Architectural Principles for Enterprise Frameworks

Richard Martin
Tinwisle Corporation
Bloomington, Indiana

Edward Robertson and John Springer
Computer Science Department
Indiana University

Architectural Principles for Enterprise Frameworks

- Landscape
- Sources
- Principles
  - Characterization
  - General
  - Framework
- Formalization

The Framework Audience

- Users of categorical comparison
  - Partitioned dimensions and domains
  - Intuitive and formal relationships
- Enterprise participants
  - Stakeholders
  - Model builders
  - Model users
  - Developers of modeling tools
  - Research engineers and scientists

Our Framework Effort

- Formalism published in 1999
- Presented to business and scientific community – see EMMSAD'00
- On-going assessment of applicability to published "enterprise frameworks"
- Continuing research activity - viewing
- Evolution of "enterprise architecture"

Our EMMSAD'04 Goals

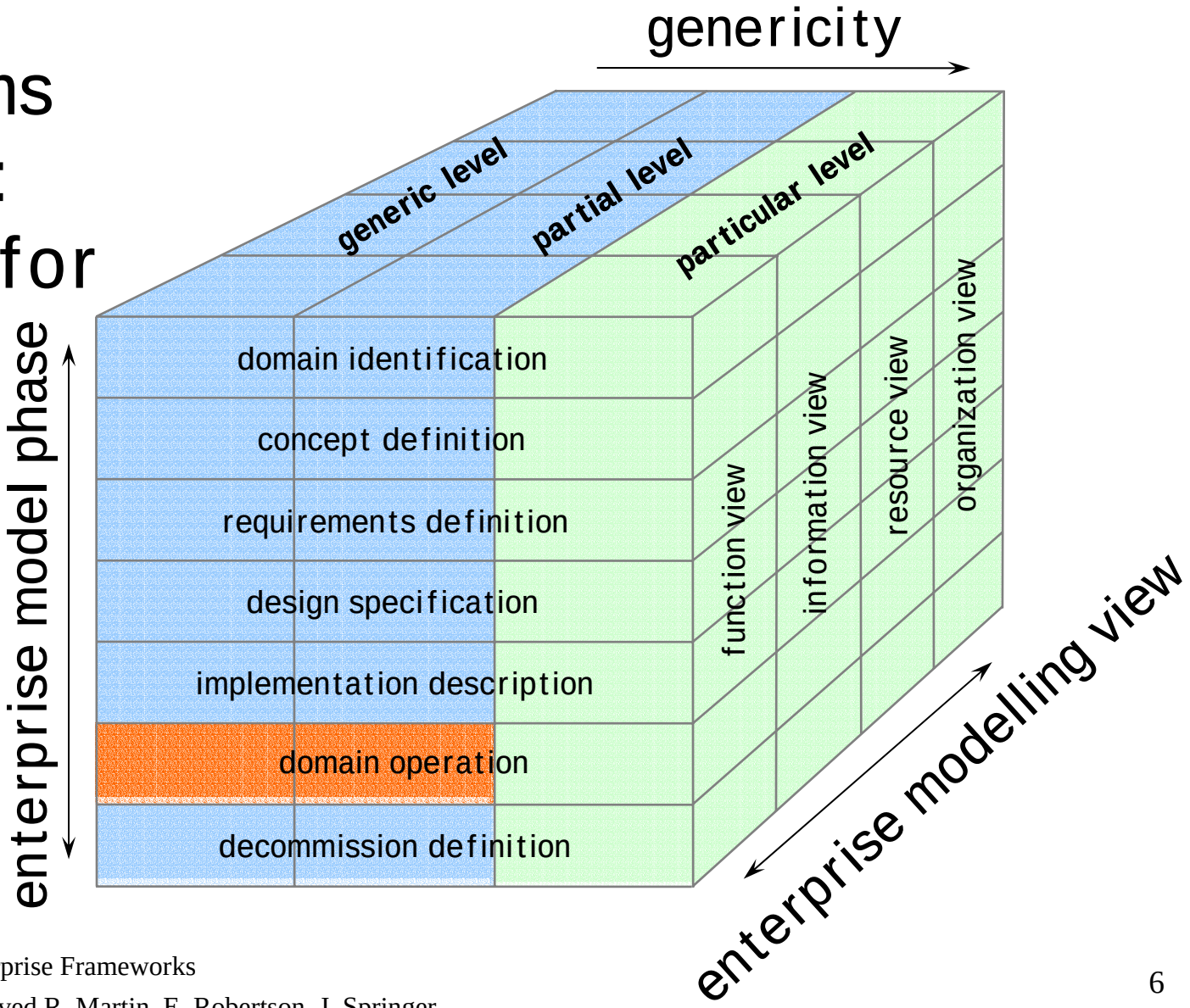
- Principles are "Requirements Specification" for formalization
- Seek your input on principles & approach
  - Do they reflect your experience?
  - Do they cover necessary aspects of architecture?
  - Do they address the real enterprise-level issues?

Origins of Principles

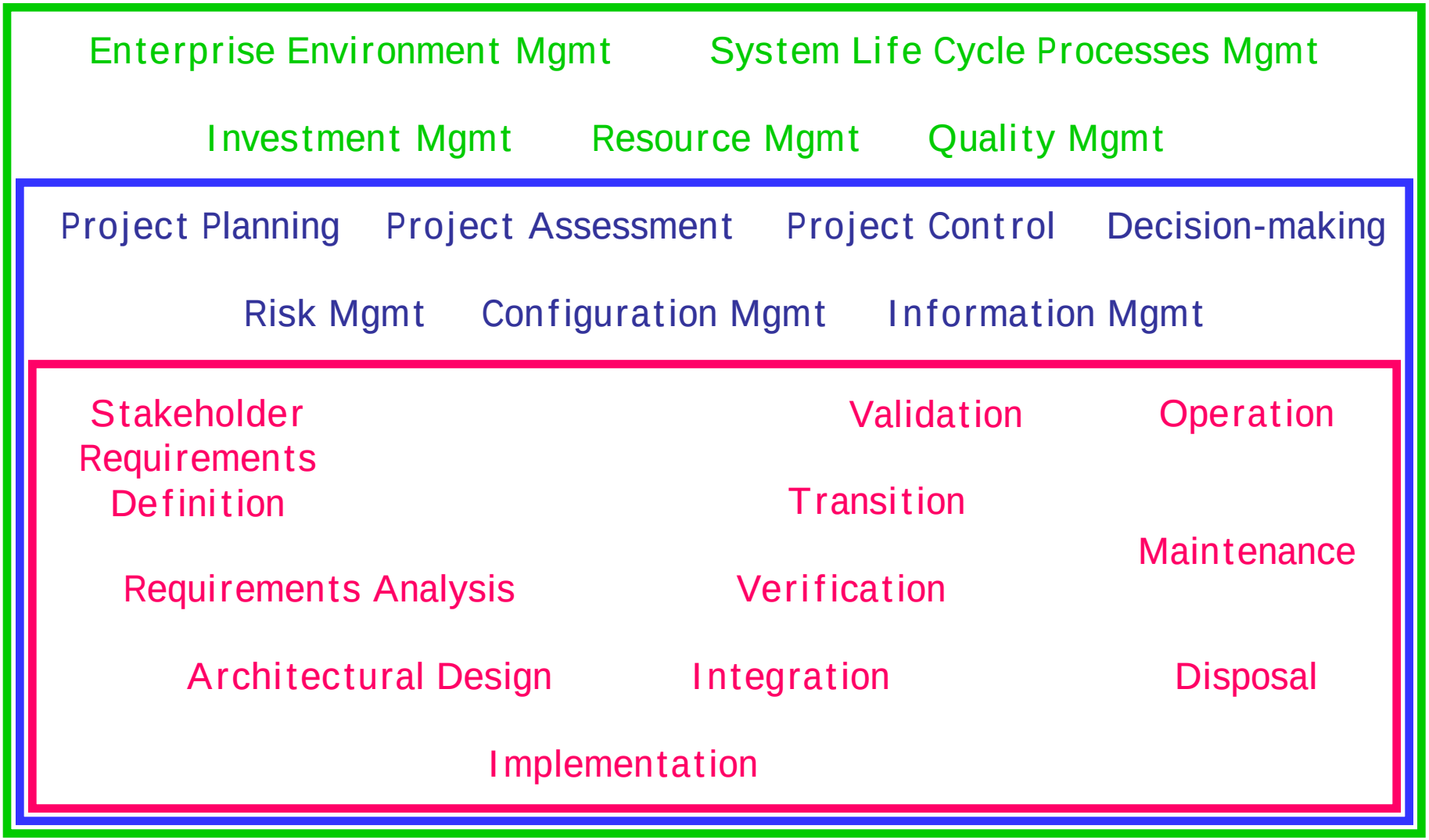
- International Standards
  - ISO/CEN FDIS 19439 CIM Systems Integration: Framework for Enterprise Modelling
  - ISO 15288:2002 Information Technology – Life Cycle Management - System Life Cycle Processes
- Industrial & Governmental Models
  - Zachman Framework for Enterprise Architecture
  - C4ISR (United States Department of Defense)
- Professional Experience

ISO/CEN FDIS 19439

CIM Systems Integration: Framework for Enterprise Modelling



15288 - Process Hierarchy



C4ISR Version 2.0

Architectural Views

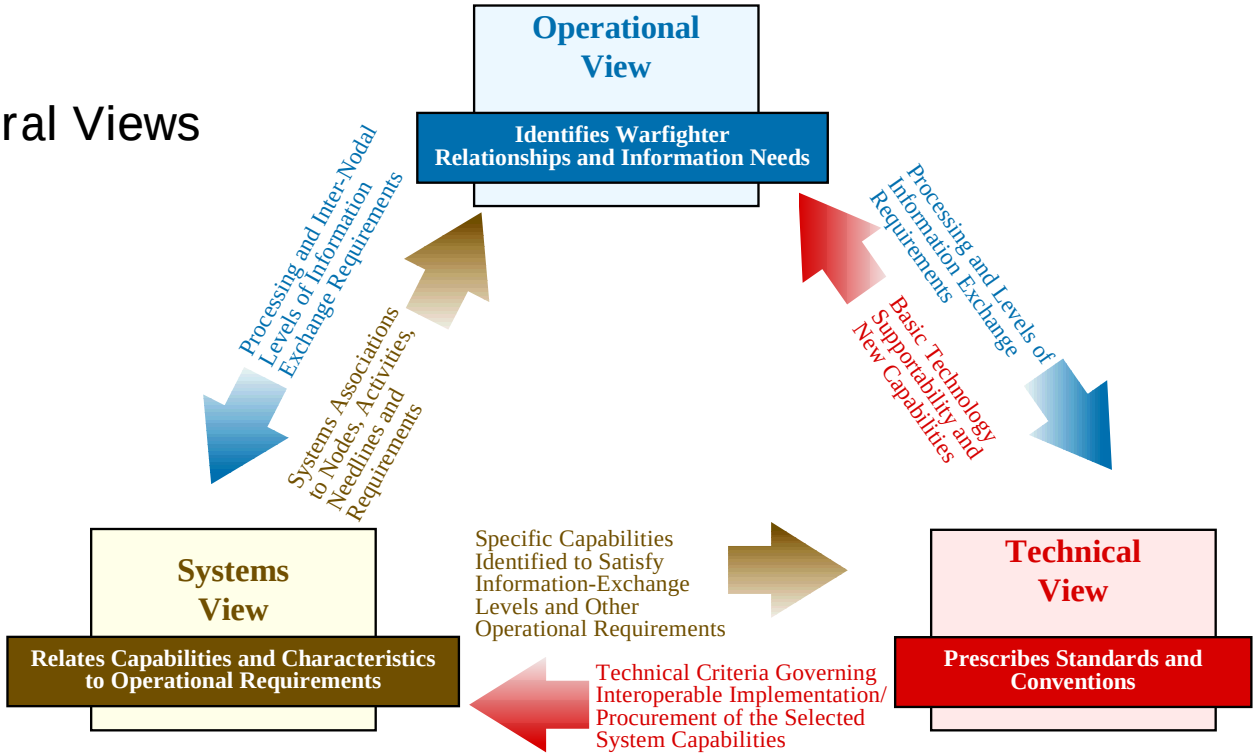


Figure 2-2. Fundamental Linkages Among the Views

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

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

Roles ↓

Interrogatives →

(used with permission)

Professional Experiences

- Observing our practice
- Performing model integration
- Developing international standards
- Teaching software engineering
- Managing in enterprises
- Participating in workshops

Framework characteristics

A containment structure

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

NOT a programming framework.

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. Separate model and instance decompositions – do not confuse meta-levels.
5. Dependency is not chronology
6. Don't hide architecture in methodology.

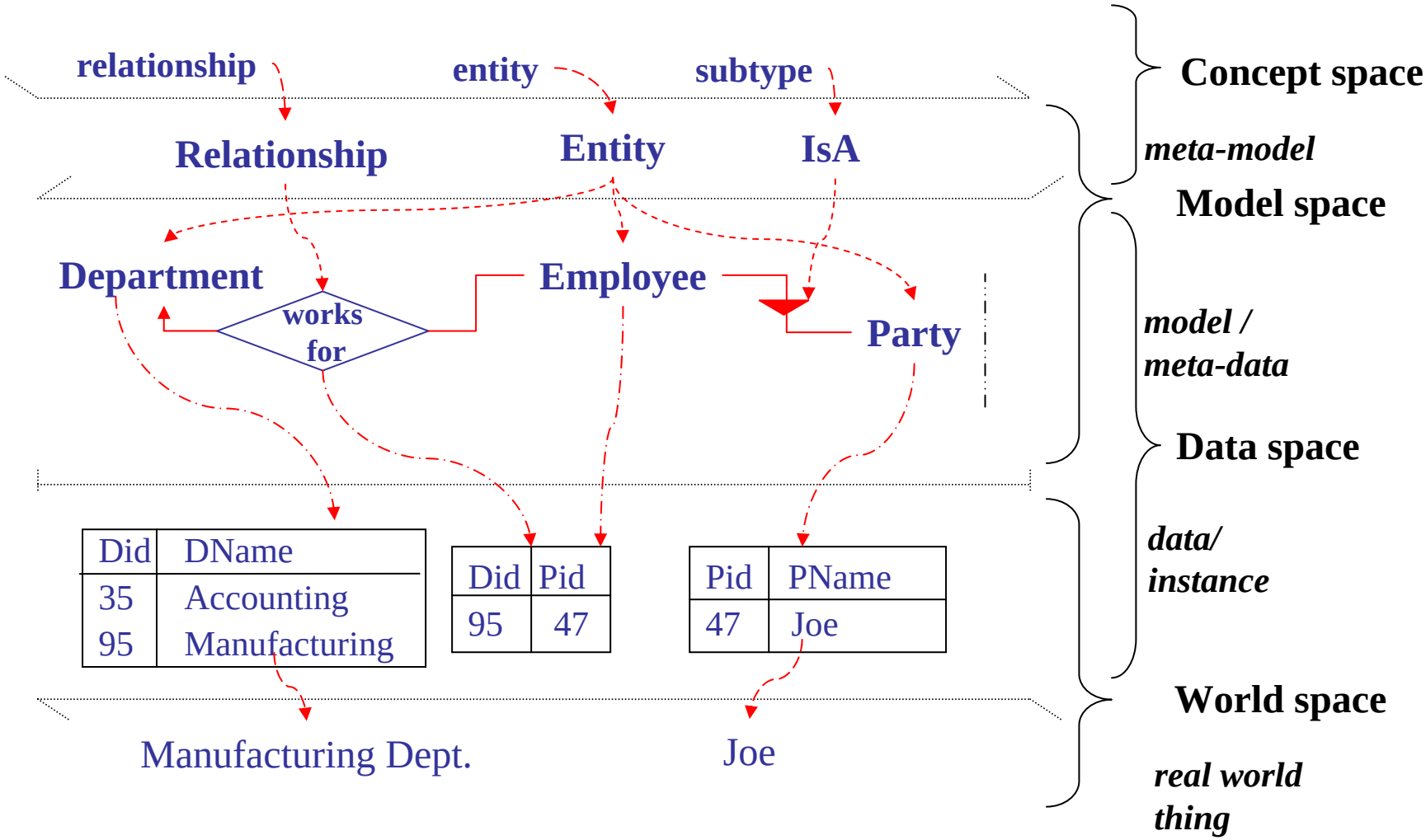
Framework Principles

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),
    scope (general to special) and
    refinement (coarse to fine).

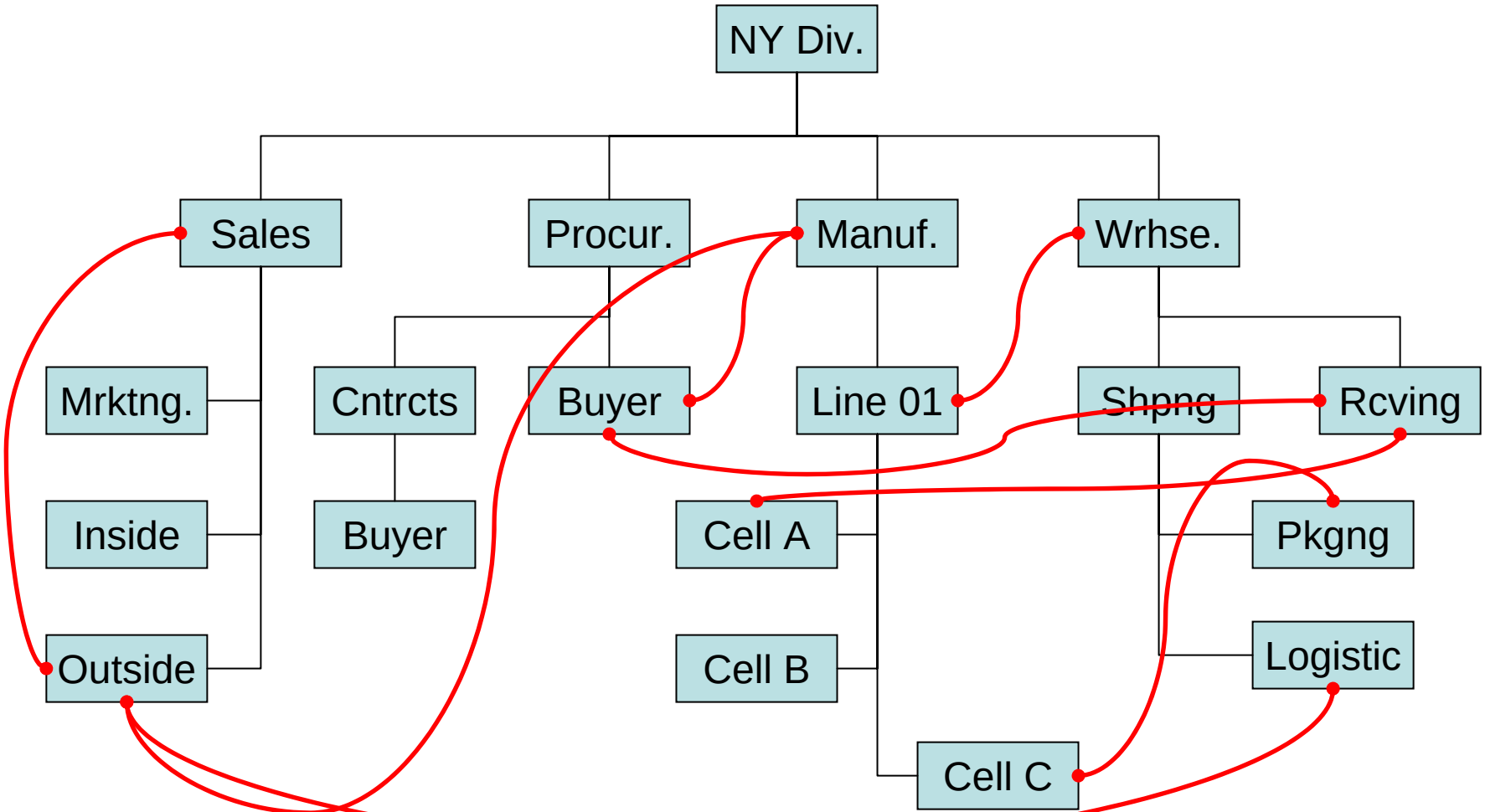
Framework Principles

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.
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.

Meta-confusion

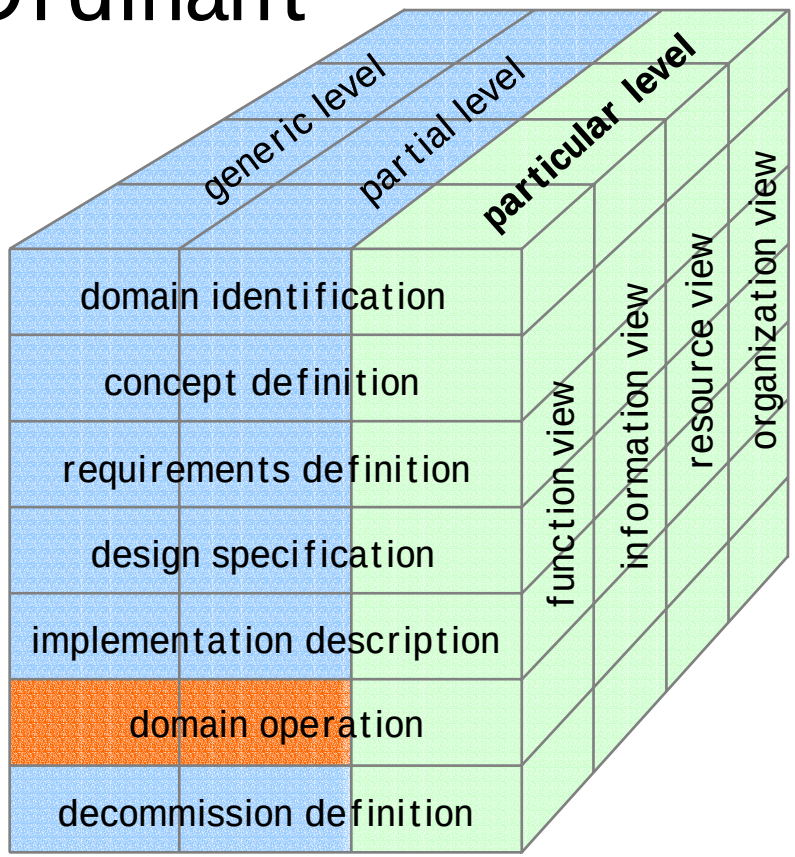


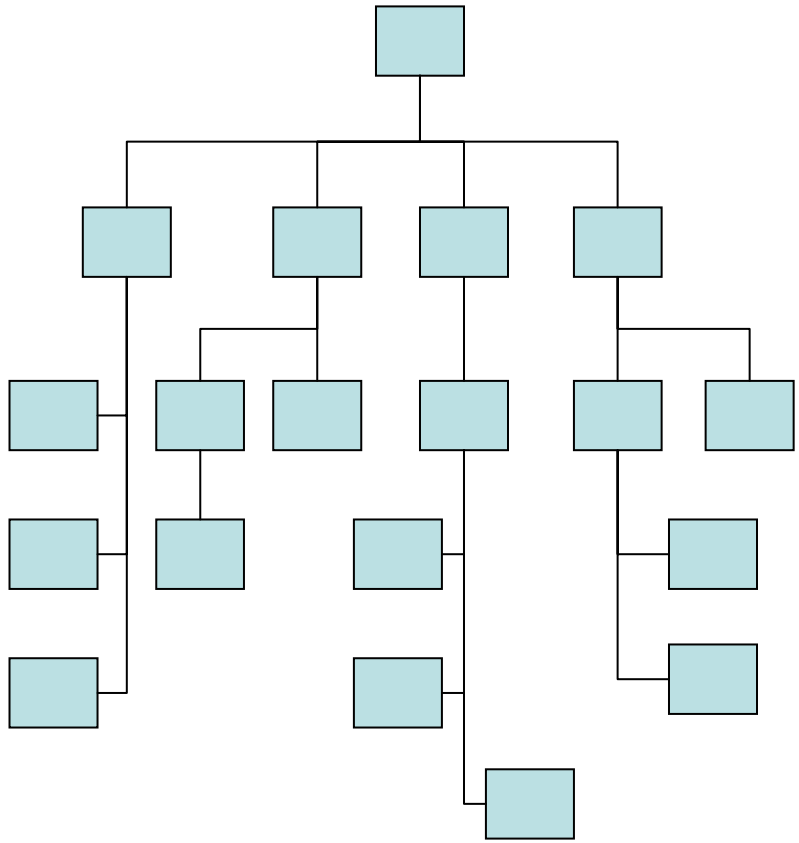
Distinguish structure from connectivity



Two structural aspects

Ordinant



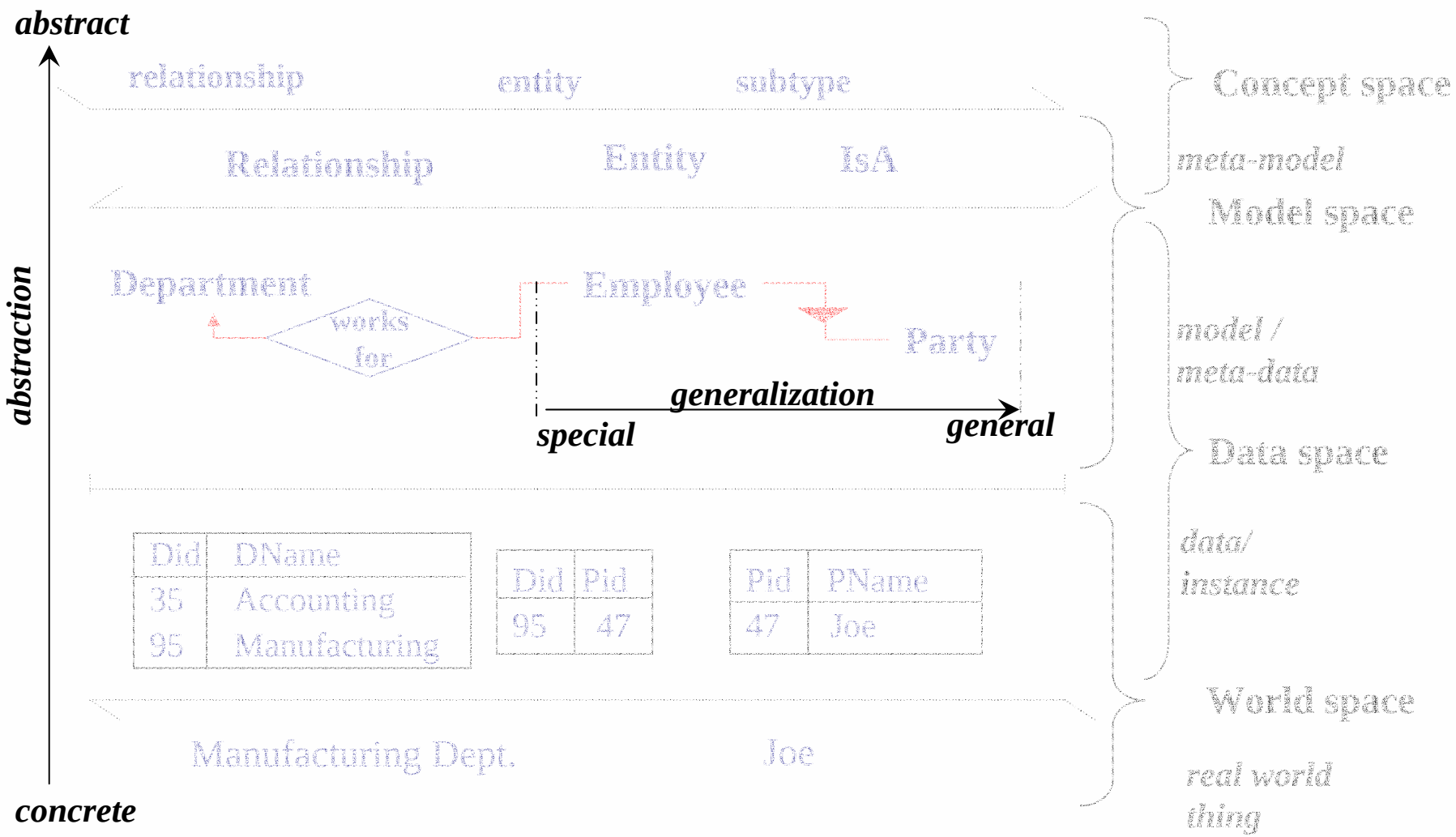


Decomposition

Three aspects of scale

- Abstractness, scope, and refinement
- Examples of dimensional independence:
  - E-R diagrams are abstract but have rich refinement when fully populated.
  - 19439 Genericity contains constructs for use along a generalization gradient with a range of phase abstractions.
  - Zachman interrogative proto-types are abstract with concrete model contents.
  - C4ISR products span operational abstractions with technical refinement.

Scope dimensions



Purposeful dimension context

- Zachman: **Role**

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

- ISO\CEN FDIS 19439: **Model Phase**

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

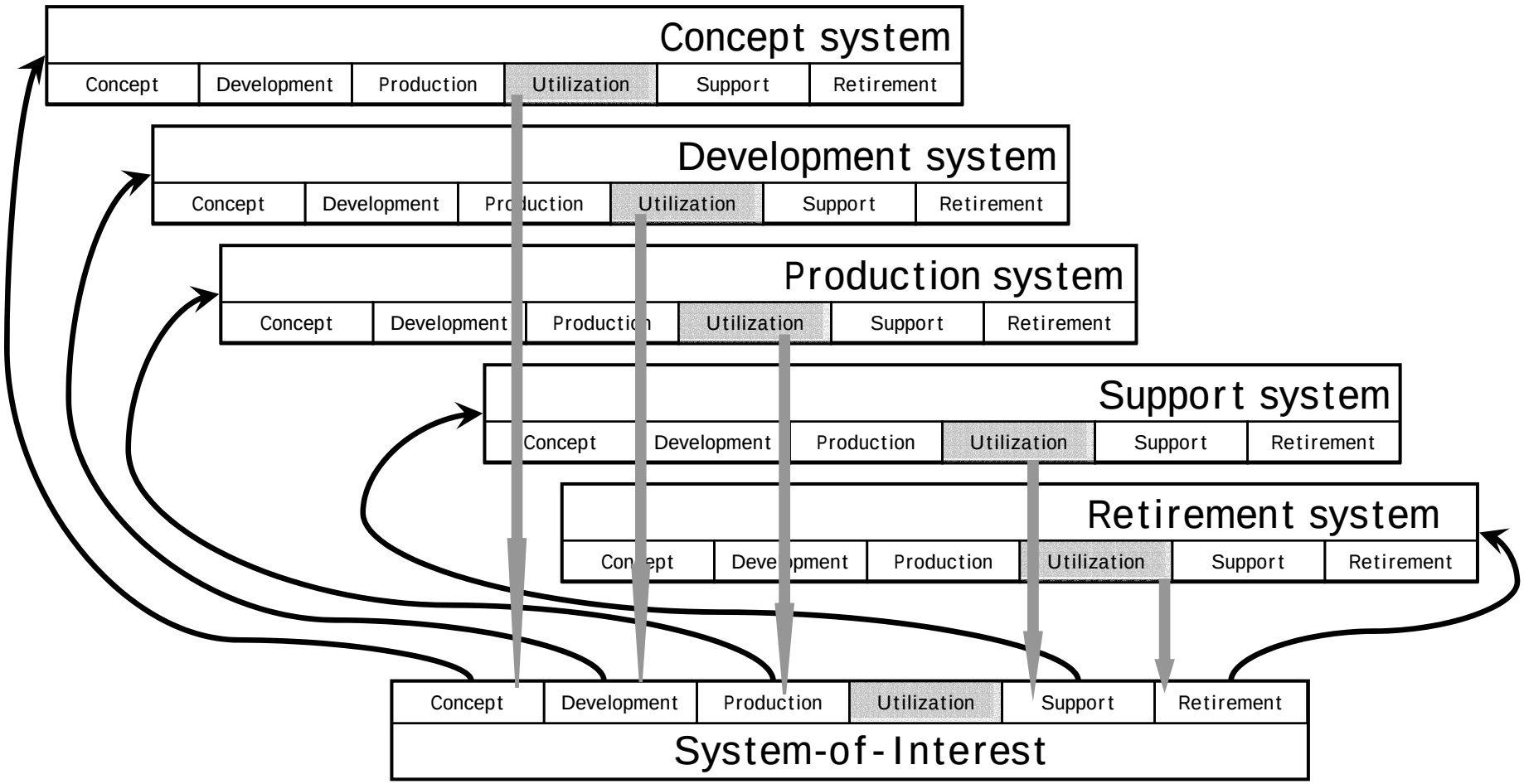
- ISO 15288: **Process Group**

  {Agreement, Enterprise, Project, Technical}

- C4ISR: **Guidance**

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

Recursive refinement cf. ISO 15288



Views are important

- For communication and analysis
- Examples:
  - ISO\CEN FDIS 19439: **View**
    {Function, Information, Resource, Organization}
  - C4ISR: **View**
    {Operational, Systems, Technical}
  - C4ISR: **Integration**
    {National, Theater, CJTF, Tactical}
- A static collection of views is insufficient.
  - ISO 15704 Amendment 1: **Economic View**

Toward Formalization

- **Structure**:
  - both tree (decomposition) and grid (ordinant)
  - frames and sub-frames
- **Connections**:
  - between frame components
  - respects purposive order
- **Constraints**:
  - model and instance
  - beyond structure and connection
- **Views**:
  - generalizes “view” in existing frameworks
  - defined on structure
  - attempts to carry forward connections and constraints

Framework meta-meta model

branch frames:

$F_\alpha \quad \langle IC_\alpha, OC_\alpha, SF_\alpha, \Phi_\alpha \rangle$

leaf frames:

$F_\alpha \quad \langle IC_\alpha, OC_\alpha, \quad S_\alpha \rangle$

Zachman specific

where

$IC_\alpha \quad \subseteq D$

$OC_\alpha \quad \subseteq D$

$\left.\begin{array}{l} \varepsilon OC_{\alpha,r} \\ \varepsilon IC_{\alpha,r} \end{array}\right\} \subset D$ restricted to row r

$SF_\alpha \quad : R \times I \times D \rightarrow F \cup VF$

$\Phi_\alpha \quad \subseteq \cup_{r \in \{\theta\} \cup R} (\varepsilon OC_{\alpha,r} \times \varepsilon IC_{\alpha,r'})$

$Types \quad D \cup \{\text{SET OF } d : d \in D\}$

$S_\alpha \quad : D \rightarrow \cup_{n \in \aleph} Types_\alpha{}^n$

Toward Standardization

- ISO TC184 SC5 WG1 and CEN TC310 WG1
  - IS 14258, IS 15704, FDIS 19439
- United States government
  - Federal Enterprise Architecture Framework
  - Enterprise Architecture Management Maturity Framework
- The Open Group Architecture Framework
- Academic & Commercial
  - PERA, GERAM, ARIS, Metis, ZIFA…