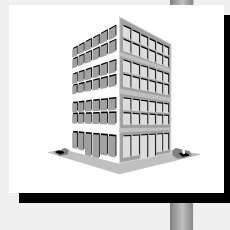


**Enterprise Architecture**

**Managing  
Complexity and Change**



**John A. Zachman**  
**Zachman International**  
**2222 Foothill Blvd. Suite 337**  
**La Canada, Ca. 91011**  
**1-818-244-3763**

© 2006 John A. Zachman, Zachman International

## **Enterprise Physics 101**

My quote to a General Management audience several years ago:

*"This seminar is NOT about increasing the stock price by the close of market, Friday afternoon.*

*"It IS about the laws of nature that determine the success of an Enterprise ... particularly, continuing success in the turbulent times of the Information Age.*

*"It is a presentation on Physics ...*

*"Enterprise Physics"*

© 1998 John A. Zachman, Zachman International

## Introduction

Enterprise Architecture presently appears to be a grossly misunderstood concept among management.

It is NOT an Information Technology issue.  
**It is an ENTERPRISE issue.**

It is likely perceived to be an Information Technology issue as opposed to a Management issue for two likely reasons:

- A. Awareness of it tends to surface in the Enterprise through the Information Systems community.
- B. Information Technology people seem to have the skills to do Enterprise Architecture if any Enterprise Architecture is being or is to be done.

## Origins of Ent. Arch.

Frederick Taylor "Principles of Scientific Management" 1911

Walter A. Shewhart "The Economic Control of Quality of Manufactured Product" 1931

Peter Drucker "The Practice of Management" 1954

Jay Forrester "Industrial Dynamics" 1961

Peter Senge "The Fifth Discipline" 1990

Eric Helfert "Techniques of Financial Analysis" 1962

Robert Anthony "Planning and Control Systems: A Framework for Analysis" 1965

Sherman Blumenthal "Management Information Systems: A Framework for Planning and Development" 1969

Alvin Toffler "Future Shock" 1970

George Steiner "Comprehensive Managerial Planning" 1972  
Etc., etc., etc.

## "Enterprise"

There are two implications to the word "Enterprise":

### I. Scope

The broadest possible boundary of the Enterprise.  
The Enterprise in its entirety.  
Enterprise-wide in scope.  
The whole thing.

### II. Content

ENTERPRISE Architecture is for ENTERPRISES.  
Enterprise Architecture has nothing to do with the Enterprise's systems or its information technology (except as they may constitute Row 4 constraints).  
The end object is to engineer and manufacture the ENTERPRISE, NOT simply to build and run systems.

"ENTERPRISE" ACTUALLY MEANS "ENTERPRISE"

## The Information Age

"The next information revolution is well underway. But it is not happening where information scientists, information executives, and the information industry in general are looking for it. It is not a revolution in technology, machinery, techniques, software, or speed. It is a revolution in CONCEPTS."

*Peter Drucker. Forbes ASAP, August 24, 1998*

"Future Shock" (1970) - The rate of change.

"The Third Wave" (1980) - The structure of change.

"Powershift" (1990) - The culture of change.

*Alvin Toffler*

"We are living in an extraordinary moment in history. Historians will look back on our times, the 40-year time span between 1980 and 2020, and classify it among the handful of historic moments when humans reorganized their entire civilization around a new tool, a new idea."

*Peter Leyden. Minneapolis Star Tribune. June 4, 1995*

*"On the Edge of the Digital Age: The Historic Moment"*

## The Challenge

What is your strategy for addressing:

Orders of magnitude increases in complexity,  
and  
Orders of magnitude increases in the rate of change?

Seven thousand years of history would suggest the only known strategy for addressing complexity and change is

**ARCHITECTURE.**

If it gets so complex you can't remember how it works,  
you have to write it down ... Architecture.

If you want to change how it works, you start with what  
you have written down ... Architecture.

The key to complexity and change: Architecture.

The question is: What is "Architecture,"  
Enterprise Architecture?

## Agenda

- I. Introduction
- II. Background - Enterprise Architecture: Why Is It Important
- III. Definitions - What does It Look Like?
- IV. The Framework and Management Issues
- V. The Framework and Architects
- VI. Conclusions

## Introduction to Enterprise Architecture

# The Framework for Enterprise Architecture



## Different Facets

Column 1 is descriptive of **WHAT** Inventories the Enterprise cares enough about to manage, that is, the countable things over which they maintain inventory control.

Column 2 is descriptive of **HOW** the Enterprise functions in Transforming (processing) raw materials and energy into finished goods and services.

Column 3 is descriptive of **WHERE** the Enterprise operates, the Locations from and to which various things are stored and transported, the Networks of the Enterprise.

Column 4 is descriptive of **WHO**, the Organizations of the Enterprise, the Roles to whom various work product responsibilities are allocated.

Column 5 is descriptive of **WHEN** things happen, the Timing cycles of the Enterprise

Column 6 is descriptive of **WHY**, the Motivation, the intent, the Ends of the Enterprise.

AUDIENCE PERSPECTIVES	OPERATIONS	COMPONENT	TECH-NOLOGY	SYSTEM	BUSINESS	SCOPE	AUDIENCE PERSPECTIVE
INVENTORY	<b>WHAT</b>	-	<b>Inventory</b>	-	<b>Things</b>		WHAT
FUNCTION	<b>HOW</b>	-	<b>Process</b>	-	<b>Transforms</b>		HOW
NETWORKS	<b>WHERE</b>	-	<b>Network</b>	-	<b>Locations</b>		WHERE
ORGANIZATION	<b>WHO</b>	-	<b>Organization</b>	-	<b>Roles</b>		WHO
TIMING	<b>WHEN</b>	-	<b>Timing</b>	-	<b>Cycles</b>		WHEN
MOTIVATION	<b>WHY</b>	-	<b>Motivation</b>	-	<b>Ends</b>		WHY
TARGET DOMAINS	WORKERS	IMPLE-MENTERS	ENGINEERS	ARCHITECTS	EXECUTIVE LEADERS	VISIONARIES	INTERROGATIVE PERSPECTIVE

## Different Perspectives

Row 1 is comprised of the Visionaries' Lists that identify Enterprise SCOPE boundaries.

Row 2 is comprised of the Executive Leaders' semantic models that define Enterprise Business concepts.

Row 3 is comprised of the Architects' schematic models that represent the Enterprise System logic.

Row 4 is comprised of the Engineers' blueprint models that specify Enterprise Technology Constructs.

Row 5 is comprised of the Implementers' listings that configure Enterprise Component instructions.

Row 6 is the Workers' instances that instantiate Enterprise Operations reality.

# A FRAMEWORK FOR ENTERPRISE ARCHITECTURE™

	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION	
<b>SCOPE (CONTEXTUAL)</b>	List of Things Important to the Business	List of Processes that the Business Performs	List of Locations Where the Business Operates	List of Organizations Important to the Business	List of Events/Grants/Significant Dates to the Business	List of Business Goals/Strategies	<b>SCOPE (CONTEXTUAL)</b>
<b>Planner (CONCEPTUAL)</b>	Entity - Class of Objects e.g. Semantic Model	Process - Class of Business Processes e.g. Business Process Model	Node - Major Business System e.g. Business System	People - Major Organization Unit e.g. Work Force Model	Time - Major Business Event/Cycle e.g. Business Calendar	Ends/Means - High-Level Business Goals/Strategy e.g. Business Plan	<b>Planner (CONCEPTUAL)</b>
<b>Owner (LOGICAL)</b>	Ent. = Business Entity Rel. = Business Relationship e.g. System Model	Proc = Business Process IO = Business Information e.g. System Architecture	Node = Business Location Link = Business Usage e.g. System Architecture	People = Organization Unit Type = Work Product e.g. System Architecture	Time = Business Event Cycle = Business Cycle e.g. System Model	Ent = Business Objective Means = Business Strategy e.g. Business Plan	<b>Owner (LOGICAL)</b>
<b>Designer (PHYSICAL)</b>	Ent. = Data Entry Rel. = Data Relationship e.g. Physical Data Model	Proc = Application Function IO = User Views e.g. Program	Node = IS Function Predecessor/Successor (etc.) Link = Data Specifications e.g. Technology Architecture	People = Role Work = Deliverable e.g. System Architecture	Time = System Event Cycle = Processing Cycle e.g. System Model	Ent = Structural Assertion Means = Action e.g. Data Design	<b>Designer (PHYSICAL)</b>
<b>DETAILED REPRESENTATIONS (OUT OF CONTEXT)</b>	Ent. = Table Segment, etc. Rel. = Key/Primary, etc. e.g. Data Definition	Proc = Computer Function IO = Data Elements/Sets e.g. Program	Node = Hardware Systems Link = Data Specifications e.g. Network Architecture	People = User Work = Screen Format e.g. System Architecture	Time = Enclose Cycle = Component Cycle e.g. Timing Diagram	Ent = Condition Means = Action e.g. Flow Specification	<b>DETAILED REPRESENTATIONS (OUT OF CONTEXT)</b>
<b>Sub-Constructor</b>	Ent. = Field Rel. = Address e.g. DATA	Proc = Language Statement IO = Control Block e.g. FUNCTION	Node = Address Link = Protocol e.g. NETWORK	People = Identity Work = Job e.g. ORGANIZATION	Time = Interval Cycle = Machine Cycle e.g. SCHEDULE	Ent = Sub-connection Means = Step e.g. STRATEGY	<b>Sub-Constructor</b>
<b>FUNCTIONING ENTERPRISE</b>							<b>FUNCTIONING ENTERPRISE</b>

©1987 John A. Zachman, Zachman International  
 For downloadable, color version of 2005, standard Enterprise Framework graphic see Framework Standards at [www.zachmaninternational.com](http://www.zachmaninternational.com)

	Audiences	Artifact Types	Process	Perspectives	Intent
<b>Visionaries'</b>	Lists	Identify	Scope	Boundaries	
<b>Executive Business Leaders'</b>	Semantic Models	Define	Business	Concepts	
<b>Architects'</b>	Schematic Models	Represent	System	Logic	
<b>Engineers'</b>	Blueprint Models	Specify	Technology	Constructs	
<b>Implementers'</b>	Listings	Configure	Component	Instructions	
<b>Workers'</b>	Instances	Instantiate	Operations	Reality	

## Caution

After three DAYS of intense instruction on the Framework, a recently retired Manufacturing Engineer from Ford observed, regarding the Framework and its implications relative to engineering and manufacturing Enterprises ... :

"Well, it's simple ... deceptively simple!  
Three days is merely 'scratching the surface'."

## Another Caution

I hesitate to show you the next foil ...

for two reasons:

1. Some labels I put on some of the Cells may not suggest "primitive" models to some people.
2. I do not want to encourage anyone to change any of the words on the Framework graphic. I put some different labels on the Cells but I did not alter the classification schema nor did I modify the meta-models of any of the Cells.

The labels are meant to "elaborate" the idea of the Framework for a particular audience as a communications device. The labels are NOT the Framework. The Framework is actually the classification schema as expressed by the metamodel of the set of Framework Cells.

# Enterprise Architecture

# The Framework and Management



## The Framework for Enterprise Architecture - *General Management Elaboration*

### CLASSES OF DESCRIPTIVE REPRESENTATIONS

	WHAT	HOW	WHERE	WHO	WHEN	WHY	SCOPE
VISIONARIES	IMPORTANT RESOURCES LIST	IMPORTANT TRANSFORMATIONS LIST	IMPORTANT NETWORKS LIST	IMPORTANT ORGANIZATIONS LIST	IMPORTANT TIMINGS LIST	IMPORTANT MOTIVATIONS LIST	
LIVE PERSONS	BUSINESS INVENTORY MODEL	BUSINESS PROCESS YIELD MODEL	BUSINESS LOGISTICS CAPACITY MODEL	BUSINESS WORK FLOW MODEL	BUSINESS CYCLES TIMING MODEL	BUSINESS OBJECTIVES MODEL	BUSINESS
SYSTEMS	BUSINESS INFORMATION MANAGEMENT FILING SYSTEM	BUSINESS FUNCTIONALITY SCHEMATIC	BUSINESS SYSTEMS SCHEMATIC	BUSINESS ROLES AND RESPONSIBILITIES	BUSINESS SYSTEM PROCESSING CYCLES	BUSINESS RULE SYSTEM LOGIC	SYSTEM
OBJECTS	FILING CONTAINER STORAGE PLACEMENT	SYSTEMS DESIGN (MANUAL OR AUTOMATED)	LOCATION CAPACITY BLUEPRINT	WORK PRODUCT DESIGN	BUSINESS SYSTEM TIMING DESIGN	BUSINESS RULE SYSTEM DESIGN	TECHNOLOGY
DIFFERS	IDENTIFICATION LISTINGS FOR PEOPLE OR MACHINES	SPECIFIC INSTRUCTIONS FOR PEOPLE OR MACHINES	LOCATION ADDRESSES FOR PEOPLE OR MACHINES	INDIVIDUAL WORK ASSIGNMENTS FOR PEOPLE OR MACHINES	TIMING SPECIFICATIONS FOR PEOPLE OR MACHINES	RULE SPECIFICATIONS FOR PEOPLE OR MACHINES	COMPONENT
WORKERS	RESOURCES	TRANSFORMATIONS	NETWORKS	ORGANIZATIONS	TIMINGS	MOTIVATIONS	OPERATIONS

### CLASSES OF ENTERPRISE ENGINEERING DESIGN ARTIFACTS

# THE ENTERPRISE

The domain of MANAGEMENT (in general)      © 2005 John A. Zachman, Zachman International

The domain of Systems (Information Systems or Manual Systems) (in general)

The domain of various Staff Organizations (in general)

Figure 1. The Framework for Enterprise Architecture (Business Vernacular Elaboration)

See CAUTION on previous foil.

## The Framework and Management

### Management reasons for the **Columns**:

Column 1 has to do with inventory management.

Column 2 has to do with yield on transformations.

Column 3 has to do with capacity management.

Column 4 has to do with performance management.

Column 5 has to do with response times.

Column 6 has to do with plans and controls.

## The Framework and Management

### Management reasons for the Rows:

Row 1 has to do with setting Enterprise boundaries.

Row 2 has to do with defining Business Policies.

Row 3 has to do with institutionalizing the Business Policies (systematization).

Row 4 has to do with implementations (manual and/or automated).

Row 5 has to do with specific instructions (for people and/or machines).

Row 6 has to do with Enterprise operations.

## The Framework and Management

Oversimplifications and generalizations, however ...

Some reasons why the record keeping system does not:

- accurately reflect the actual inventories of resources or
- accurately reflect the financial characteristics of the Enterprise or
- provide consistent or accurate regulatory compliance (Sarbanes-Oxley type) information

are likely because:

1. The business policies that govern inventory management are not defined or not defined and managed consistently across the scope of the Enterprise. (Col.1, R 2)
2. The record-keeping system(s) (Col.1, Row 3) do not accurately and consistently reflect the business inventory policies. (Col. 1, Row 2)
3. The transaction data about resource acquisition and consumption is not accurately and "primitively" recorded. (Col. 2, Row 6; Col. 1, Row 6)

## The Framework and Management

Oversimplifications and generalizations, however ...

One reason why G & A and indirect expenses increase is likely because:

1. Compensation for inconsistencies in business policies regarding inventory management (Col. 1, Row 2) or in the filing system that records actual inventories. (Col. 1, Row 3.) (This is entropy ... compensation for disorder in the system.)

One reason why the yield on the business transformation of raw materials and energy to finished goods deteriorates over time is likely because:

1. The Business Processes (Col. 2, Row 2) and their supporting systems (Col. 2, Row 3) evolve (like the Winchester House evolved) ... they are not being engineered and have never been optimized.

## The Framework and Management

Oversimplifications and generalizations, however ...

Some reasons why the distribution costs increase and network reliability decreases are likely because of:

1. Suboptimal positioning of storage and transmission capacities (Col. 3, Row 2)
2. Lack of standardization of locations and connections requiring "interfacing" or "translations." (Column 3, Rows 2 - 5)

Some reasons personnel costs increase are likely because:

1. Work product assignments are complex and overlapping and not clearly specified (Col. 4, Row 2)
2. Skills are not well-matched with the work assignments. (Col. 4, Row 2)

One reason why cycle times are excessive and difficult to predict is likely because:

1. Multiple cycles are interdependent and are interacting capriciously. (The law of unintended consequences.) (Col. 5, Rows 2 - 5)

© 2005 John A. Zachman, Zachman International

## The Framework and Management

Oversimplifications and generalizations, however ...

One reason why business objectives are difficult to realize is likely because:

1. Objectives are not defined such they change the state of some specific (single) thing that is within the Enterprise's control to change (Col. 6 and some Entity in Row 2). (Plans not attainable and/or not measurable.)

Some reasons why the Enterprise is not flexible are likely because:

1. No engineering has been done to separate the things that change independently from one another. (No Framework primitives.)
2. It's hard to figure out what to change ... or what changes will actually work. (No explicit Architecture.)

One reason why it is so difficult to take out costs is likely because:

1. There is no way to find recurring concepts that should or must be reused (i.e. no reusable primitive components.) (No Classification, i.e. no Framework.)

© 2005 John A. Zachman, Zachman International

## The Framework and Management

Oversimplifications and generalizations, however ...

The reason why there is:

NO Integration - can't find reusable components

NO Interoperability - no standardization (of contents  
or containers)

NO Alignment - nothing explicit to which to align

NO Security - nothing to examine before implementation

NO Reduced time to market for systems implementations  
- nothing in inventory before you get the order

NO Flexibility - no separation of independent variables

NO Predictability - no knowledgebase

Etc., etc., etc.

In short: There is NO ARCHITECTURE.

## The Framework and Management

And ... life (business) is getting more complex ...

And ... the rate of change continues to increase ...

And ... once again, I submit ...

... Someday you are going to wish  
you had all these models  
(that is, Enterprise Architecture) ...

... and when that day comes,  
it's going to be too late to build them!

# Primitive Models

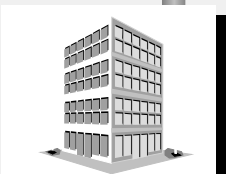
A "Primitive" Model is one that is comprised of elements from a single Framework Cell ... one single "abstraction" from one single "perspective."

	What	How	Where	Who	When	Why
Scope						
Business	<b>Primitive Models</b>					<i>Executive Leaders</i>
System						<i>Architects</i>
Technology		■				<i>Engineers</i>
Components					■	<i>Implementers</i>
	<i>Resources</i>	<i>Functions</i>	<i>Networks</i>	<i>Organizations</i>	<i>Timings</i>	<i>Motivations</i>

© 1999 John A. Zachman, Zachman International

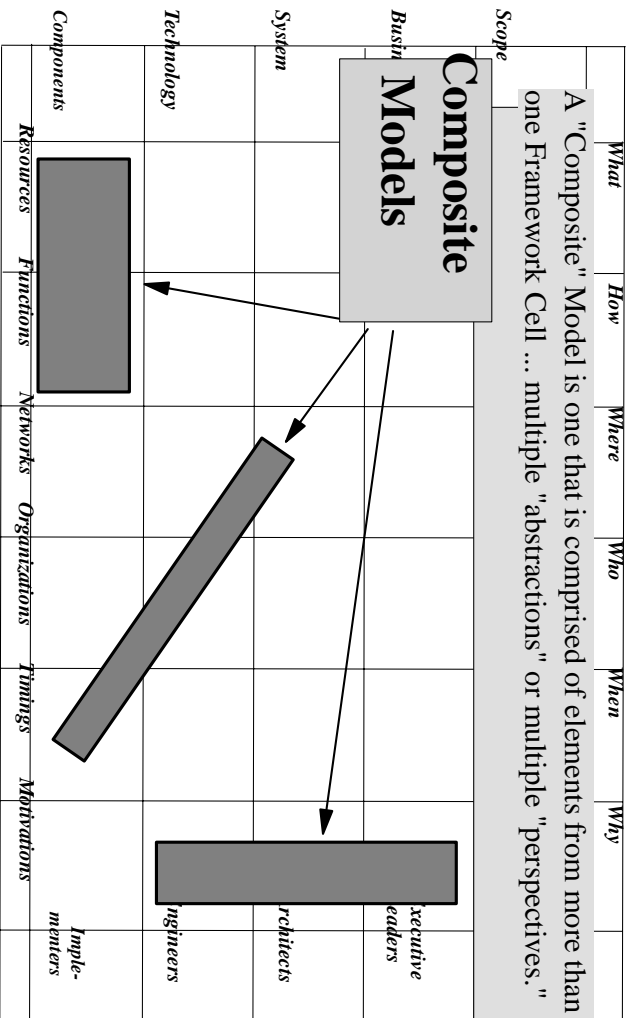
Enterprise Architecture

Framework and Architects



# Composite Models

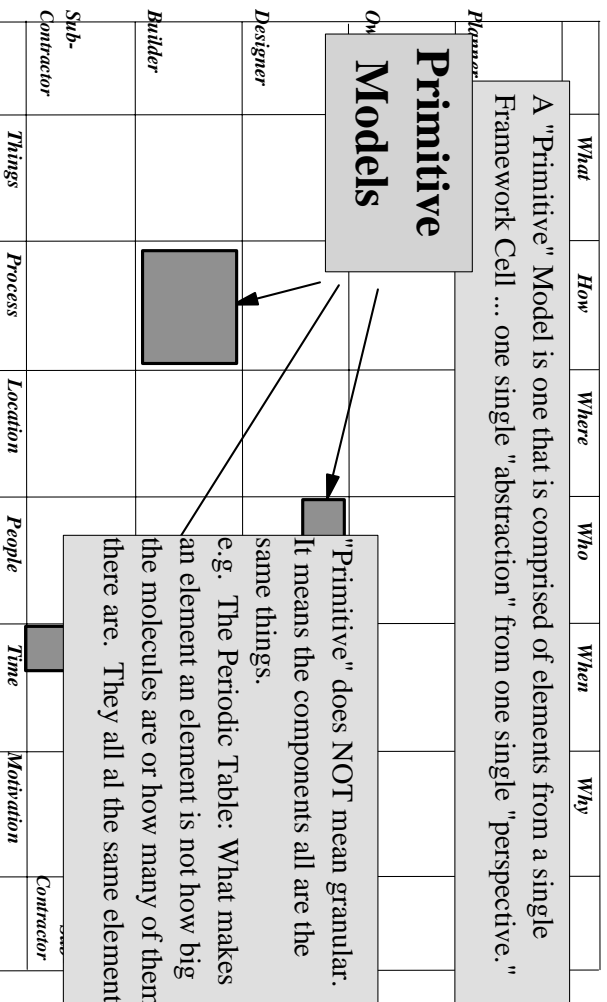
A "Composite" Model is one that is comprised of elements from more than one Framework Cell ... multiple "abstractions" or multiple "perspectives."



© 1999 John A. Zachman, Zachman International

# Primitive Models

A "Primitive" Model is one that is comprised of elements from a single Framework Cell ... one single "abstraction" from one single "perspective."

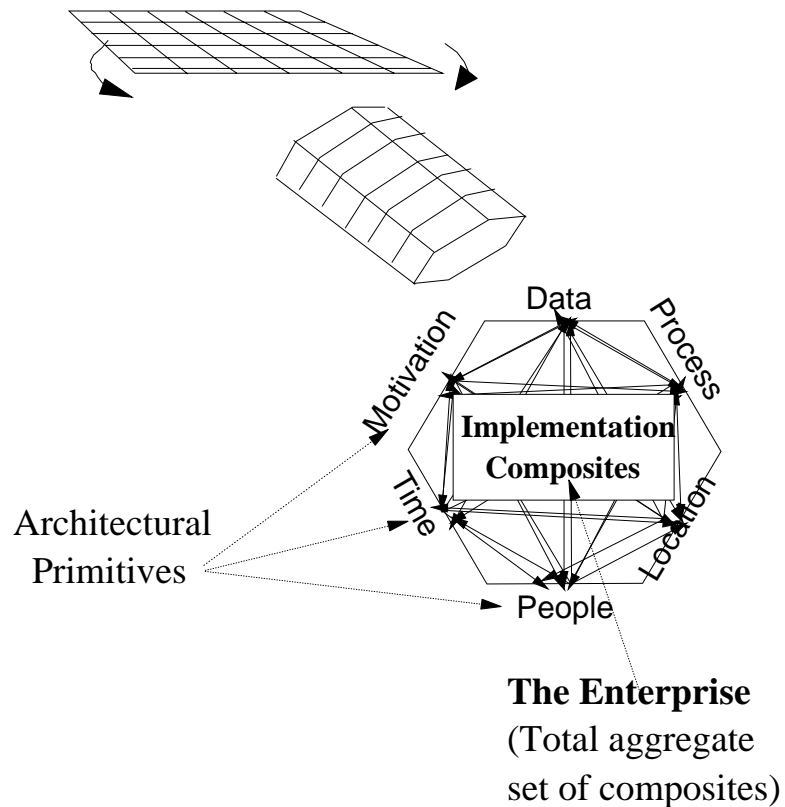


© 2000 - 2006 John A. Zachman, Zachman International



## Architecture vs Implementation

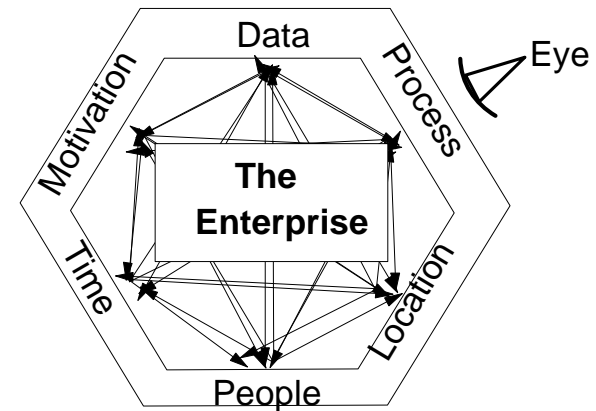
From a fixed set of 36 Architectural Primitives, you could create a virtually infinite set of Implementation Composites.



© 1999 John A. Zachman, Zachman International

## The Enterprise Hologram

If the Composites are not hard bound together the Enterprise is virtual, like a hologram. It could be viewed from any facet.



You can view the Enterprise through any facet of the hologram and see all the other facets relative to the viewing facet.

© 2006 John A. Zachman, Zachman International

## Building Legacy

If you have no Primitives and you base your implementation Composites on a single Enterprise-wide facet, you are going to optimize that facet and sub-optimize all the others, that is, you will fix one problem and cause at least five others. You are just building more legacy. It doesn't make any difference which facet you pick.

If the only tool you have is a hammer, all the world looks like a nail.

If the only tool you have is Process,  
all the Enterprise looks like Inputs/Outputs to Process.

If the only tool you have is Data,  
all the Enterprise looks like attributes of an object.

If the only tool you have is Technology,  
all the Enterprise looks like an application.

If the only tool you have is People,  
all the Enterprise looks like Services/Web Pages.

If the only tool you have is Time,  
all the Enterprise looks like Systems Thinking (Dynamics).

If the only tool you have is Motivation,  
all the Enterprise looks like Constraints/Business Rules.

© 2006 John A. Zachman, Zachman International

## Architecture vs Implementation

If you are not building "primitive models," you are not doing Architecture, you are doing implementation.

Composite models are implementations.

Primitive models are Architecture.

Composite models should be created from primitive models.

If composite models are being created and no primitive models exist, then the composite model is likely being defined relative to a specific implementation (one component of one facet), not relative to *the Enterprise*. You are optimizing the *implementation* and SUB-OPTIMIZING the ENTERPRISE. It is a point-in-time solution. It is good only as long as nothing changes. The likelihood of it being reusable is low to zero. It is more "legacy."

### The "Silver Bullet"

Building implementations (composite models) and SAYING you are doing Enterprise Architecture (primitive models) is the worst possible strategy.

2000 John A. Zachman, Zachman International

## The Framework Is a Schema

The Framework is a two-dimensional classification system for ENTERPRISE descriptive representations NOT I/S.

The classification scheme for each axis grew up quite independently from the Framework application.

The classification for each axis is:

- a. Comprehensive
- b. Non-redundant

Therefore, each cell of the Framework is:

- a. Unique
  - b. Primitive (one single Abstraction by one single Perspective)
- and the total set of cells is complete.

The Framework logic is universal, independent of its application - totally neutral relative to methods/tools.

**The Framework is a "normalized" schema ...  
... NOT a matrix.**

That's what makes it a good analytical tool.

© 1999 John A. Zachman, Zachman International

## Lean and Mean

End Object: Minimum possible costs  
Minimum possible complexity

How do you do that?

Normalize EVERYTHING!

Remove ALL redundancy - NO recurring concepts

Redundancy:

1. Unnecessary costs of duplication - waste.
2. Compensatory costs of discontinuity - Entropy (Entropy = energy not available for productive work)
  - a. Internal costs - operating expenses
  - b. External costs - damage control, litigation

Second law of thermodynamics - the aging process. Over time, the energy required to support the system (entropy) increases. At the point in time the energy required to support the system exceeds the energy in the system, the system dies. How do you remove entropy? Re-engineer the system to remove disorder. Take out all discontinuous duplication. Engineer for simplicity.

© 2003 John A. Zachman, Zachman International

## Finding Redundancies

How do you discover recurring concepts?

How do you "normalize" anything? CLASSIFY.

But - the classification scheme has to be "clean." You can't have mixtures (apples and oranges) in any category because you won't be able to detect redundancies. The schema has to be "normalized" - one fact in one place.

And - the schema has to be comprehensive. You must have a category for every concept or you won't find the duplication of concepts that are not classified.

That is, the schema has to be comprised of single variable, "primitive" categories. No mixtures (composites.) The schema has to be complete, the total possible set of categories.

For example, the Periodic Table.

Anything less than the total set would either, by definition, be DE-normalized (contain composite categories) or could not accommodate the totality of the concepts.

© 2003 John A. Zachman, Zachman International

## The Framework

Primitive Models are architecture

Primitive models defined relative to the Enterprise are ENTERPRISE Architecture. Long term investments.

Composite Models are implementations

Composite models defined relative to one project are implementations. It is doubtful that you could define a composite, Enterprise-wide Model. It would be so complex, who could possibly understand it? Composite models are short term implementations.

**YOU DON'T HAVE TO NORMALIZE ALL 30  
PRIMITIVE MODELS TO REALIZE SHORT TERM  
OPTIMIZATION BENEFITS!**

(Note: discontinuity in some Columns may directly, negatively impact management's performance.)

**POINT NO. 2**

If you retain and maintain the primitive models, they are the baseline for managing change.

© 2003 John A. Zachman, Zachman International

# The Future

- A. Build Primitive Models
- B. Store Primitive Models
- C. Manage (Enforce) Primitive Models
- D. Change Primitive Models
- E. Assemble Composite Models  
from Primitive Models

It is not adequate merely to produce running code.  
(That was an Industrial Age idea.)

The long term Enterprise value  
lies in Enterprise "Engineering,"  
i.e. in the MODELS THEMSELVES!  
The "Knowledgebase" of the Enterprise  
(This is an Information Age idea!)

## END STATE VISION

	What	How	Where	Who	When	Why	
Scope							Attributes
Business							Primitive Elements
System							Concepts
Technology							Interfaces
Components							Implementations
	Resources	Functions	Networks	Organizations	Timings	Motivations	Implementations

Some day  
You are going to wish you had  
all these models made explicit,  
Enterprise-wide,  
horizontally and vertically integrated,  
at excruciating level of detail !!!

## Enterprise Architecture

## Conclusions



## 1965 Systems Problems

1. Didn't meet Requirements. (not "aligned")
2. The data was no good:
  - Not consistent from system to system.
  - Not accurate.
  - Not accessible.
  - Too late.
3. Couldn't change the system. (Inflexible)
4. Couldn't change the technology. (Not adaptable)
5. Couldn't change the business. (Couldn't change the system or the technology so couldn't change business.)
6. Little new development (80% \$ for maintenance)
7. Took too long.
8. Cost too much.
9. Always over budget.
10. Always missed schedules.
11. DP budget out of control.
12. Too complicated - can't understand it, can't manage it.
13. Just frustrating.

*(Adapted from Doug Erickson)*

## 2006 Systems Problems

1. Don't meet Requirements. (not "aligned")
2. The data is no good:
  - Not consistent from system to system.
  - Not accurate.
  - Not accessible.
  - Too late.
3. Can't change the system. (Inflexible)
4. Can't change the technology. (Not adaptable)
5. Can't change the business. (Can't change the system or the technology so can't change business.)
6. Little new development (80% \$ for maintenance)
7. Takes too long.
8. Costs too much.
9. Always over budget.
10. Always missed schedules.
11. IT budget out of control.
12. Too complicated - can't understand it, can't manage it.
13. Just frustrating.

*(Adapted from Doug Erickson)*

## It's Funny ...

COBOL didn't fix those problems!

MVS didn't fix those problems!

Virtual Memory didn't fix those problems!

IMS, DB2, Oracle, Sybase, Access, Fortran, PL/1, ADA, C++, Visual Basic, JAVA 2, 360's, 390's, MPP's, DEC VAX's, H200's, Crays, PC's, MAC's, Distributed Processing, didn't fix those problems!

Word, Excel, Powerpoint, Outlook Express, eMAIL, DOS, Windows 95, 98, 2000, NT, ME, XP, Unix, Linux, Object Oriented, COM, DCOM, CORBA, EDI, HTML, XML, UML, the Internet, B2B, B2C, Portals, Browsers didn't fix those problems!

IEF, IEW, ADW, ERWIN, POPKIN, Rational, PTECH, Rochade, Platinum, Design Bank, Data Warehouse, SAP, Baan, Peoplesoft, Oracle Financials, BSP, ISP, EAP, EAI didn't fix those problems!

And, I doubt that Web Services, .Net, Websphere, Extreme Programming, Service Oriented Architecture or Component Development (whatever that is) is going to fix the problems.

**IT MAKES ONE WONDER IF THERE ACTUALLY IS A TECHNICAL SOLUTION TO THE PROBLEM!!!**

# Engineering Problem

I'm not saying that there is anything wrong with any of these technologies.

In fact, any or all of them may well be very good ...

In fact, you may not be able to solve the Enterprise problem without employing some of these technologies.

However,

The Enterprise problem is an ENGINEERING problem,  
NOT a technical problem.

My perception is that it is going to take actual work, ENGINEERING work, to solve the problem. My plan would be to start building out models, PRIMITIVE models, engineering them for alignment, integration, flexibility, reduced time-to-market, etc., etc., etc.

What would be YOUR plan for solving the problems???