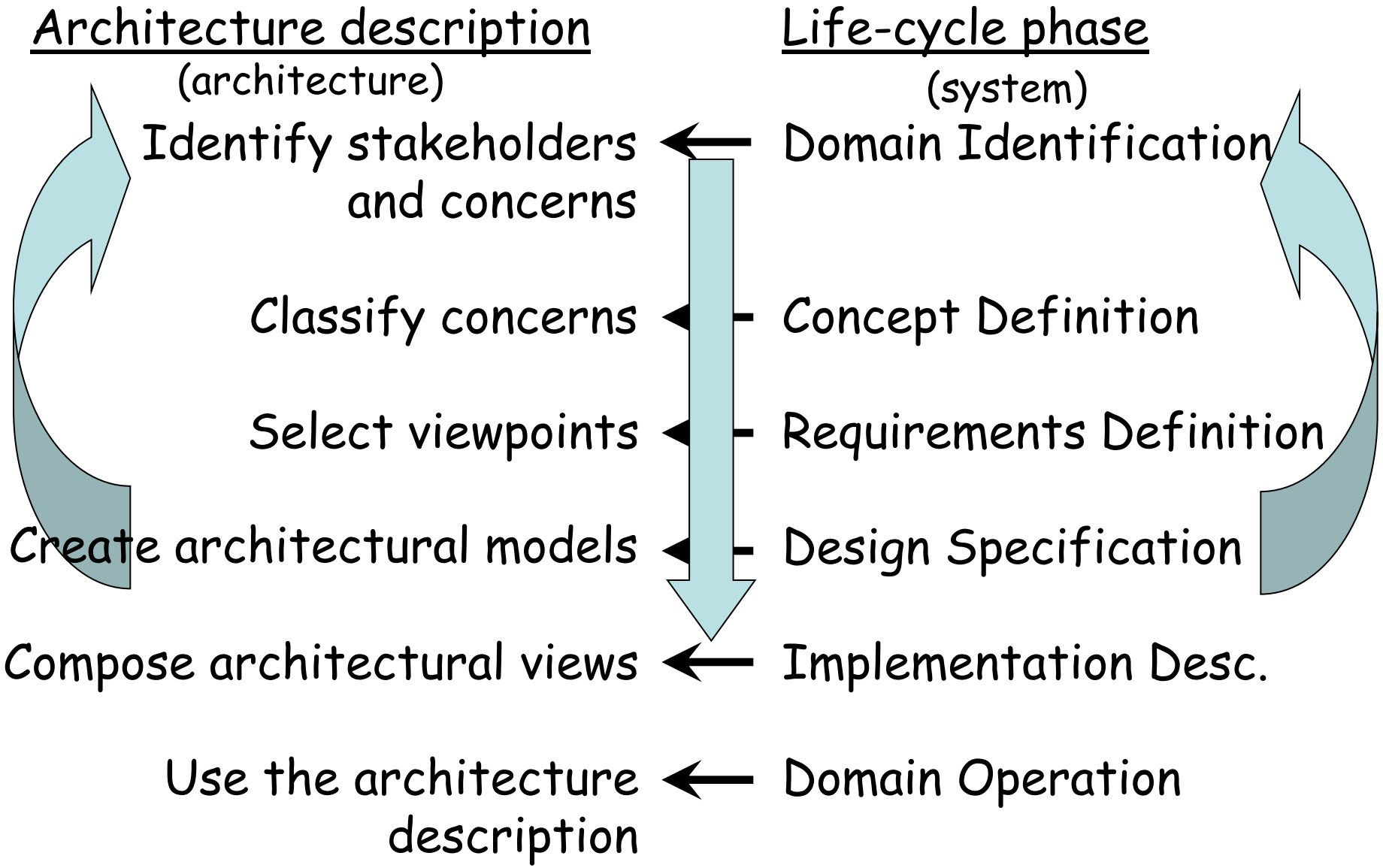
Do 15704 and 42010 fit?

- Check for conformance
  - Are stakeholders and concerns explicit?
  - Is there a viewpoint for concerns?
  - Are there corresponding views?
  - Is there rationale provided?
- What about the constraint that a viewpoint has only one view?
  - Each life-cycle phase has a different AD
  - AD's can be generic, partial or particular

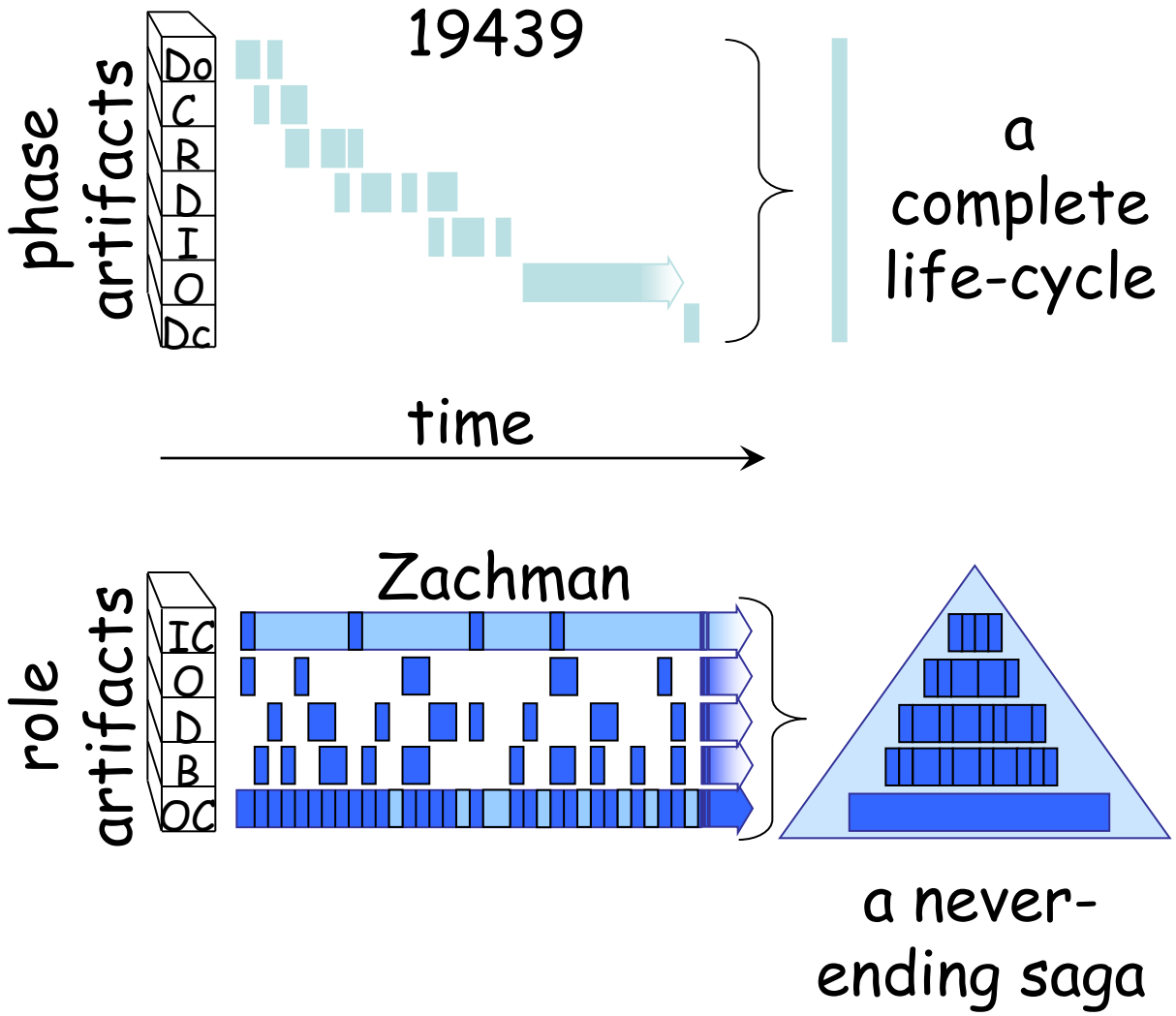
Another fit for 15704

- Now consider 15704 as an architectural viewpoint for the enterprise system
  - The viewpoint specifies many (sub)views of three kinds: phase, view, and genericity
  - The architectural view, expressed as many (sub)view enterprise models, spans the whole enterprise from concept through demise
- 15704 is a reference enterprise architecture and the standard is its AD

Using 15704 to build AD's



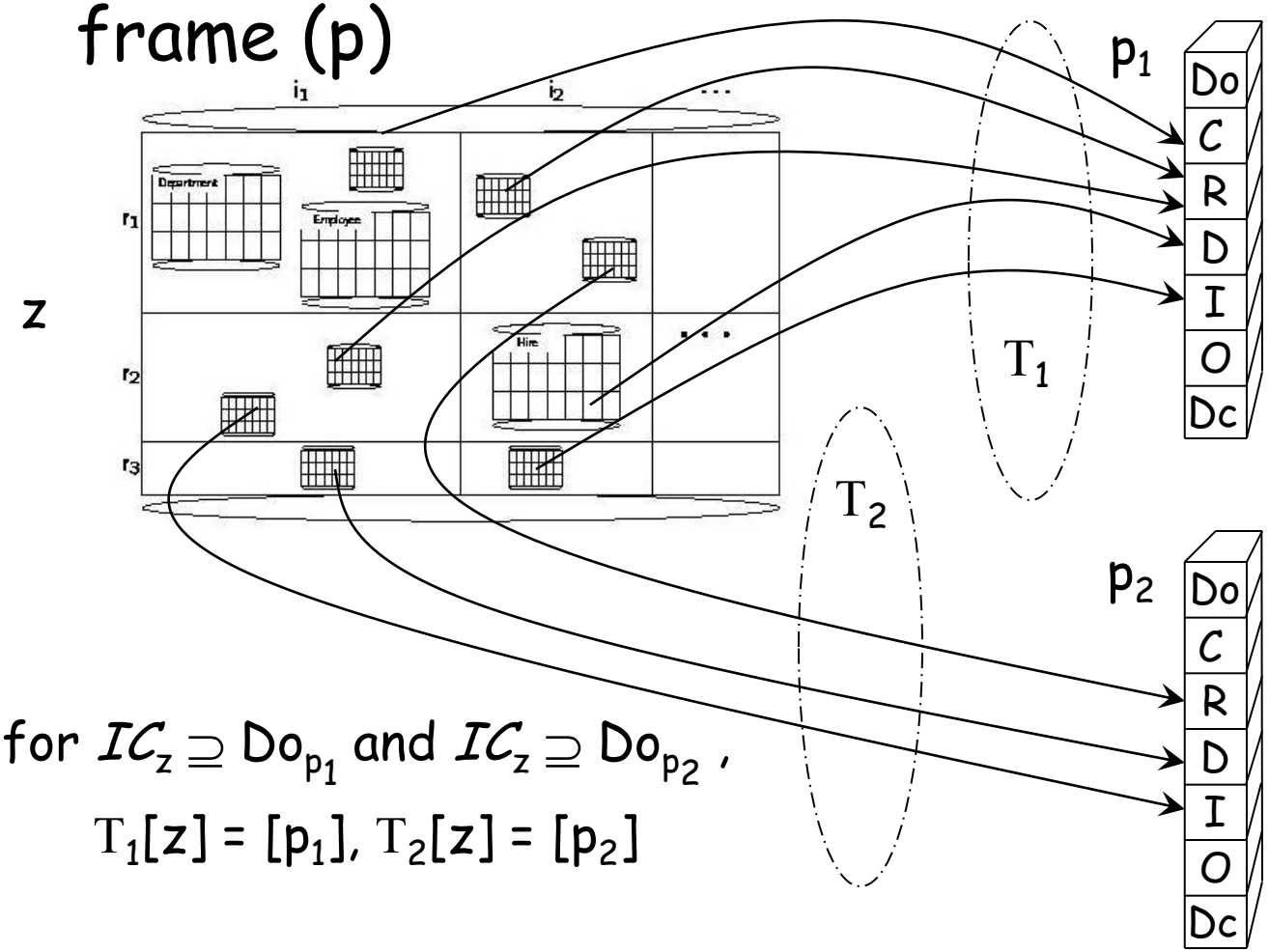
Different Life History



The appearance of artifacts in time imposes a temporal order on the purposive dimension of 19439, whereas the Zachman purposive dimension order is strictly the result of dependency among artifacts.

Taking a Snapshot

A Zachman continuant frame (z) can participate in an 19439 occurrent frame (p)

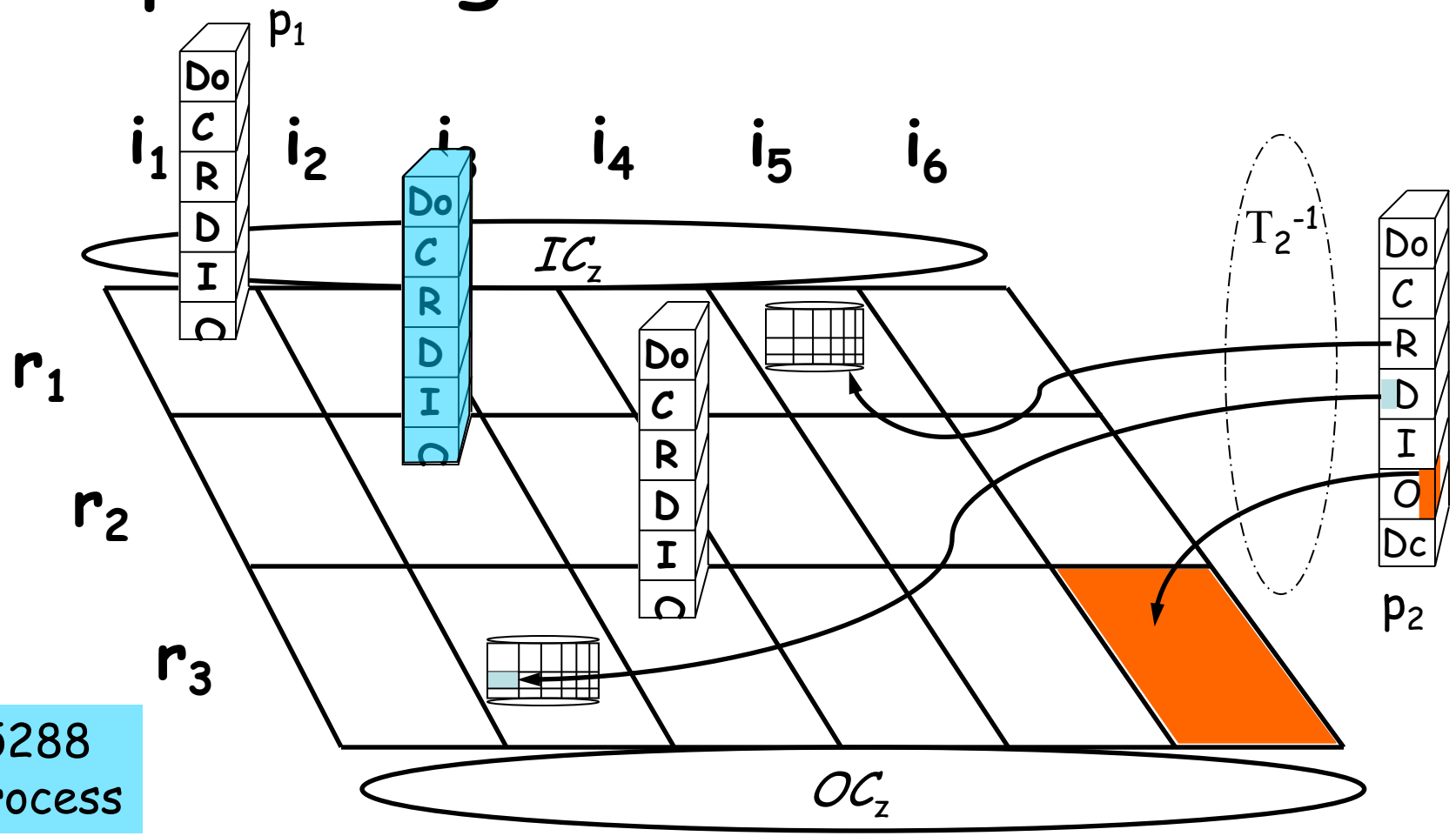


for $IC_z \supseteq Do_{p_1}$ and $IC_z \supseteq Do_{p_2}$,

$T_1[z] = [p_1]$, $T_2[z] = [p_2]$

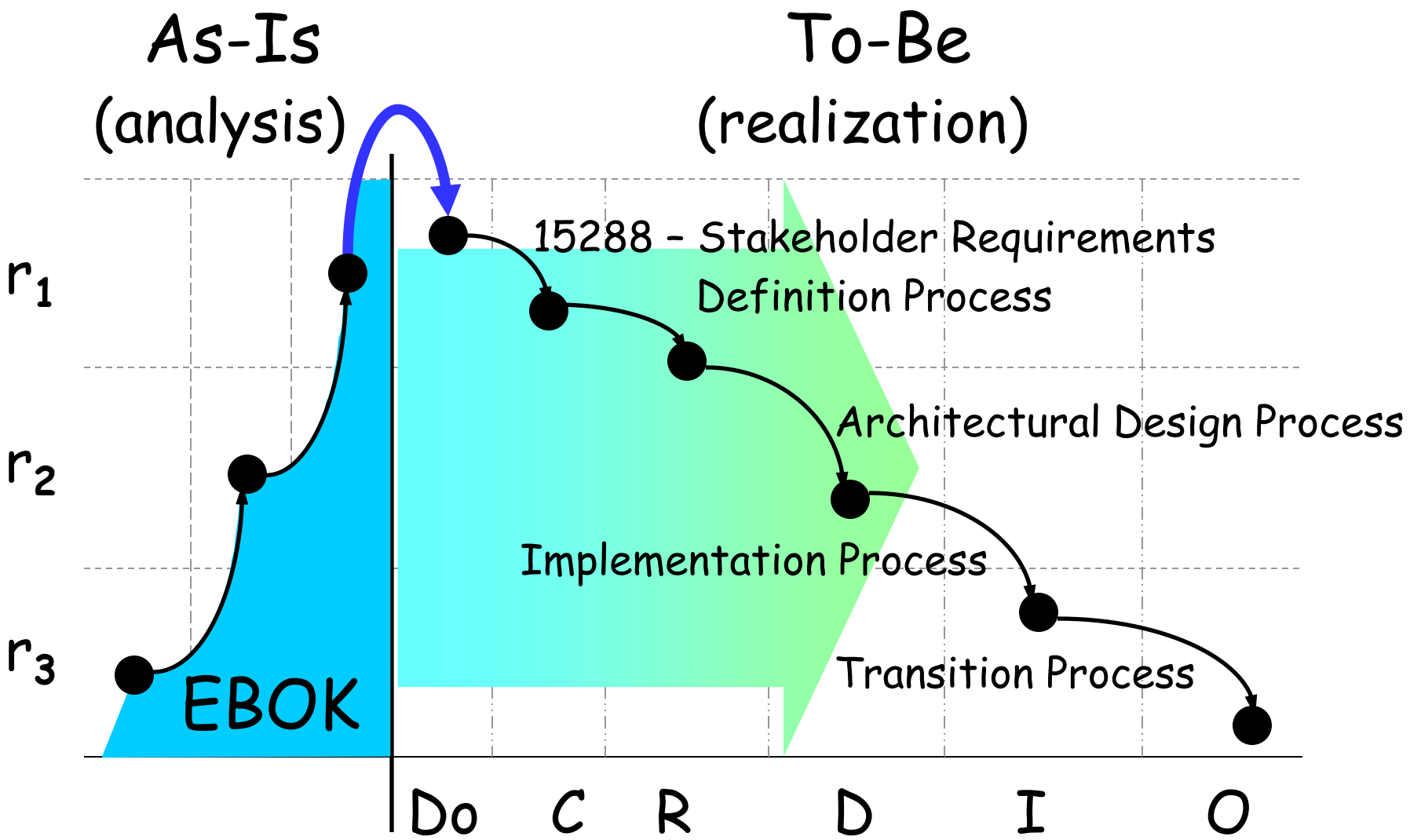
15288 processes from "how" column map to p1 and p2 function views

Populating with Artifacts



for $IC_z \supseteq Do_{p_1}$ and $IC_z \supseteq Do_{p_2}$, $T_1^{-1}[p_1] \subseteq [z]$ and $T_2^{-1}[p_2] \subseteq [z]$

Profile of Change



Another way to view it

ISO 14258:1998 Figure 1 Transposed

Issue-solving activities / Phase		Plan and Build Phase (e.g., before sell/buy title transfer)	Use and Operate Phase (e.g., after sell/but title transfer)	Dispose and Recycle Phase (e.g., after product is no longer useful)
		w| How |w|w|w|w	w| How |w|w|w|w	w| How |w|w|w|w
Specify	"What" Activities	•Develop goals •Define strategy •Define product needs	•Define support needs •Define Use	•Define recycle/dispose needs
Design	"How" Activities	•Develop Requirements •Define concept •Design product •Plan to produce product •Plan to support product	•Define Use Requirements •Define Support Requirements	•Define recycle/dispose requirements
Build Operate	"Do" Activities	•Procure parts •Produce product •Test product •Ship product	•Use the product •Support product	•Recycle product •Dispose product

Managing Change

To respond to a change in the environment of **z**,
say widget W for customer C requires a new process P,
we use components of continuant **z** to instantiate the occurrent **p** that realizes the new process operation in one of two ways:

$T_{W,C}[\mathbf{z}] = [\mathbf{p}_{W,C}]$	document the current P
$M : \mathbf{z} \rightarrow \mathbf{z}'$	modify **z** for new process
$T_{W,C}[\mathbf{z}'] = [\mathbf{p}'_{W,C}]$	create new process realization

or

$T_{W,C}[\mathbf{z}] = [\mathbf{p}_{W,C}]$	document the current P
$R_{W,C} : \mathbf{p}_{W,C} \rightarrow \mathbf{p}'_{W,C}$	realize new process P'
$T^{-1}_{W,C}[\mathbf{p}'_{W,C}] \subseteq [\mathbf{z}']$	document new **p** in **z**

Overall Assessment 1

Many semi-independent international architecture standards & activities

Too many conflicting terms, many perspectives, varied applications

US/DoD architecture perspectives (DODAF, CADM, FEAF) – DODAF 2.0 ramping up

INCOSE Architecture related WGs biased towards DoD/US Gov't perspectives/needs

ISO TC184/SC5 EA Standards biased to Int'l & Manufacturing Industry Architecture Perspectives

Overall Assessment 2

- Harmonization is critical, but harmonizing these diverse international enterprise architecture points of view together is a major challenge
- Being involved provides an opportunity to contribute toward and learn about the future enterprise architecture for more intelligent enterprise realizations

Contact

Richard Martin

- Convener: ISO TC184/ SC 5/WG 1
- richardm@tinwisle.com ; tinwisle@bloomington.in.us
- www.tinwisle.com

We acknowledge the contributions of Johan Bendz, Rich Hilliard, David Emery, James Martin, Mark Walker, and Edward Robertson, Sandeep Purao

Graphics assistance from K. P. Martin, Indiana University