

INTRODUCTION TO  
ENTERPRISE ARCHITECTURE

THE FRAMEWORK  
FOR ENTERPRISE  
ARCHITECTURE

# 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?

# ARCHITECTURE

Architecture ... what is it?

Some people think this is Architecture:



That is a common  
**MISCONCEPTION**

(Note: This same misconception about Enterprises is what leads people to misconstrue Enterprise Architecture as being big, monolithic, static, inflexible and unachievable and ... it takes too long and costs too much.)

# ARCHITECTURE

This is the RESULT of architecture. In the RESULT you can see the Architect's "architecture".

The RESULT is an implementation, an instance.



"Architecture" IS the set of descriptive representations relevant for describing a complex object (actually, any object) such that an instance of the object can be created and such that the descriptive representations serve as the baseline for changing an object instance (assuming that the descriptive representations are maintained consistent with the instantiation).

# “ARCHITECTURE”

If the object you are trying to create is simple, you can see the whole thing all at one time, and it is not likely to change, (e.g. a log cabin, a program, etc.), then you don't need Architecture.



```
for m1 = 1,M do begin
  for m2 = 1,M do begin
    for u1 = u_min,u_max do begin
      for u2 = u_min,u_max do begin
        if u1 gt u2 then begin
          for v1 = v_min,v_max do begin
            if v1 lt u1 then begin
              for v2 = v_min,v_max do begin
                if v2 ge v1 then begin
                  KE_B = double(m1*u1^2+m2*u2^2)
                  KE_A = double(m1*v1^2+m2*v2^2)
                  if (KE_B gt KE_A) and (KE_A ge 0.965*KE_B) then begin
                    x_axis[index]=index
                    LM_B = double(m1*u1+n2*u2)
                    LM_A = double(m1*v1+n2*v2)
                    y_LM_Diffs[index]=LM_B-LM_A
                    Total_LM=Total_LM+LM_B-LM_A
                    y_LM_Total[index]=double(Total_LM/(index+1))
                    index=index+1
                    if index gt 65535 then goto, end_of_loop
                  endif
                endif
              endfor
            endif
          endfor
        endif
      endfor
    endfor
  endfor
endfor
endfor
endfor
endfor
```

All you need is a tool (e.g. an ax, a compiler, etc.), some raw material (e.g. a forest, some data, etc.) and some time (then, build log cabins, write programs, etc.).

# “ARCHITECTURE”



On the other hand, if the object is complex, you can't see it in its entirety at one time and it is likely to change considerably over time (e.g. a hundred story building, or an Enterprise, etc.), now you need Architecture.



In short, the reasons you need  
Architecture:  
**COMPLEXITY AND CHANGE**

# “ARCHITECTURE”

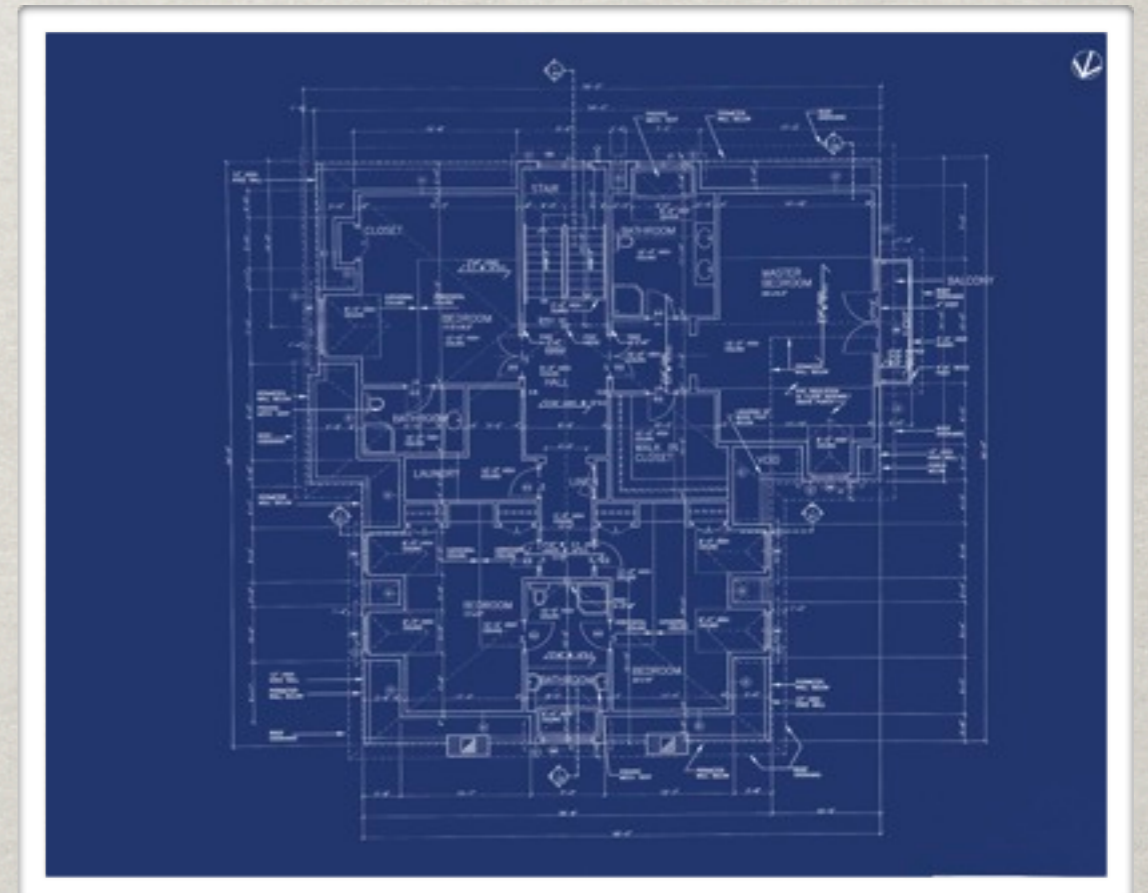
## COMPLEXITY

If you can't describe it, you can't create it (whatever "it" is).

## CHANGE

If you don't retain the descriptive representations after you create them (or if you never created them in the first place) and you need to change the resultant implementation, you have only three options:

- ☀ Change the instance and see what happens. (High risk!)
- ☀ Recreate ("reverse engineer") the architectural representations from the existing ("as is") implementation. (Takes time and costs money!)
- ☀ Scrap the whole thing and start over again.



# “ARCHITECTURE”

There is not a single descriptive representation for a complex object ... there is a SET of descriptive representations.

Descriptive representations (of anything) typically include :

## **Abstractions:**

## as well as **Perspectives:**

✱ Bills of Material	(What)	✱ Scoping Boundaries	(Planners)
✱ Functional Specs	(How)	✱ Requirement Concepts	(Owners)
✱ Drawings	(Where)	✱ Design Logic	(Designers)
✱ Operating Instructions	(Who)	✱ Plan Physics	(Builders)
✱ Timing Diagrams	(When)	✱ Part Configurations	(Implementers)
✱ Design Objectives	(Why)	✱ Product Instances	(Operators)



# ABSTRACTIONS

Version 3.0



Bills of Material

Functional Specs

Drawings

Operating Instructions

Timing Diagrams

Design Objectives

# “ARCHITECTURE”

There is not a single descriptive representation for a complex object ... there is a SET of descriptive representations.

Descriptive representations (of anything) typically include :

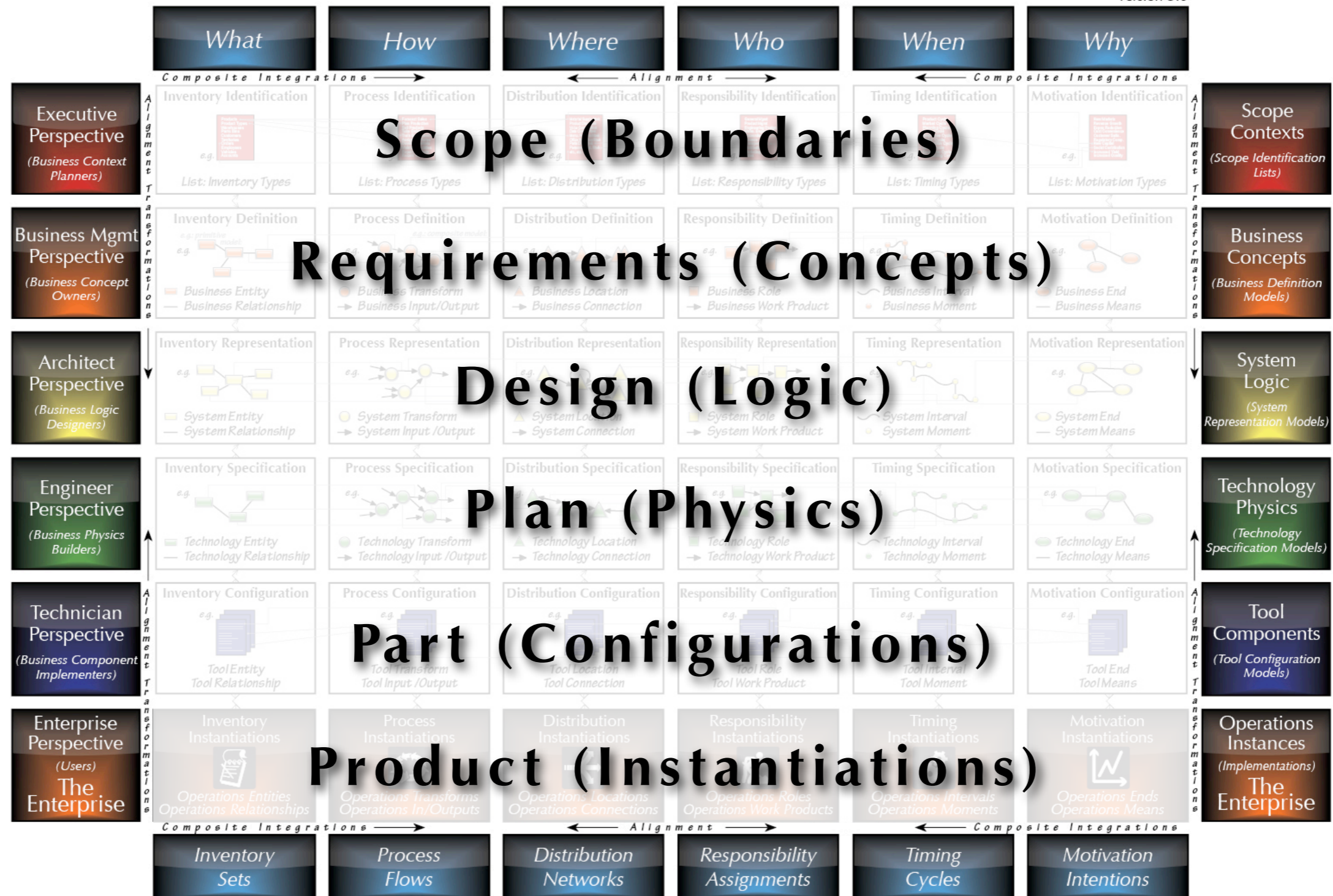
**Abstractions:**

**as well as Perspectives:**

✱ Bills of Material	(What)	✱ Scoping Boundaries	(Planners)
✱ Functional Specs	(How)	✱ Requirement Concepts	(Owners)
✱ Drawings	(Where)	✱ Design Logic	(Designers)
✱ Operating Instructions	(Who)	✱ Plan Physics	(Builders)
✱ Timing Diagrams	(When)	✱ Part Configurations	(Implementers)
✱ Design Objectives	(Why)	✱ Product Instances	(Operators)

# PERSPECTIVES

Version 3.0



# REIFICATION

Version 3.0



# ARCHITECTURE IN GENERAL

"Architecture" (for anything) would be the total set of descriptive representations (models) relevant for describing a complex object such that it can be created and that constitute a baseline for changing the object after it has been instantiated. The relevant descriptive representations would necessarily have to include all the intersections between:

## Abstractions:

- ✱ Bills of Material (What)
- ✱ Functional Specs (How)
- ✱ Drawings (Where)
- ✱ Operating Instructions (Who)
- ✱ Timing Diagrams (When)
- ✱ Design Objectives (Why)

## Perspectives:

- ✱ Scoping Boundaries (Identification)
- ✱ Requirement Concepts (Definition)
- ✱ Design Logic (Representation)
- ✱ Plan Physics (Specification)
- ✱ Part Configurations (Configuration)
- ✱ Product Instances (Instantiation)

# “ENTERPRISE ARCHITECTURE”

Therefore "Enterprise Architecture" would be the total set of descriptive representations (models) relevant for describing an Enterprise, that is, the descriptive representations required to create (a coherent, optimal) Enterprise and required to serve as a baseline for changing the Enterprise once it is created. The total set of relevant descriptive representations would necessarily have to include all the intersections between the...

## Abstractions:

- ✱ Inventory Models (Bills of Material)
- ✱ Process Models (Functional Specs)
- ✱ Distribution Models (Drawings)
- ✱ Responsibility Models (Operating Instructions)
- ✱ Timing Models (Timing Diagrams)
- ✱ Motivation Models (Design Objectives)

## Perspectives:

- ✱ Scope Contexts (Scoping Boundaries)
- ✱ Business Models (Requirement Concepts)
- ✱ System Models (Design Logic)
- ✱ Technology Models (Plan Physics)
- ✱ Tooling Configurations (Part Configurations)
- ✱ Enterprise Implementation (Product Instances)

# ENTERPRISE ARCHITECTURE

The total set would necessarily have to include **Abstractions**:

## 1 WHAT

**Inventory Models equal Bills of Materials**  
(Entity Models and Data Models ARE Bills  
of Material)

## 2 HOW

**Process Models equal Functional Specs**  
(Transformation Models)

## 3 WHERE

**Distribution Models equal Drawings**  
(Geographic Models)  
(Network Models) (Geometry)

## 4 WHO

**Responsibility Models equal Operating  
Instructions**  
(Work Flow Models) (Presentation  
Architecture)

## 5 WHEN

**Timing Models equal Timing Diagrams**  
(Control Structures) (Cyclical Models)  
(Dynamics Models)

## 6 WHY

**Motivation Models equal Design Objectives**

# ABSTRACTIONS

Version 3.0





# ENTERPRISE ARCHITECTURE

The total set would necessarily have to include **Perspectives:**

## 1 EXECUTIVE

**Scope Boundaries equal Scope Boundaries**  
("CONOPS" or Concepts Package)

## 2 BUSINESS MGMT

**Business Models equal Requirement**  
**Concepts**

(Concepts Models) (Customer's Usage)  
("Computation Independent")

## 3 ARCHITECT

**System Models equal Design Logic**  
(Logic Models) (Engineering Descriptions)  
("Platform Independent")

## 4 ENGINEER

**Technology Models equal Plan Physics**  
(Physics Models) (Mfg. Eng. Descriptions)  
("Platform Specific")

## 5 TECHNICIAN

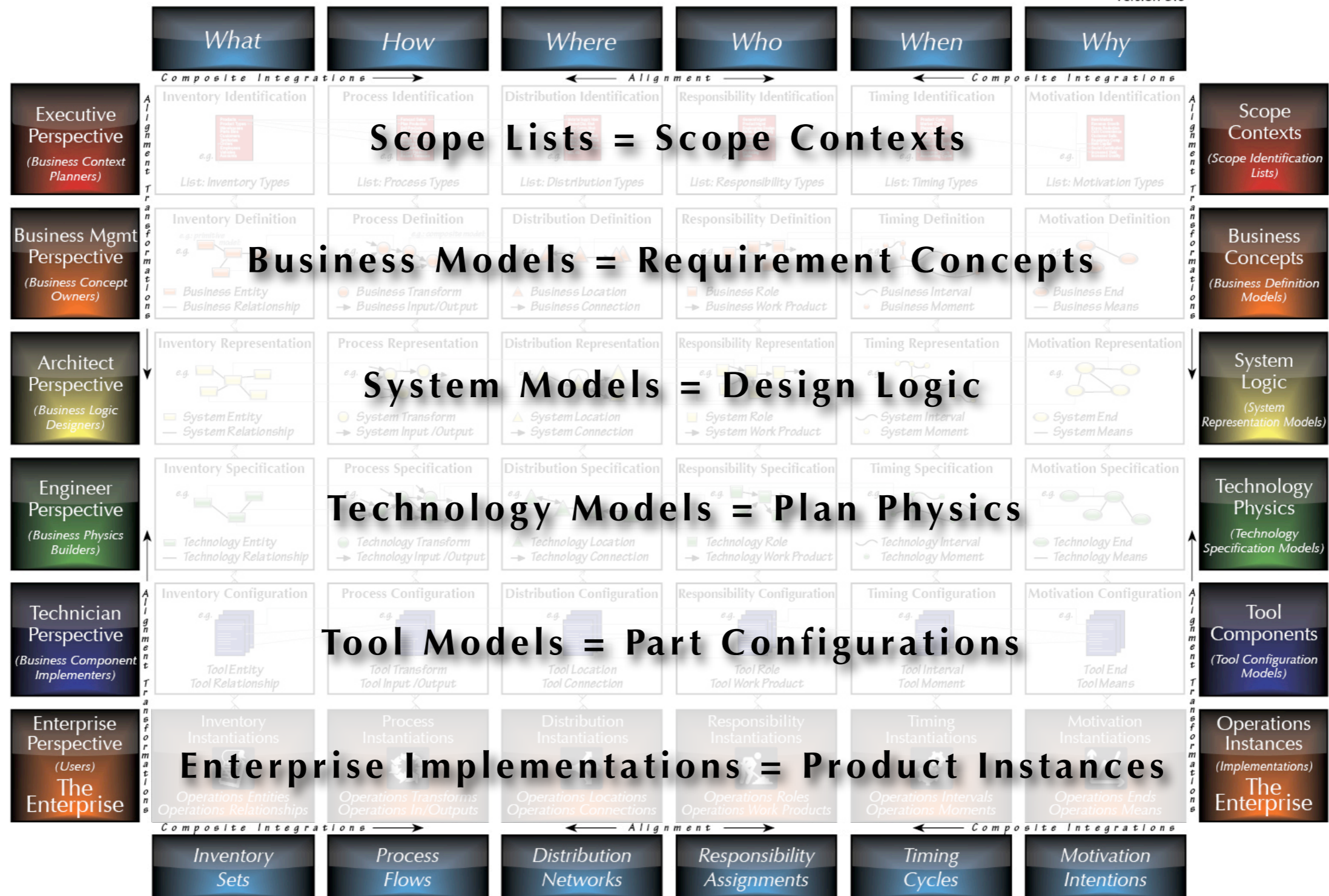
**Tooling Configurations equal Part**  
**Configurations**  
(Vendor Product Specific) (Machine  
Tool Specific)

## 6 BUSINESS

**Enterprise Implementation equals**  
**Product Instance**  
(Operations Instances)

# PERSPECTIVES

Version 3.0



# The Zachman Framework for Enterprise Architecture™

## The Enterprise Ontology™

Version 3.0



© 1987-2011 John A. Zachman, all rights reserved. Zachman® and Zachman International® are registered trademarks of John A. Zachman

\*Horizontal integration lines are shown for example purposes only and are not a complete set. Composite, integrative relationships connecting every cell horizontally potentially exist.

# FRAMEWORK GRAPHIC

For the latest version of the Framework Graphic,  
register at [Zachman.com](http://Zachman.com)  
for a high resolution .pdf file.

For a publication release of the Framework Graphic  
send requests to the Contact Us link on Zachman.com

# INTRODUCTION TO ZACHMAN FRAMEWORK

The Framework for  
Enterprise Architecture  
(The Zachman Framework)  
presently appears to be a  
grossly misunderstood  
concept among the IT  
community.

It is NOT a methodology.

It is an ONTOLOGY.

It is likely perceived to be a  
methodology for two reasons:

- ✿ IT, in general, thinks in terms of methodologies because we perceive our role to be one of building and running systems - Implementation.
- ✿ Every other popular “Framework” is either a methodology or derived from a methodology. (I will show you how to test for that later.)

# ZACHMAN FRAMEWORK MISUNDERSTANDING No. 1

The Zachman Framework is the Enterprise ONTOLOGY  
NOT  
a METHODOLOGY

# ONTOLOGY

The Zachman Framework™ schema technically is an ontology - a theory of the existence of a structured set of essential components of an object for which explicit expression is necessary (is mandatory?) for designing, operating and changing the object (the object being an Enterprise, a department, a value chain, a "sliver," a solution, a project, an airplane, a building, a bathtub or whatever or whatever).

The Zachman Framework™ is NOT a methodology for creating the implementation (an instantiation) of the object (i.e. the Framework is an ontology, not a methodology).

# ONTOLOGY

**PERIODIC TABLE OF THE ELEMENTS**

<http://www.kjf-split.hr/periodni/en/>

GROUP	1 IA	2 IIA											13 IIIA	14 IVA	15 VA	16 VIA	17 VIIA	18 VIIIA
PERIOD 1	1 1.0079 <b>H</b> HYDROGEN																	2 4.0026 <b>He</b> HELIUM
PERIOD 2	3 6.941 <b>Li</b> LITHIUM	4 9.0122 <b>Be</b> BERYLLIUM											5 10.811 <b>B</b> BORON	6 12.011 <b>C</b> CARBON	7 14.007 <b>N</b> NITROGEN	8 15.999 <b>O</b> OXYGEN	9 18.998 <b>F</b> FLUORINE	10 20.180 <b>Ne</b> NEON
PERIOD 3	11 22.990 <b>Na</b> SODIUM	12 24.305 <b>Mg</b> MAGNESIUM											13 26.982 <b>Al</b> ALUMINIUM	14 28.086 <b>Si</b> SILICON	15 30.974 <b>P</b> PHOSPHORUS	16 32.065 <b>S</b> SULPHUR	17 35.453 <b>Cl</b> CHLORINE	18 39.948 <b>Ar</b> ARGON
PERIOD 4	19 39.098 <b>K</b> POTASSIUM	20 40.078 <b>Ca</b> CALCIUM	21 44.956 <b>Sc</b> SCANDIUM	22 47.867 <b>Ti</b> TITANIUM	23 50.942 <b>V</b> VANADIUM	24 51.996 <b>Cr</b> CHROMIUM	25 54.938 <b>Mn</b> MANGANESE	26 55.845 <b>Fe</b> IRON	27 58.933 <b>Co</b> COBALT	28 58.693 <b>Ni</b> NICKEL	29 63.546 <b>Cu</b> COPPER	30 65.39 <b>Zn</b> ZINC	31 69.723 <b>Ga</b> GALLIUM	32 72.64 <b>Ge</b> GERMANIUM	33 74.922 <b>As</b> ARSENIC	34 78.96 <b>Se</b> SELENIUM	35 79.904 <b>Br</b> BROMINE	36 83.80 <b>Kr</b> KRYPTON
PERIOD 5	37 85.468 <b>Rb</b> RUBIDIUM	38 87.62 <b>Sr</b> STRONTIUM	39 88.906 <b>Y</b> YTTRIUM	40 91.224 <b>Zr</b> ZIRCONIUM	41 92.906 <b>Nb</b> NIOBIUM	42 95.94 <b>Mo</b> MOLYBDENUM	43 (98) <b>Tc</b> TECHNETIUM	44 101.07 <b>Ru</b> RUTHENIUM	45 102.91 <b>Rh</b> RHODIUM	46 106.42 <b>Pd</b> PALLADIUM	47 107.87 <b>Ag</b> SILVER	48 112.41 <b>Cd</b> CADMIUM	49 114.82 <b>In</b> INDIUM	50 118.71 <b>Sn</b> TIN	51 121.76 <b>Sb</b> ANTIMONY	52 127.60 <b>Te</b> TELLURIUM	53 126.90 <b>I</b> IODINE	54 131.29 <b>Xe</b> XENON
PERIOD 6	55 132.91 <b>Cs</b> CAESIUM	56 137.33 <b>Ba</b> BARIUM	57-71 <b>La-Lu</b> Lanthanide	72 178.49 <b>Hf</b> HAFNIUM	73 180.95 <b>Ta</b> TANTALUM	74 183.84 <b>W</b> TUNGSTEN	75 186.21 <b>Re</b> RHENIUM	76 190.23 <b>Os</b> OSMIUM	77 192.22 <b>Ir</b> IRIDIUM	78 195.08 <b>Pt</b> PLATINUM	79 196.97 <b>Au</b> GOLD	80 200.59 <b>Hg</b> MERCURY	81 204.38 <b>Tl</b> THALLIUM	82 207.2 <b>Pb</b> LEAD	83 208.98 <b>Bi</b> BISMUTH	84 (209) <b>Po</b> POLONIUM	85 (210) <b>At</b> ASTATINE	86 (222) <b>Rn</b> RADON
PERIOD 7	87 (223) <b>Fr</b> FRANCIUM	88 (226) <b>Ra</b> RADIUM	89-103 <b>Ac-Lr</b> Actinide	104 (261) <b>Rf</b> RUTHERFORDIUM	105 (262) <b>Db</b> DUBNIUM	106 (266) <b>Sg</b> SEABORGIUM	107 (264) <b>Bh</b> BOHRNIUM	108 (277) <b>Hs</b> HASSIUM	109 (268) <b>Mt</b> MEITNERIUM	110 (281) <b>Uun</b> UNUNNIUM	111 (272) <b>Uuu</b> UNUNUNIUM	112 (285) <b>Uub</b> UNUNBIUM				114 (289) <b>Uuq</b> UNUNQUADIUM		

**Legend:**

- Metal (Blue)
- Semimetal (Orange)
- Nonmetal (Green)
- Alkali metal (1)
- Alkaline earth metal (2)
- Transition metals (3-10)
- Lanthanide (Pink)
- Actinide (Purple)
- Chalcogens element (16)
- Halogens element (17)
- Noble gas (18)

**STANDARD STATE (25 °C; 101 kPa):**  
 Ne - gas, Fe - solid, Ga - liquid, Te - synthetic

**LANTHANIDE**

57 138.91 <b>La</b> LANTHANUM	58 140.12 <b>Ce</b> CERIUM	59 140.91 <b>Pr</b> PRASEODYMIUM	60 144.24 <b>Nd</b> NEODYMIUM	61 (145) <b>Pm</b> PROMETHIUM	62 150.36 <b>Sm</b> SAMARIUM	63 151.96 <b>Eu</b> EUROPIUM	64 157.25 <b>Gd</b> GADOLINIUM	65 158.93 <b>Tb</b> TERBIUM	66 162.50 <b>Dy</b> DYSPROSIUM	67 164.93 <b>Ho</b> HOLMIUM	68 167.26 <b>Er</b> ERBIUM	69 168.93 <b>Tm</b> THULIUM	70 173.04 <b>Yb</b> YTTERBIUM	71 174.97 <b>Lu</b> LUTETIUM
-------------------------------------	----------------------------------	--	-------------------------------------	-------------------------------------	------------------------------------	------------------------------------	--------------------------------------	-----------------------------------	--------------------------------------	-----------------------------------	----------------------------------	-----------------------------------	-------------------------------------	------------------------------------

**ACTINIDE**

89 (227) <b>Ac</b> ACTINIUM	90 232.04 <b>Th</b> THORIUM	91 231.04 <b>Pa</b> PROTACTINIUM	92 238.03 <b>U</b> URANIUM	93 (237) <b>Np</b> NEPTUNIUM	94 (244) <b>Pu</b> PLUTONIUM	95 (243) <b>Am</b> AMERICIUM	96 (247) <b>Cm</b> CURIUM	97 (247) <b>Bk</b> BERKELIUM	98 (251) <b>Cf</b> CALIFORNIUM	99 (252) <b>Es</b> EINSTEINIUM	100 (257) <b>Fm</b> FERMIUM	101 (258) <b>Md</b> MENDELEVIUM	102 (259) <b>No</b> NOBELIUM	103 (262) <b>Lr</b> LAWRENCIUM
-----------------------------------	-----------------------------------	--	----------------------------------	------------------------------------	------------------------------------	------------------------------------	---------------------------------	------------------------------------	--------------------------------------	--------------------------------------	-----------------------------------	---------------------------------------	------------------------------------	--------------------------------------

(1) Pure Appl. Chem., 73, No. 4, 667-683 (2001)  
 Relative atomic mass is shown with five significant figures. For elements having no stable nuclides, the value enclosed in brackets indicates the mass number of the longest-lived isotope of the element.  
 However three such elements (Th, Pa, and U) do have a characteristic terrestrial isotopic composition, and for these an atomic weight is tabulated.  
 Editor: Aditya Varshan (adivar@netlink.com)

Copyright © 1998-2003 EniG. (eni@kjf-split.hr)

Until an ontology exists, nothing is repeatable, nothing is predictable.  
 There is no DISCIPLINE.



# PROCESS

(aka. METHODOLOGY)

Add Bleach to an Alkali and  
it is transformed into Saltwater.



## Composites

Salt	$\text{NaCl}$
Aspirin	$\text{C}_9\text{H}_8\text{O}_4$
Vicodin	$\text{C}_{18}\text{H}_{21}\text{NO}_3$
Naproxen	$\text{C}_{14}\text{H}_{14}\text{O}_3$
Ibuprophen	$\text{C}_{13}\text{H}_{18}\text{O}_2$
Viagra	$\text{C}_{22}\text{H}_{30}\text{N}_6\text{O}_4\text{S}$
Sulphuric Acid	$\text{H}_2\text{SO}_4$
Water	$\text{H}_2\text{O}$

etc., etc., etc.



# ONTOLOGY VS METHODOLOGY

An Ontology is the classification of the total set of “Primitive” (elemental) components that exist and that are relevant to the existence of an object.

A Methodology produces composite (compound) implementations of the Primitives.

Primitives (elements) are timeless.  
Composites (compounds) are temporal.

# This is a Methodology **WITHOUT** an Ontology

A Process with no ontological structure is ad hoc, fixed and dependent on practitioner skills. This is **NOT** a science. It is *ALCHEMY*, a "practice."



# WHY ENTERPRISE ARCHITECTURE? (INVENTORY OF PRIMITIVE COMPONENTS)

- ☼ Reduce Enterprise Operating Costs, General and Administrative costs, make the Enterprise LEAN. Minimum possible cost of operations.
  - ☼ Design Objective: ENTERPRISE INTEGRATION
- ☼ Reduce the time, disruption and cost of Enterprise Change, predict impacts.
  - ☼ Design Objective: ENTERPRISE FLEXIBILITY
- ☼ Ensure Enterprise operations reflects the intentions of Management
  - ☼ Design Objective: ENTERPRISE ALIGNMENT
- ☼ Make the Enterprise "MEAN" - Reduce response time to external demands.
  - ☼ Design Objective: ENTERPRISE MASS-CUSTOMIZATION, REUSE
- ☼ Enable the Enterprise to "INTEROPERATE" with other Enterprises outside of its jurisdictional control.
  - ☼ Design Objective: FEDERATED ARCHITECTURE

# ZACHMAN FRAMEWORK MISUNDERSTANDING No. 2

I NEVER said, “stop the music for 15 or 20 years and build a bunch of models and then you can do actual work again.”

I SAID ... “SOMEDAY, you are going to WISH ...”, in fact, I said, “someday THE ENTERPRISE is going to wish ...”

(The 80:20 rule ... or maybe the 20:80 rule.)