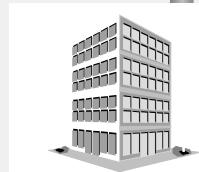


Framework for Enterprise Architecture

Enterprise Physics 101



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

© John A. Zachman, Zachman International

Preface

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.

© John A. Zachman, Zachman International

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?

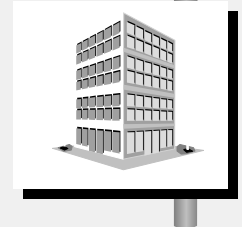
Agenda

Half Day Agenda

- I. Introduction to Enterprise Architecture
 - A. The Framework for Enterprise Architecture
 - B. Basic Enterprise Physics
- II. Enterprise Engineering Design Objectives
 - A. Alignment, Integration, Flexibility, etc.
 - B. Reducing Time-to-Market
- III. Value Proposition
 - A. Industrial Age
 - B. Information Age
- IV. Frequently Asked Questions
 - A. Cheaper and Faster
- V. Conclusions

Introduction to Enterprise Architecture

The Framework for Enterprise Architecture



Different Perspectives

Buildings	Airplanes	Enterprise
OWNER		
Architect's Drawings	Wk. Bk. Dwn. Structure	Model of Business
DESIGNER		
Architect's Plans	Engineering Design	Model of Info. Sys.
BUILDER		
Contractor's Plans	Mfg. Eng. Design	Technlgy Model

A Framework

	DATA	FUNCTION	NETWORK
SCOPE			
BUSINESS MODEL			
SYSTEM MODEL			
TECH MODEL			
DETAIL RPSNTNS			
SYSTEM			

Different Abstractions

WHAT	HOW	WHERE
------	-----	-------

Material	Function	Location
----------	----------	----------

Bill of Materials	Functional Specs	Drawings
-------------------	------------------	----------

Data Models	Functional Models	Network Models
-------------	-------------------	----------------

A Framework

	WHAT	HOW	WHERE
OWNER			
DESIGNER			
BUILDER			

A Framework

	WHAT	HOW	WHERE
SCOPE			
OWNER			
DESIGNER			
BUILDER			
OUT OF CONTEXT			
PRODUCT			




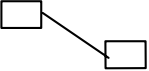
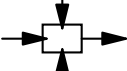
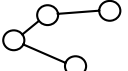
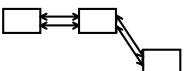
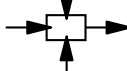
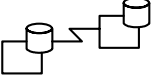
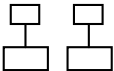

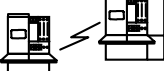



© John A. Zachman, Zachman International

A Framework

	DATA	FUNCTION	NETWORK
SCOPE			
BUSINESS MODEL			
SYSTEM MODEL			
TECH MODEL			
DETAIL RPSNTNS			
SYSTEM			

© John A. Zachman, Zachman International

ENTERPRISE ARCHITECTURE - A FRAMEWORK

	DATA <i>What</i>	FUNCTION <i>How</i>	NETWORK <i>Where</i>	
SCOPE (CONTEXTUAL)	List of Things Important to the business 	List of Processes the Business Performs 	List of Locations in which the Business Operates 	
<i>Planner</i>	ENTITY = Class of Business Thing	Process = Class of Business Process	Node = Major Business Location	SCOPE
BUSINESS MODEL (CONCEPTUAL)	e.g. Semantic Model 	e.g. Business Process Model 	e.g. Business Logistics System 	
<i>Owner</i>	Ent = Business Entity ReIn = Business Relationship	Proc = Bus Process I/O = Bus Resources	Node = Business Location Link = Business Linkage	OWNER
SYSTEM MODEL (LOGICAL)	e.g. Logical Data Model 	e.g. Application Architecture 	e.g. Distributed System Architecture 	
<i>Designer</i>	Ent = Data Entity ReIn = Data Relationship	Proc = Application Function I/O = User Views	Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	DESIGNER
TECHNOLOGY MODEL (PHYSICAL)	e.g. Physical Data Model 	e.g. System Design 	e.g. Technology Architecture 	
<i>Builder</i>	Ent = Segment/Table/etc. ReIn = Pointer/Key/etc.	Proc = Computer Function I/O = Data Elements/Sets	Node = Hardware/Systems Software Link = Line Specifications	BUILDER
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)	e.g. Data Definition 	e.g. Program 	e.g. Network Architecture 	
<i>Sub-Contractor</i>	Ent = Field ReIn = Address	Proc = Language Statement I/O = Control Block	Node = Address Link = Protocol	OUT OF CONTEXT
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	PRODUCT




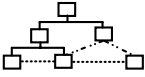
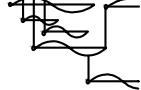
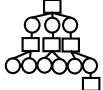
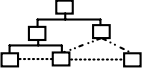
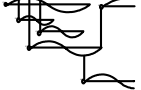
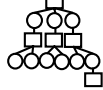
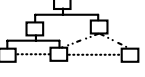
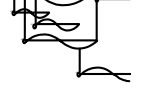
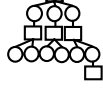



© John A. Zachman, Zachman International

A Framework

WHO	WHEN	WHY	
			SCOPE
			OWNER
			DESIGNER
			BUILDER
			OUT OF CONTEXT
			PRODUCT

© John A. Zachman, Zachman International

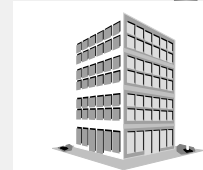
ENTERPRISE ARCHITECTURE THE "OTHER THREE COLUMNS"

PEOPLE	TIME	MOTIVATION	
List of Organizations Important to the Business  People = Major Organization Unit e.g., Work Flow Model	List of Events/Cycles Significant to the Business  Time = Major Business Event/Cycle e.g., Master Schedule	List of Business Goals/Strategies  Ends/Means = Major Business Goal/Strategy e.g., Business Plan	SCOPE (CONTEXTUAL) <i>Planner</i>
 People = Organization Unit Work = Work Product e.g., Human Interface Architecture*	 Time = Business Event Cycle = Business Cycle e.g., Processing Structure	 End = Business Objective Means = Business Strategy e.g., Business Rule Model	BUSINESS MODEL (CONCEPTUAL) <i>Owner</i>
 People = Role Work = Deliverable e.g., Presentation Architecture	 Time = System Event Cycle = Processing Cycle e.g., Control Structure	 End = Structural Assertion Means = Action Assertion e.g., Rule Design	SYSTEM MODEL (LOGICAL) <i>Designer</i>
 People = User Work = Screen Formats e.g., Security Architecture	 Time = Execute Cycle = Component Cycle e.g., Timing Definition	 End = Condition Means = Action e.g., Rule Specification	TECHNOLOGY MODEL (PHYSICAL) <i>Builder</i>
 People = Identity Work = Job e.g., ORGANIZATION	 Time = Interrupt Cycle = Machine Cycle e.g., SCHEDULE	 End = Sub-condition Means = Step e.g., STRATEGY	DETAILED REPRESENTATIONS (OUT-OF-CONTEXT) <i>Sub-Contractor</i>
			FUNCTIONING ENTERPRISE

© John A. Zachman, Zachman International

Enterprise Architecture

Basic "Physics"



© John A. Zachman, Zachman International

Less than Excruciating Detail

	DATA	FUNCTION	LOCATION	PEOPLE	TIME	MOTIVATION	SCOPE (CONTEXTUAL)
	<i>What</i>	<i>How</i>	<i>Where</i>	<i>Who</i>	<i>When</i>	<i>Why</i>	
SCOPE (CONTEXTUAL)							SCOPE (CONTEXTUAL)
<i>Planner</i>							<i>Planner</i> BUSINESS MODEL (CONCEPTUAL)
BUSINESS MODEL (CONCEPTUAL)							<i>Owner</i> BUSINESS MODEL (CONCEPTUAL)
<i>Owner</i>							SYSTEM MODEL (LOGICAL)
SYSTEM MODEL (LOGICAL)							<i>Designer</i> SYSTEM MODEL (LOGICAL)
<i>Designer</i>			Re: Any Cell				TECHNOLOGY MODEL (PHYSICAL)
TECHNOLOGY MODEL (PHYSICAL)							<i>Builder</i> TECHNOLOGY MODEL (PHYSICAL)
<i>Builder</i>							DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)							<i>Sub-Contractor</i> DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)
<i>Sub-Contractor</i>							FUNCTIONING SYSTEM
FUNCTIONING SYSTEM	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING SYSTEM

John A. Zachman, Zachman International

Less than Enterprise Scope

	DATA	FUNCTION	LOCATION	PEOPLE	TIME	MOTIVATION	SCOPE (CONTEXTUAL)
	<i>What</i>	<i>How</i>	<i>Where</i>	<i>Who</i>	<i>When</i>	<i>Why</i>	
SCOPE (CONTEXTUAL)							SCOPE (CONTEXTUAL)
<i>Planner</i>							<i>Planner</i> BUSINESS MODEL (CONCEPTUAL)
BUSINESS MODEL (CONCEPTUAL)							<i>Owner</i> BUSINESS MODEL (CONCEPTUAL)
<i>Owner</i>							SYSTEM MODEL (LOGICAL)
SYSTEM MODEL (LOGICAL)							<i>Designer</i> SYSTEM MODEL (LOGICAL)
<i>Designer</i>			Re: Any Cell				TECHNOLOGY MODEL (PHYSICAL)
TECHNOLOGY MODEL (PHYSICAL)							<i>Builder</i> TECHNOLOGY MODEL (PHYSICAL)
<i>Builder</i>							DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)
DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)							<i>Sub-Contractor</i> DETAILED REPRESENTATIONS (OUT-OF-CONTEXT)
<i>Sub-Contractor</i>							FUNCTIONING SYSTEM
FUNCTIONING SYSTEM	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING SYSTEM

John A. Zachman, Zachman International

Basic Physics

1. If the Enterprise exists, ALL of the descriptive representations (models) exist ... by definition.

If they are not explicit, they are implicit (that is, you are making assumptions about them.)

2. The system IS the Enterprise

Manual systems employ pencils, paper, file cabinets.
Automated systems employ stored programming devices and electronic media.

3. High level descriptions (models) are good for planning, scoping, bounding, segmenting.
(High level descriptions are NO good for implementation.)
4. Narrow-in-scope descriptions are quick.
(Narrow in scope descriptions result in "stove pipes.")

Implications of Basic Physics

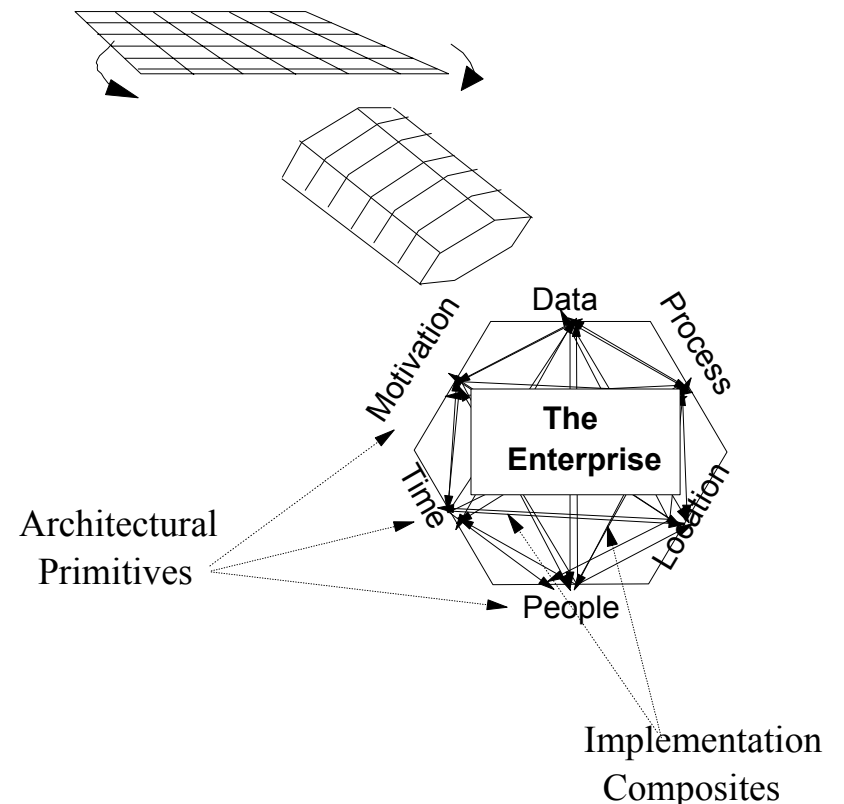
1. The governance system should define, for the next planning period, which Cells or slivers of Cells you intend to make explicit. Any Cell (or portion of Cell) you do not make explicit is where there is risk of defects or discontinuity, entropy (disorder, energy not available for work).
2. Do not lose sight of the fact that the end object is to produce a coherent, integrated ENTERPRISE, not simply to build and run systems. If you are simply building and running systems you are DIS-integrating, SUB-optimizing, DIS-ordering, DE-normalizing the Enterprise (See "Entropy" above).
3. High level descriptions (models) are good for planning, scoping, bounding, segmenting ... but not for implementing. If you do not define the excruciating level of detail, do you think it goes away?!! No. You are just making assumptions about it ... i.e. potential defects.

Implications of Basic Physics

4. You can compromise Enterprise-wide integrity of some Columns of models in the interest of reducing the time it takes for implementation with impunity ... it is only inefficient, not optimal. However, compromising Enterprise-wide integrity in Columns 1, 3 and 6 will directly, negatively impact management's performance. (See next Section: "Industrial Age Breakdown.")

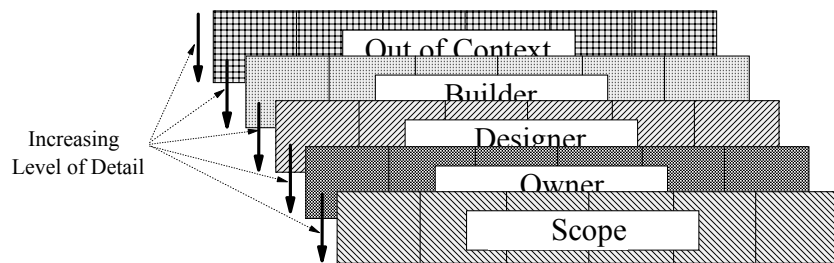
Architecture vs Implementation

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



Excruciating Level of Detail

Level of detail is a function of a Cell, NOT a Column.



The Framework Is a Schema

The Fmwrk 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. Primitive

Therefore, each cell of the Framework is:

- a. Unique
- b. Primitive

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.

Lean and Mean

End Object: Minimum possible costs
Minimum possible complexity

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.

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.

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 Columns 1, 3 and 6 will directly, negatively impact management's performance.)

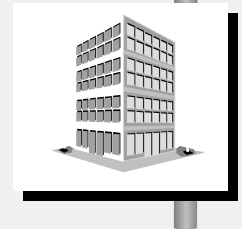
POINT NO. 2

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

© John A. Zachman, Zachman International

Frequently Asked Questions

It's Cheaper and Faster



© John A. Zachman, Zachman International

Cheaper and Faster

Using a top-down, Enterprise Architecture approach, a rigorous, enhanced Information Engineering Method, Three-Schema Data Architecture and CASE technology:

Cost per new entity type/RDBMS table
reduced from >\$150,000 to <\$10,000.

Enterprise data handling labor reduced 50%.

Reduced development time of 25% through
improved communication and conflict resolution.

Development time and cost reductions for every
succeeding implementation of >50%,
compounded, through reuse of database and
application components with no modifications.

Reduced disk space for data (including history)
of 20% - 80% through elimination of redundancy.

Reference: Doug Erickson 614-751-5078

Cheaper and Faster

State of Ohio: Workers Compensation Board

Rates System 1,030 entity types
(operational - 2 1/2 years elapsed time)

Benefits Payments 720 e/t's (Reused 470)
(operational)

Retro Rated Billing 230 e/t's (Reused 220)
(operational)

4 years total elapsed time, no database failures,
< 40 hours required maintenance

Health Provider Mgmt 415 e/t's (Reused 255)
(under development)

Total Cost per entity type \$25,000 (conservative)
includes legacy data cleansing
all data conversion costs
all interfaces with remaining legacy
no redundancy - reduced entropy
complete Enterprise alignment and integration

Total savings: 945 (reused) x \$25,000 = \$23,625,000

Cheaper and Faster

Recent Package implementation: \$50,000/ET (conserv.)
 (no data cleansing, no data conversions, no legacy
 interfaces, added redundancy and 60% functionality)

Recent Custom Apps: \$100,000- \$150,000/Ent. Type
 typical legacy, redundant environment (re: entropy)

Comparative Costs

Trad. Appln Devel 2395 e/t's x \$140,000 = \$335,300,000
 Package 2395 e/t's x \$50,000 = \$119,750,000
 Ent. Arch. 2395 - 945 e/t's x \$25,000 = \$36,000,000
 (and Enterprise Architecture approach
 "aligned," low maintenance, no entropy)

Re: "Reusable code"

In three operational Systems:

6,128 action blocks Avg. Reuse factor = 7.0
 (Trad. A/D: Code, test, maintain 42,896 subroutines)
 (attributable to granularity and precision of the data
 model, i.e. many processes use the same data.)

Reference: Doug Erickson 1-614-751-5078

Cheaper and Faster

State of Ohio

Different State

Workers Comp.	<i>Application</i>	Child Welfare
IEF/CoolGen	<i>Same CASE Tool</i>	IEF/CoolGen
Architected	<i>Different Methodology</i>	Classic
1030	<i>Entity Types</i>	300
2.5 Years	<i>Elapsed Time</i>	12 Years
-	<i>Development Costs</i>	\$42 Mil.
\$25,000	<i>Cost per Entity Type</i>	\$140,000
		(2 prime contractors and one local cntrtr. Estimating 3 more years to enhance/fix)
		Reference: Doug Erickson 614-751-5078

The REAL Benefit

From: Jim Tompkins (**Large Customer**)

To: Lauree Raica (**Chief Risk Officer**)

THANK YOU! THANK YOU!! THANK YOU!!!

This is great news for Frank Gates and The Service Assn. of Ohio. I appreciate so much your continuing efforts to help facilitate the improved turnaround time on these quarterly claim cost updates. I also want to express my appreciation to the "systems staff" at BWC in moving us forward with receiving the claims status and effective date information in the updated PIRS system. This will be a significant benefit.

From: Jim Romig (**Chief, Employer Relations**)

To: Admiral Jim Conrad (**CEO**)

Adm. Conrad, the IT Department did a great job in speeding up the turnaround time for quarter ending reserves being posted on Dolphin. What used to take 24 days now takes less than 10 . We have received several thank you's from TPAs since they are now able to get Sept. 30th data by Oct. 10th instead of Oct. 24th.

The REAL Benefit

From: Leo Genders (**Dpty. Mgr. Appln. Dvlpmnt.**)

To: Rates & Payments Project team

This is excellent news and worthy of high praise. Great job to all involved. It is not often that an outside customer praises the internal IT Team!

From: Kevin Ribble (**Appln. Dvlpmt. Mgr.**)

To: Nary Loganathan, Russel Marwitz (**Erickson Contractors**)

Great job on this! Not only have you improved service for our customers, but the significant decrease in processing time makes things much simpler for all of us in IT.

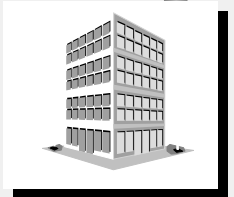
From: Nary Loganathan (**Erickson Contractor**)

To: Doug Erickson

And the icing on the cake, tabular reserves completed in 11 hours yesterday night (ran for ~ 4.5 days in previous quarters) ... minor read statement change.

Enterprise Architecture

Conclusions



Plan A (roughly)

SCORE (CONTEXTUAL)	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION	SCORE (CONTEXTUAL)
	<i>Major</i>	<i>Major</i>	<i>Major</i>	<i>Major</i>	<i>Major</i>	<i>Major</i>	
	Identifies important to the Business	Identifies the Business Performs	Identifies Where the Business Operates	Identifies Important to the Business	Identifies Significant to the Business	Identifies General Strategies	
<i>Planner</i>	Entity - Class eg. Semantic	Process - Class eg. Business F	Node - Major Business System eg. Business Logic	People - Major Organization Unit eg. Work Flow Model	Time - Major Business Cycle eg. Master Schedule	Establishes Business Strategy eg. Business Plan	<i>Planner</i> BUSINESS MODEL (CONCEPTUAL)
<i>Owner</i>	Entity - Business eg. Business Relationship	Process - Business eg. Business Process	Node - Business Location eg. Physical Location	People - Organization Unit eg. Human Architecture	Time - Business Event Cycle eg. Processing Structure	End - Business Objective eg. Business Plan Node	<i>Owner</i> SYSTEM MODEL (LOGICAL)
<i>Designer</i>	Entity - Data Entity eg. Data Relationship	Process - Function eg. System Order	Node - IS/IT Function (Process, Storage, etc.) eg. Data Center	People - Role eg. Presentation Architecture	Time - System Event Cycle eg. Processing Cycle	End - Structural Assesment eg. Business Action	<i>Designer</i> TECHNOLOGY MODEL (PHYSICAL)
<i>Builder</i>	Entity - Table Segment, etc. eg. Data Definition	Process - Control Function eg. Data Embedding	Node - Hardware eg. Network	People - User eg. Security Architecture	Time - Executive Cycle eg. Timing Definition	End - Condition eg. Rule Specification	<i>Builder</i> DETAILED RELATIONS (OUT-OF-CONTEXT)
<i>Sub-Contractor</i>	Entity - Fact eg. Data	Process - Language Statement eg. Code Block	Node - Address eg. Network	People - Identity eg. Work Job	Time - Interval Cycle eg. Machine Cycle	End - Sub-condition eg. Means Step	<i>Sub-Contractor</i> FUNCTIONING ENTERPRISE

John A. Zachman, Zachman International

Top Down - Do It Right Approach

Key Points

1. The reasons you do Enterprise Architecture have to do with **complexity and change**. You cannot create a complex object if you can't describe it ... and once you get it built and want to change it, the descriptive representations (architecture) constitute the base-line for changing it.

2. The Framework for Enterprise Architecture (the "Zachman Framework") is a classification scheme for descriptive representations ... descriptive representations of **anything**. (I learned about it from looking at descriptive representations of airplanes, buildings, locomotives, battleships, etc.) I applied the same logical schema to **Enterprises**. The Framework has **nothing to do** with information systems unless the Enterprise has some stored programming devices and electronic media installed in which case, those technologies will be described in **Row 4** (not Row 1, nor Row 2 nor Row 3). The Rows 1, 2 and 3 models are descriptive of the **Enterprise**.

3. The "Zachman Framework" is a "normalized" schema - one (meta) fact in one place. It implies **nothing** about implementation process (that is, methodology: top-down, bottom-up, left-to-right, right-to-left or where to start.)

Key Points

4. It can be used to help you think about (analyze) anything ... the broader you define the boundary of the analytical target, the more leverage you get on integration, reusability, interoperability, etc., of the end object (e.g. "Enterprise") but the more complex the analysis ... and conversely, the narrower you define the boundary, the simpler the analysis, but the less leverage you are going to get on integration, reusability, interoperability, etc.

5. It is okay to attempt to after the fact, post-integrate parts that you manufactured but that don't fit together ... but you can only "integrate" (interface) cosmetic anomalies ... it is like putting lipstick on a pig. It makes the pig look better but it doesn't change the fact that it is a pig.

6. You don't have to build out all the models defined by the Framework, Enterprise-wide at excruciating levels of detail, before you can implement something ... you just have to understand that whatever slivers (vertical or horizontal) of whatever cells you are not building out (making explicit), you are making assumptions about, that is, you are assuming risk ... risk of defects ... ultimately, risk of scrap and re-work.

Key Points

7. You don't have to build Enterprise-wide models for implementations but you'd better pay attention to discontinuities in Column 1 (meaning), Column 3 (connectivity) and Column 6 (motivation) because if, after you get a bunch of things implemented (like, "the legacy"), and the data doesn't mean the same thing to everyone, the network is instable and management is not able to administer the objectives/strategies (business rules) consistently, they are going to be frustrated.

8. If you are not observing the engineering design principles as related to the primitive (cell) models, you are **not** going to realize the engineering design objectives (alignment, integration, interoperability, reusability, flexibility, reduced time-to-market, etc., etc., etc.) Composite (multi-cell) models are required for implementations. Primitive (single-cell) models are required to engineer alignment, integration, reusability, etc.

9. If you are not building (and storing, managing and changing) **primitive** models you are not doing "Architecture" ... you are doing implementations.

Key Points

10. Until you have some (primitive) models stored somewhere in such a fashion that you can find them and reuse their components, you are by definition, a "job shop" (a "waterfall") manufacturing "custom" products. You are never going to reduce time-to-market for anything substantive until you have something (parts that are designed to be reused in more than one implementation) in inventory **before** you get the order.

Note:

If anyone refers to the "Zachman Framework" and says something inconsistent with these key points, they either don't understand the logic of the Framework or they don't understand the implications of the logic.

Role of Architect

1. Determine the analytical target boundaries, i.e. the definition of "Enterprise," from the Owners and R1 models.
2. Define the templates for the models (i.e. the "meta models") for Rows 1, 2 and 3.
3. Decide which primitive models to build ... scope and level of detail ... based on the Enterprise Engineering Design Objectives.
4. Advise Enterprise Owners regarding the short-term/long-term trade-offs in not building vs building each model.
5. Assist the Owners in transcribing the contents of R1/2.
6. Assess the quality of the Rows 1 and 2 model contents:
 - a. Are they "clean" ... i.e. "Primitive?"
 - b. Are they complete in terms of the template, the analytical target (scope) and the level of detail?
 - c. Are they "normalized?" And, horizontally integrated?
7. Transform (or supervise an Engineer in transforming) the Row 2 models to Row 3 models.
8. Ensure the R4 models accurately reflect the intent of R3.
9. Ensure all the models are retained (i.e. stored).
10. Ensure the stored models accurately reflect the reality of the Enterprise as it changes over time.
11. Define and manage the EA Governance System.

Role of Owner

1. Decide on the analytical target ... that is, the definition of "Enterprise" based on jurisdictional control versus the broadest possible requirement for integration.
2. Decide on the Enterprise engineering design objectives based on Enterprise strategy.
3. Understand the implications of compromising the optimum End State Vision and make the appropriate short-term/long-term trade-off decisions.
4. Define the content of the Rows 1 and 2 Models based on management intent for the Enterprise.
5. Supervise the Architect in the transformation of Row 2 to Row 3 and review the Builder's Row 4 models.
6. Assess the resultant Row 6 implementation in terms of accurately reflecting the intent of Rows 1/2 and the R6 effectiveness in realizing the Enterprise objectives.
7. Adjust Row 1 and 2 models to improve Enterprise performance and initiate transformation to new R3-6.
8. Ensure the Architecture planning and control system (governance) is in place and functioning effectively.

Role of Builder

1. Assess the Row 3 models and decide on and acquire the implementation technologies.
2. Define the templates (the "metamodels") for the Rows 4 and 5 models.
3. Transform the Row 3 models into Row 4 models based on the Enterprise engineering design objectives.
4. Ensure the Row 5 models accurately reflect the intent of the Row 4 models.
5. Assess, in conjunction with the Owner and Architect, the resultant Row 6 implementation in terms of how accurately it reflects the intent of the Row 1/2 models.
6. Ensure the Rows 4 and 5 models are retained and accurately reflect the Enterprise as it changes over time.
7. Continuously improve the process of transforming Row 3 models to Rows 4, 5 and 6.

As Bluntly As I Know How

You may think this is too much work ...

Or, it takes too long

And costs too much

Or is too theoretical

Or too high risk

Or too whatever.

However, if that's your assessment ...

You can't complain that

the systems aren't "aligned" with the enterprise,

or are inflexible, or cost too much,

or that vital information is not available,

or the data you get isn't any good, or too late,

or you can't change anything,

or that I/S is slow and unresponsive ...

and, I am here to tell you

Outsourcing isn't going to fix the problem.

Packages (in themselves) won't fix the problem.

Decentralization won't fix the problem.

And, the Internet isn't going to fix the problem.

As Bluntly As I Know How (cont)

No amount of money
or technology
is going to fix the problem!

It is NOT a technical problem,
it is an Enterprise ENGINEERING problem.

Only ACTUAL WORK is going to fix the problem

and

"Someday, you are going to wish you had
all those models,
Enterprise wide,
horizontally and vertically integrated
at excruciating level of detail."

You might as well start working on them ...
... anytime this afternoon is probably not too early!!