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

• 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

- generic level
- partial level
- particular level

enterprise model phase

- domain identification
- concept definition
- requirements definition
- design specification
- implementation description
- domain operation
- decommission definition

enterprise modelling view

Landscape
Sources
Principles
Formalization

Architectural Principles for Enterprise Frameworks
© Copyright 2004 All Rights Reserved R. Martin, E. Robertson, J. Springer
C4ISR Version 2.0

Architectural Views

Operational View
- Identifies Warfighter Relationships and Information Needs

Systems View
- Relates Capabilities and Characteristics to Operational Requirements

Technical View
- Prescribes Standards and Conventions

Figure 2-2. Fundamental Linkages Among the Views

# Zachman Framework for Enterprise Architecture

**ENTERPRISE ARCHITECTURE - A FRAMEWORK™**

<table>
<thead>
<tr>
<th>Roles</th>
<th>Scope (Contextual)</th>
<th>Planner</th>
<th>Owner</th>
<th>Designer</th>
<th>Builder</th>
<th>Sub-Contractor</th>
<th>Functioning Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SCOPE (CONTEXTUAL)</td>
<td>Planner</td>
<td>Owner</td>
<td>Designer</td>
<td>Builder</td>
<td>Sub-Contractor</td>
<td>Functioning Enterprise</td>
</tr>
<tr>
<td></td>
<td>ENTERPRISE MODEL (CONCEPTUAL)</td>
<td>Owner</td>
<td>Owner</td>
<td>Designer</td>
<td>Builder</td>
<td>Sub-Contractor</td>
<td>Functioning Enterprise</td>
</tr>
<tr>
<td></td>
<td>SYSTEM MODEL (LOGICAL)</td>
<td>Designer</td>
<td>Designer</td>
<td>Designer</td>
<td>Builder</td>
<td>Sub-Contractor</td>
<td>Functioning Enterprise</td>
</tr>
<tr>
<td></td>
<td>TECHNOLOGY MODEL (PHYSICAL)</td>
<td>Builder</td>
<td>Builder</td>
<td>Builder</td>
<td>Builder</td>
<td>Sub-Contractor</td>
<td>Functioning Enterprise</td>
</tr>
<tr>
<td></td>
<td>DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)</td>
<td>Sub-Contractor</td>
<td>Sub-Contractor</td>
<td>Sub-Contractor</td>
<td>Builder</td>
<td>Sub-Contractor</td>
<td>Functioning Enterprise</td>
</tr>
<tr>
<td></td>
<td>FUNCTIONING ENTERPRISE</td>
<td>Functioning Enterprise</td>
<td>Functioning Enterprise</td>
<td>Functioning Enterprise</td>
<td>Functioning Enterprise</td>
<td>Functioning Enterprise</td>
<td>Functioning Enterprise</td>
</tr>
</tbody>
</table>

**Interrogatives**

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

**Landscape**

- Sources
- Principles
- Formalization

**Zachman Institute for Framework Advancement** - (810) 231-0531

**Copyright - John A. Zachman, Zachman International**
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

Concept space
- meta-model
- Model space
  - model / meta-data
  - Data space
  - data/instance
- World space
  - real world thing

Landscape
Sources
Principles
Formalization

EntityRelationship
Did Pid 47
95
Employee
IsA
Did DName
35
Accounting
95
Manufacturing

Department
Did Pid 47
35
Did 95
Party

Did PName
47
Joe

Manufacturing Dept.
Joe
Distinguish structure from connectivity

Landscape
Sources
Principles
Formalization

NY Div.

Sales
Mrktng.
Inside
Outside

Procur.
Cntrcts
Buyer

Manuf.
Line 01

Wrhse.
Shpng
Rcving

Pkgng
Logistic

Cell A

Cell B

Cell C
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

<table>
<thead>
<tr>
<th>abstract</th>
<th>relation</th>
<th>entity</th>
<th>subtype</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Concept space

meta-model

Model space

meta-data

Data space

data/instance

World space

real world thing

<table>
<thead>
<tr>
<th>abstract</th>
<th>Did</th>
<th>DName</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>35</td>
<td>Accounting</td>
</tr>
<tr>
<td></td>
<td>95</td>
<td>Manufacturing</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Did</th>
<th>Pid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>95</td>
<td>47</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Pid</th>
<th>PName</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>47</td>
<td>Joe</td>
</tr>
</tbody>
</table>

Manufacturing Dept. Joe
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_α 〈IC_α, OC_α, SF_α, Φ_α〉

leaf frames:
F_α 〈IC_α, OC_α, S_α〉

where

IC_α ⊆ D
OC_α ⊆ D
εOC_α,r ⊆ D restricted to row r
εIC_α,r ⊆ R × I × D → F ∪ VF
SF_α : R × I × D → F ∪ VF
Φ_α ⊆ ∪_{r ∈ {θ}}∪_{R(εOC_α,r × εIC_α,r')}
Types D ∪ \{SET OF d : d ∈ D\}
S_α : D → ∪_{n ∈ ℕ} Types_α^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...