System Integration Frameworks

Joseph J. Simpson

Mary J. Simpson

July, 2005

Overview

- Definitions of System, Meta-System
- Integration and Complexity Reduction
- Classical System Engineering Approach
- System Engineering Patterns, CCFRAT
- System Engineering Cognitive Support
- System Engineering Language Proposal

Definitions

System

"**A constraint on variety.**" (Heylighen 1994)

['Functional' Definition – Constraint identifies and defines the system]

"**A non-empty set of objects and a non-empty set of relationships mapped over these objects and their attributes.**" (Simpson & Simpson 2003)

['Construction Rule' Definition]

Meta-system*

"**A constrained variation of constrained varieties.**" (Heylighen 1994)

"**A set of value sentences which describe the wanted physical system, and which imply or actually comprise the parts and relationships of the meta-system.**" (A.D. Hall, 1989)

"**Indicates the field within which the system arises and within which it interacts with other systems.**" (K.D. Palmer, 2000)

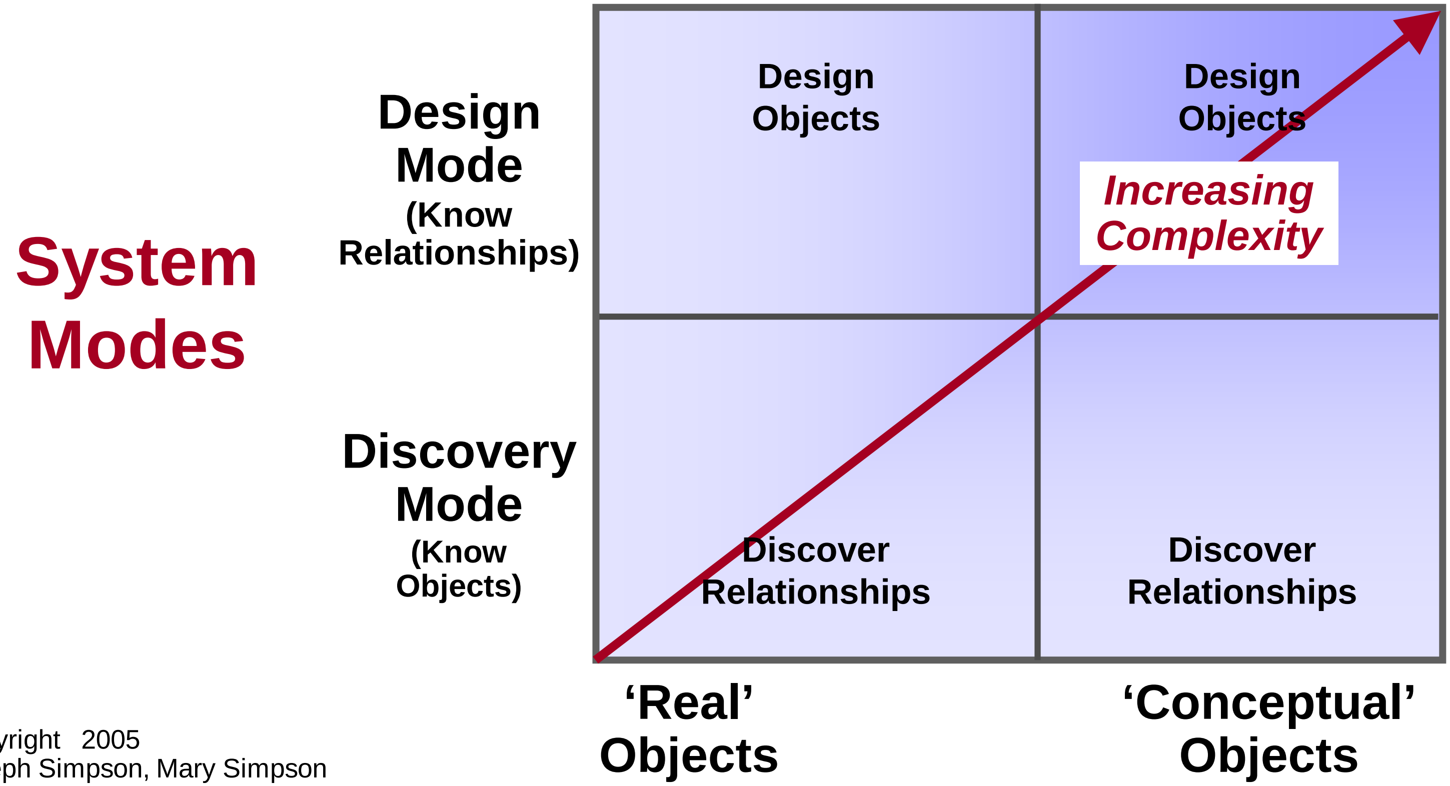
**** Related definitions in that:***

1. ***All value determined in context***
2. ***Concept of system used in different modes***

A Mapping Context for Complexity

System	*Objects* *“Over which we map”*	*Relationships*
Discovery Mode	Know the Objects	Discover the Relationships
Design Mode	Design the Objects	Know the Relationships

Discovery (Kepler)	Know the Planets	Discover the Mathematical Relationships
Design (Kennedy)	Design the Objects, Config	Know “Man on the Moon”



Types of System Complexity

- **Type 1 System Complexity - Allocates the complexity characteristic to the system of interest**
  - Number of objects – Weinberg
  - Number of relationships
  - Number of different types of objects
  - Number of different types of relationships (Casti)
  - Rate of change of objects
  - Rate of change of relationships (Casti)
  - Rate of change of the environment
  - Range of variability
- **Type 2 System Complexity - Allocates the complexity characteristic to the relationship between two systems (Warfield and Casti)**
  - Number of relationships
  - Number of observers
  - Difference in abstraction level (ant hill to ant – not complex
ant hill to human – very complex)

Problem Space Topology

Increasing Complexity

Complexity # of Individuals, # of Variables	Problem Space Well-defined vs Ill-Defined	Solution Space Unique vs Multiple Solution(s)
Simple	Well-Defined	Closed
Simple	Ill-Defined	Closed
Simple	Well-Defined	Open
Simple	Ill-Defined	Open
Complex	Well-Defined	Closed
Complex	Ill-Defined	Closed
Complex	Well-Defined	Open
Complex	Ill-Defined	Open

Deductive

Inductive

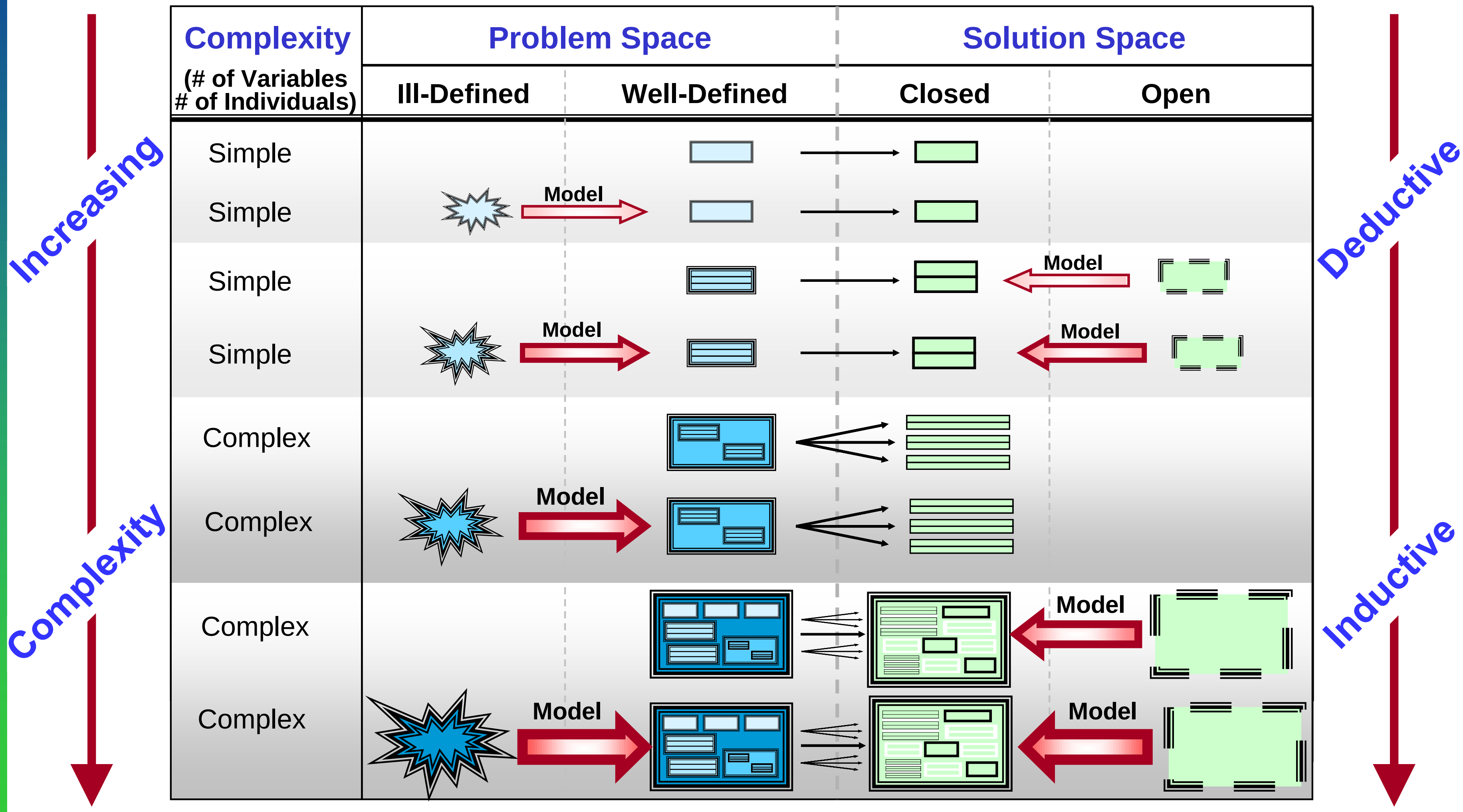
Features of a system are largely driven by its problem space

A systems approach is characterized hierarchically by

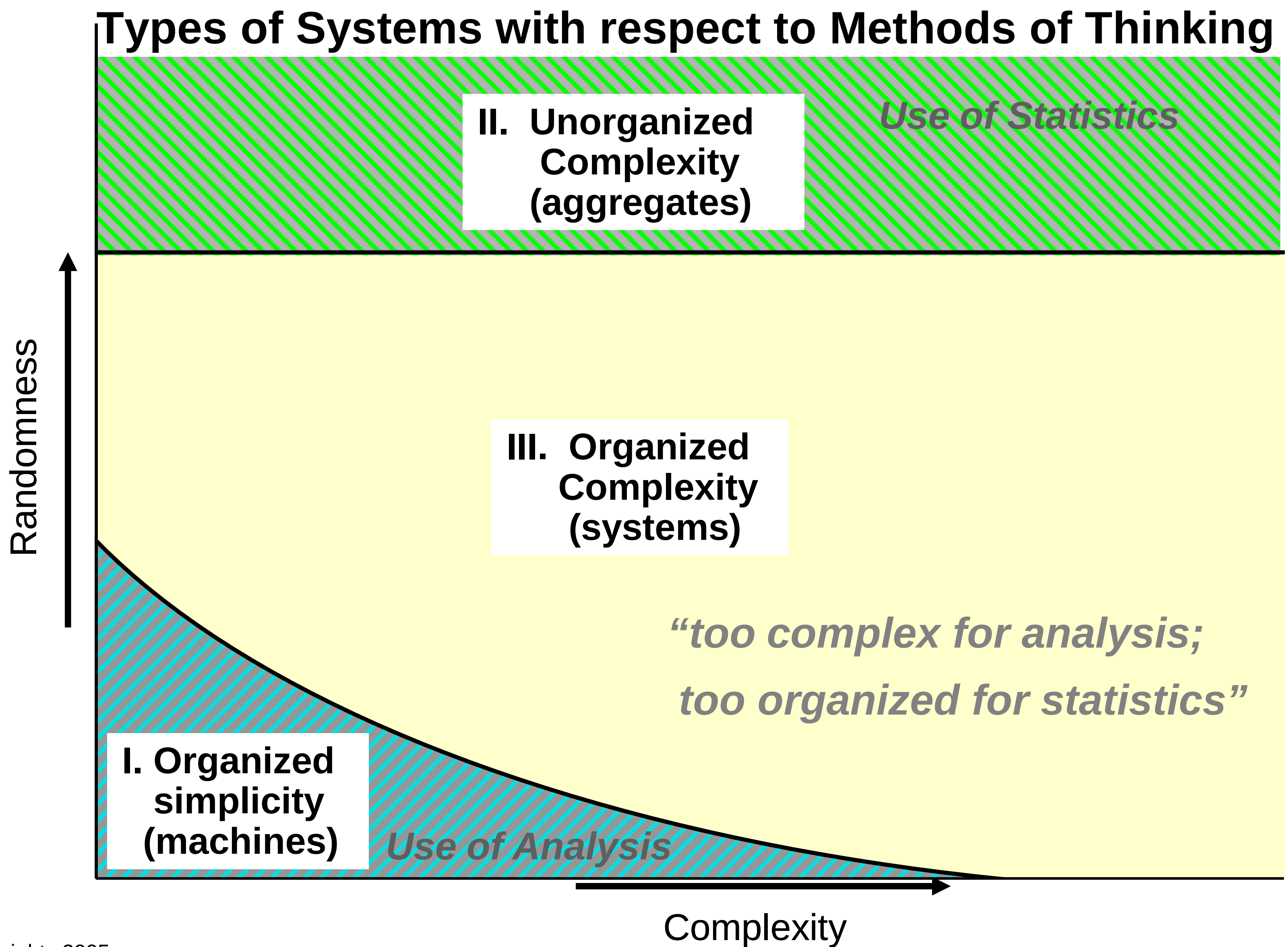
- **Abstraction frames**
- **Degree of complexity**
- **Levels of detail**

Table adapted from Chapter 9 Problem Solving Skills, *Introduction to the Engineering of Complex Systems*, Brian W. Mar, 1996.

System Conceptual Topology

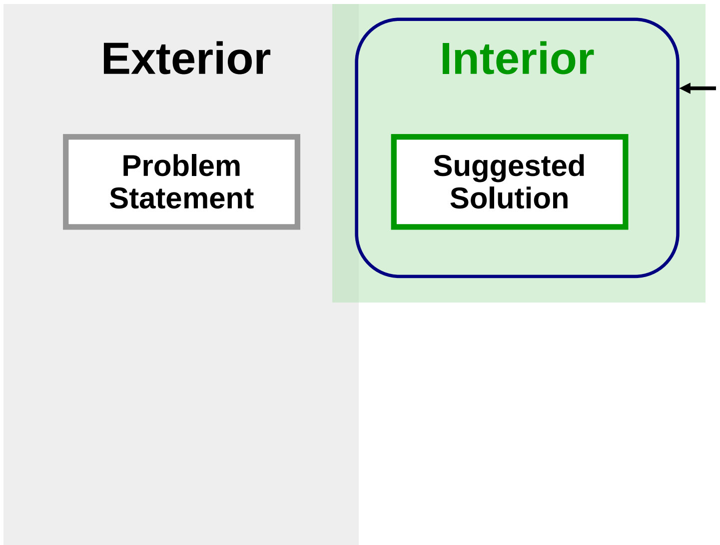


Weinberg's Conceptual Regions



Adapted from "introduction to General Systems Thinking" G.M. Weinberg

Goode & Machol System Design (1)

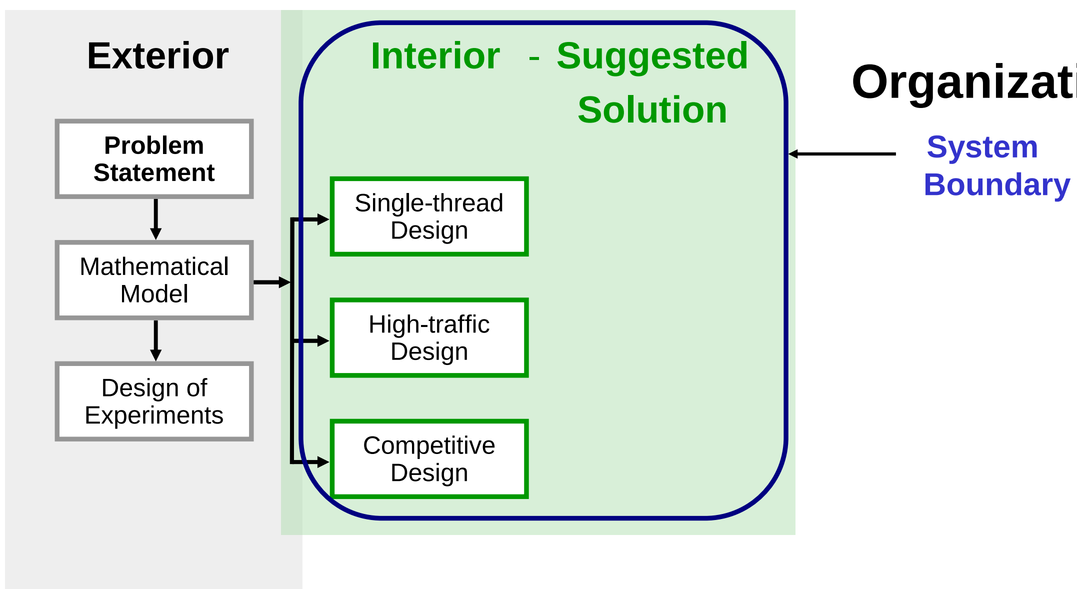


Initiation Phase – Initiates the Project

- ***Exterior*** portion concerns itself with things outside of the system: requirements on the system and its environment
- ***Interior*** portion concerns itself with design choices relative to equipment, procedures and people: the system itself

Problem is outside the System

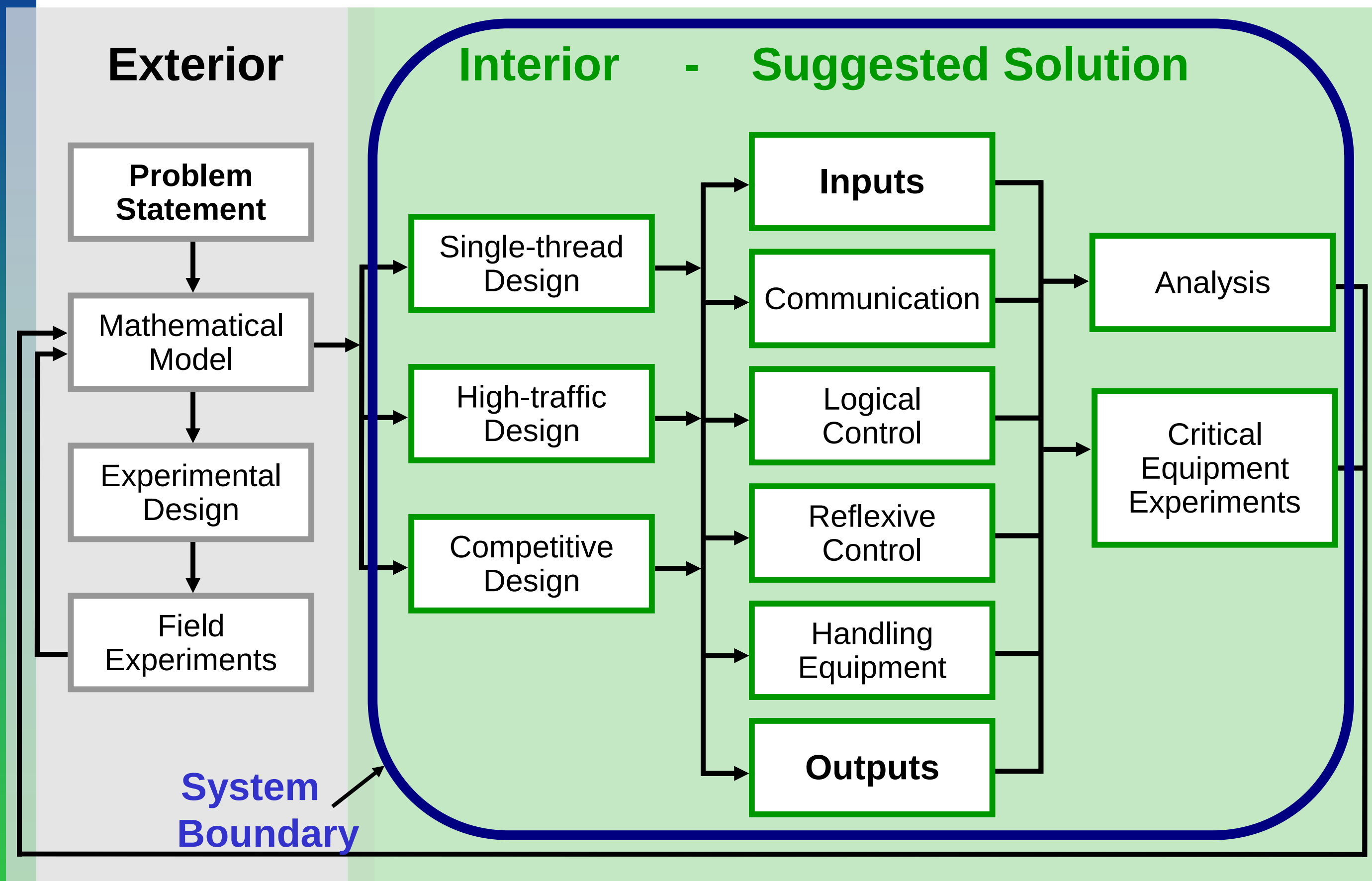
Solution is inside the System



Organization Phase

Goode & Machol System Design (2)

Preliminary Design Phase



Achieves 'First' System

- Answers questions which have only equivocal answers, iterates & documents the reasoning for choices made
- Describes overall system operation in considerable detail

Creates the Initial System Boundary

- Delineates each subsystem by (a) inputs and outputs, (b) its functioning (operation on its input to produce its output), (c) limiting specifications concerning allowable sizes, weights, etc., and (d) at least one method of physically realizing proposed function within these limitations

* Goode, Harry H. and Machol, Robert E., *System Engineering,* McGraw-Hill Book Company, Inc, 1957

Relationship between Systems

Warfield, Poly-Loop Model in Problem Solving

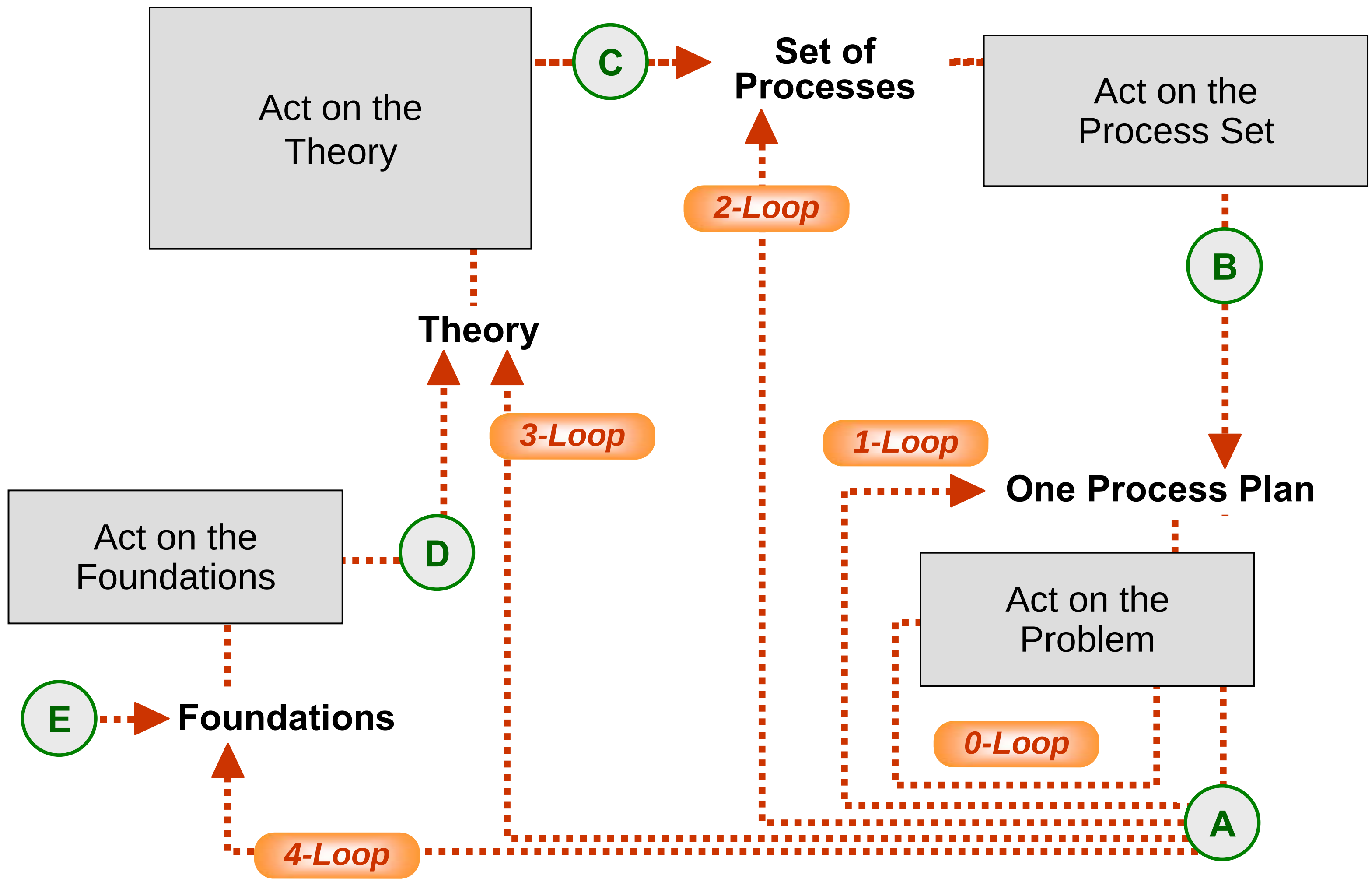
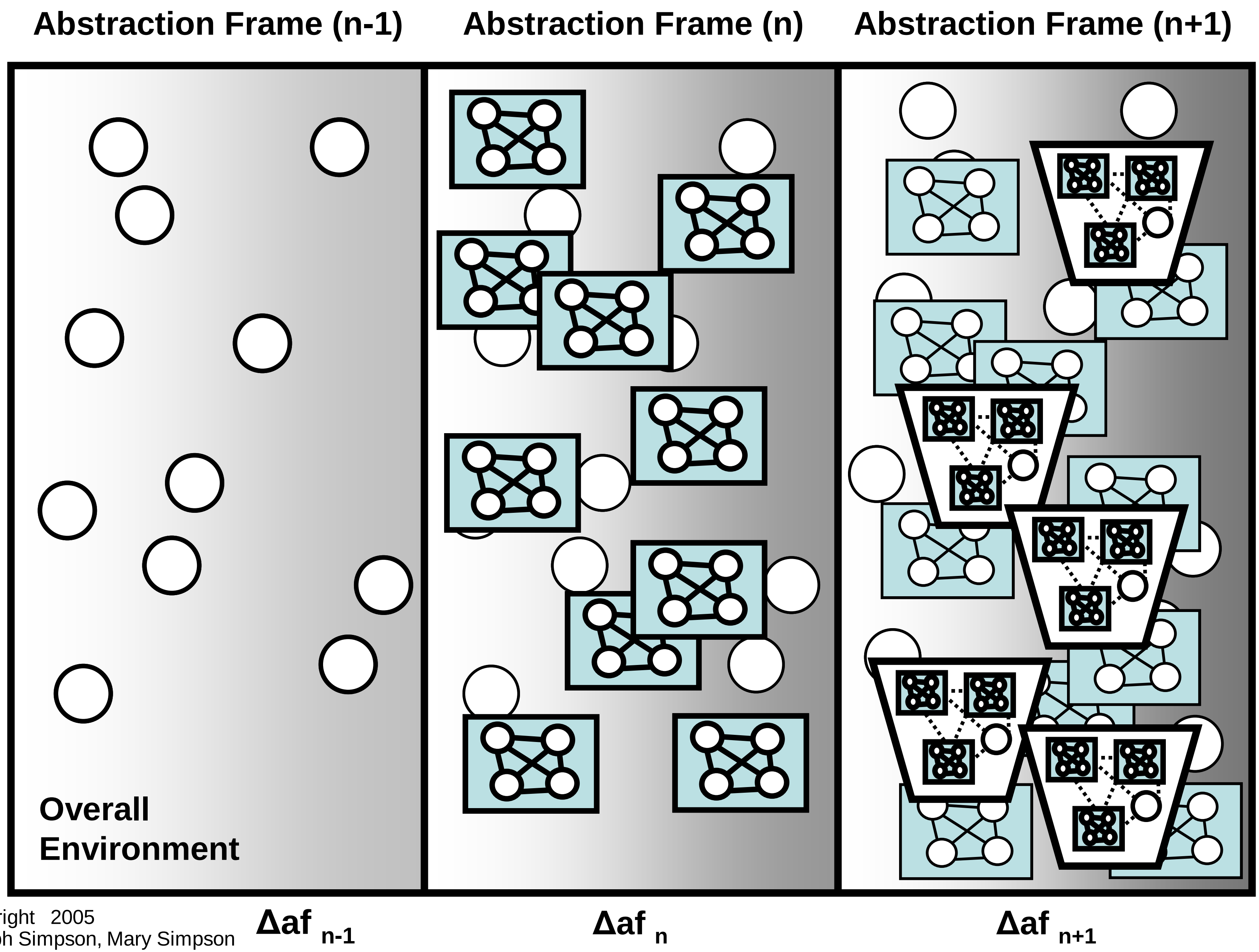


Figure adapted from Chapter 4 Managing Complexity through Systems Design: The Use of the Science, *A Science of Generic Design*, John W Warfield, 1994.

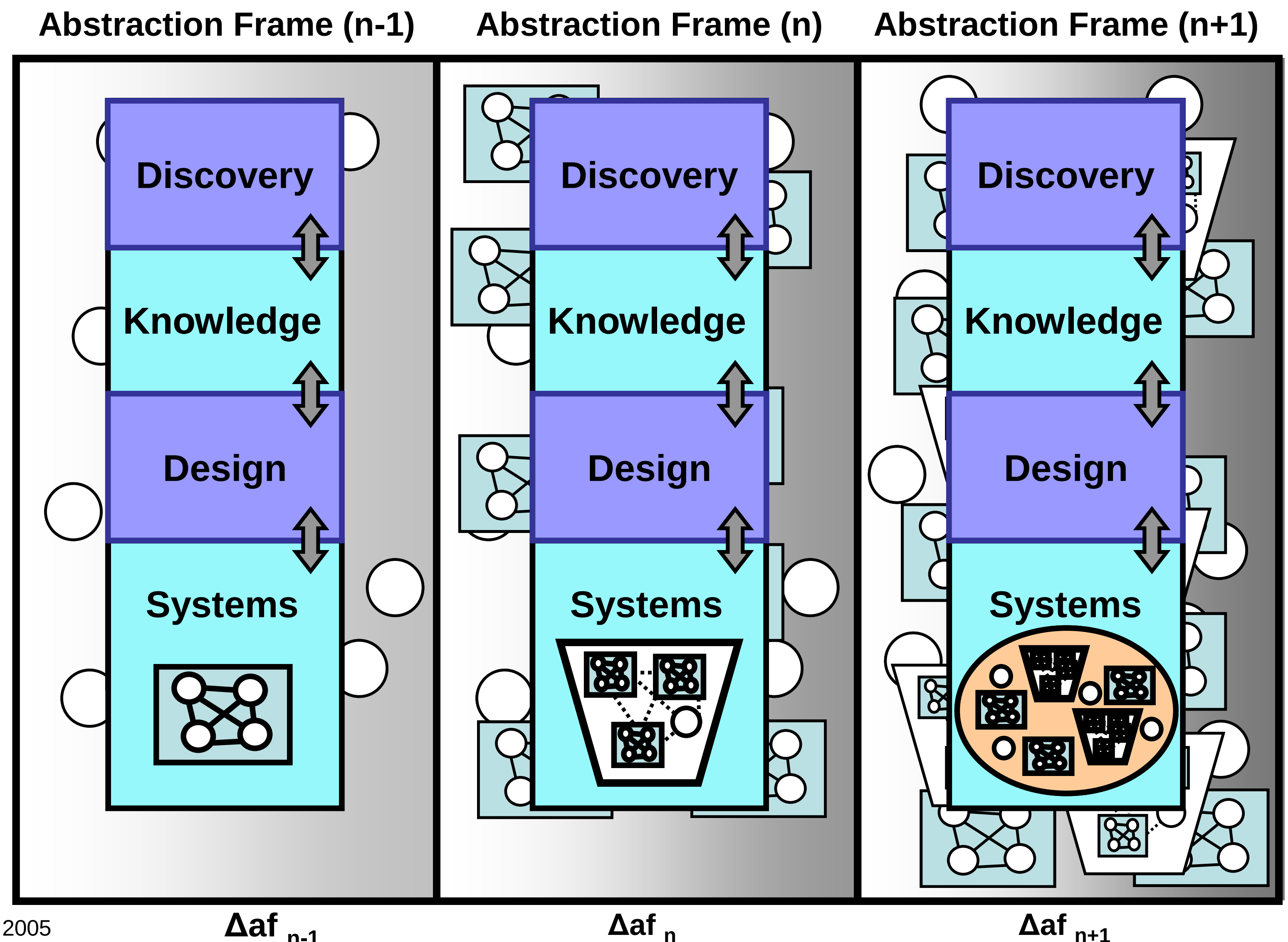
System Integration Frameworks

Abstraction Frame Sequencing

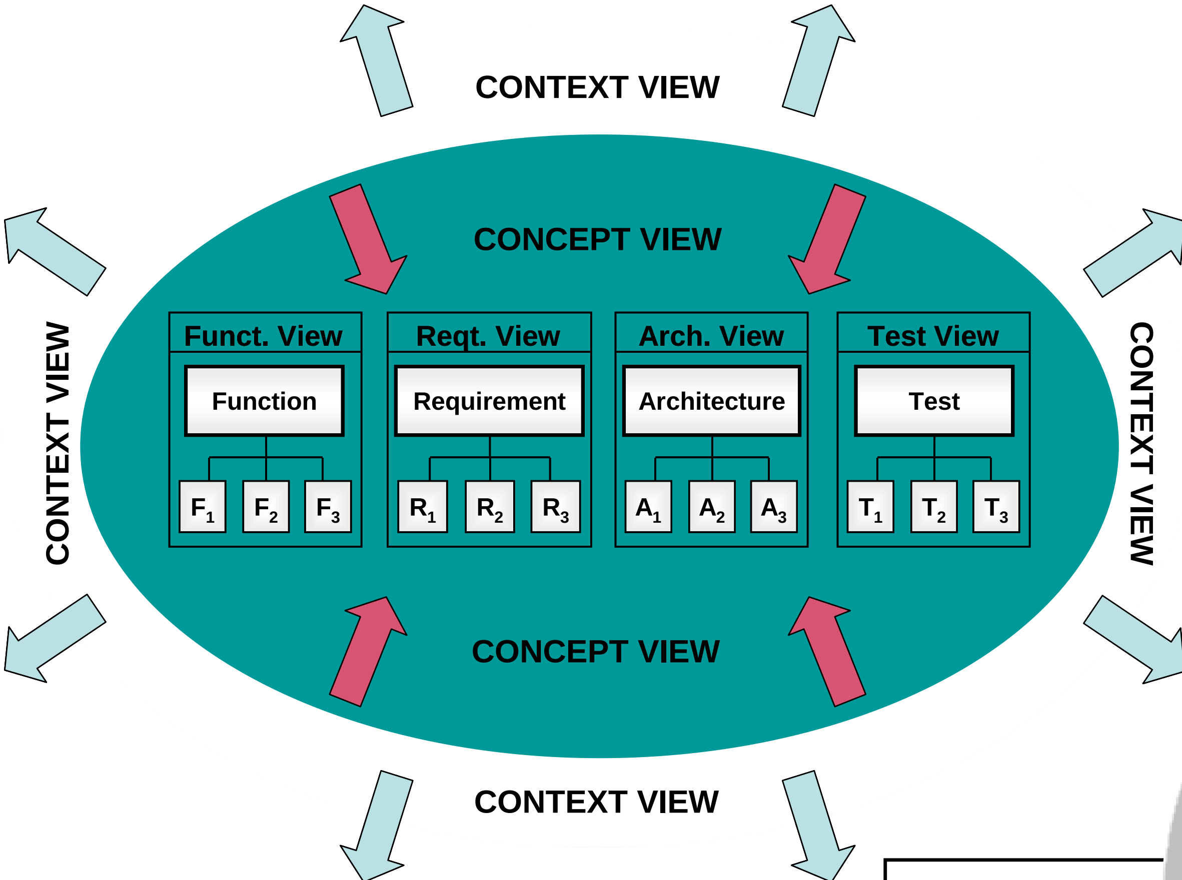


Frame Context Relationships

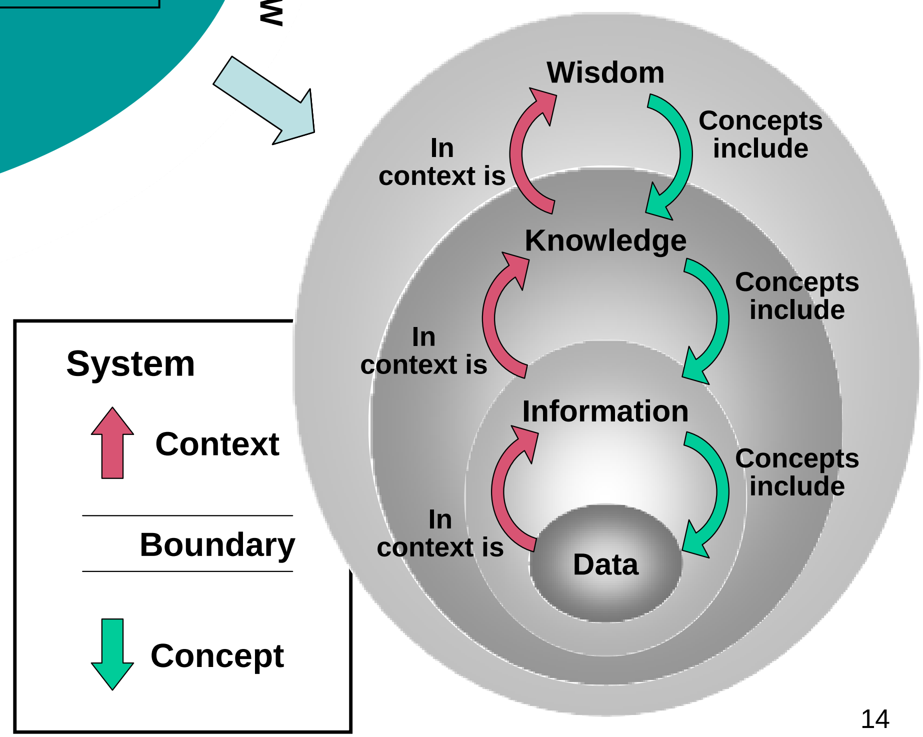
Coupling SE Process Modes through Knowledge



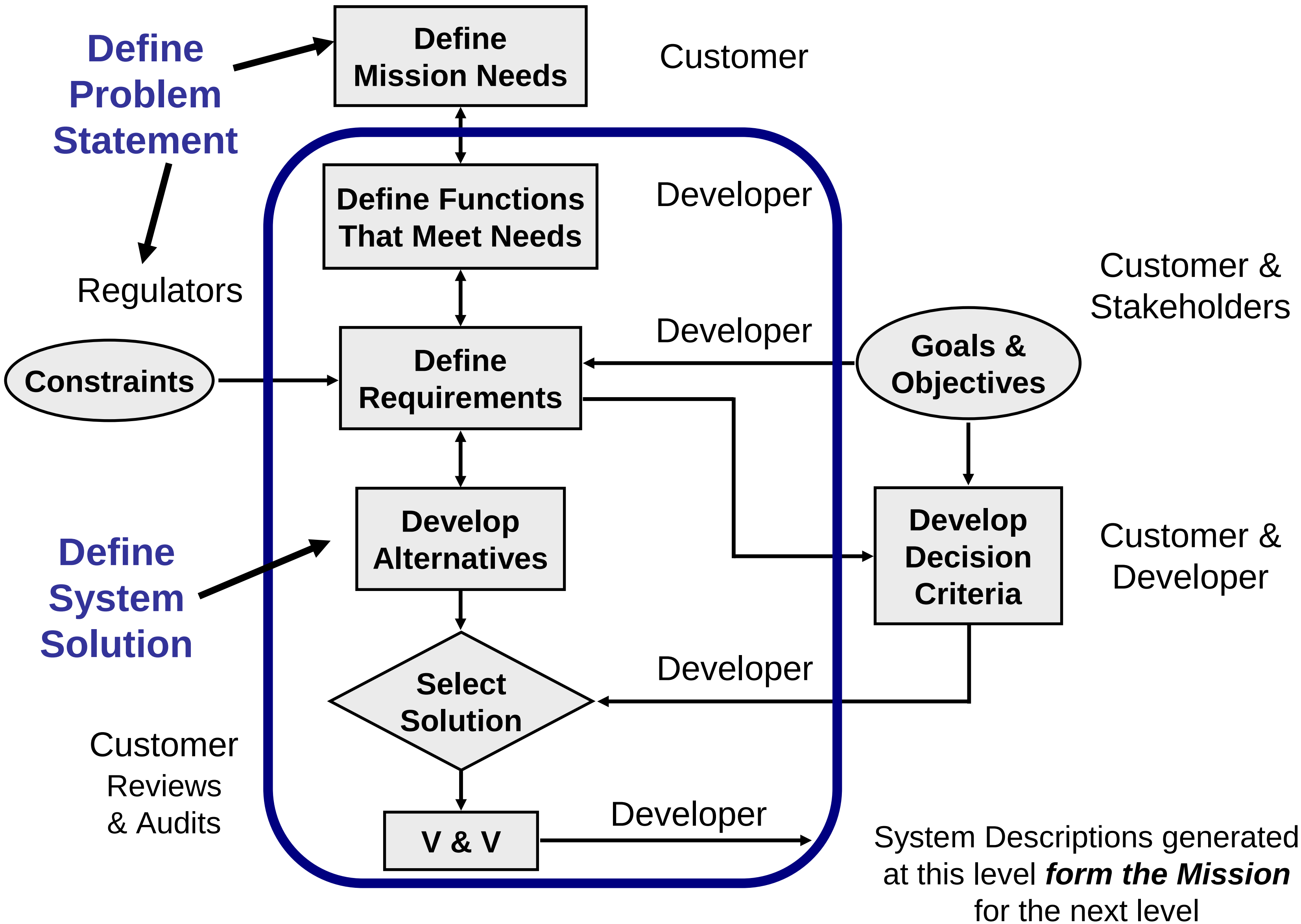
CCFRAT



Example

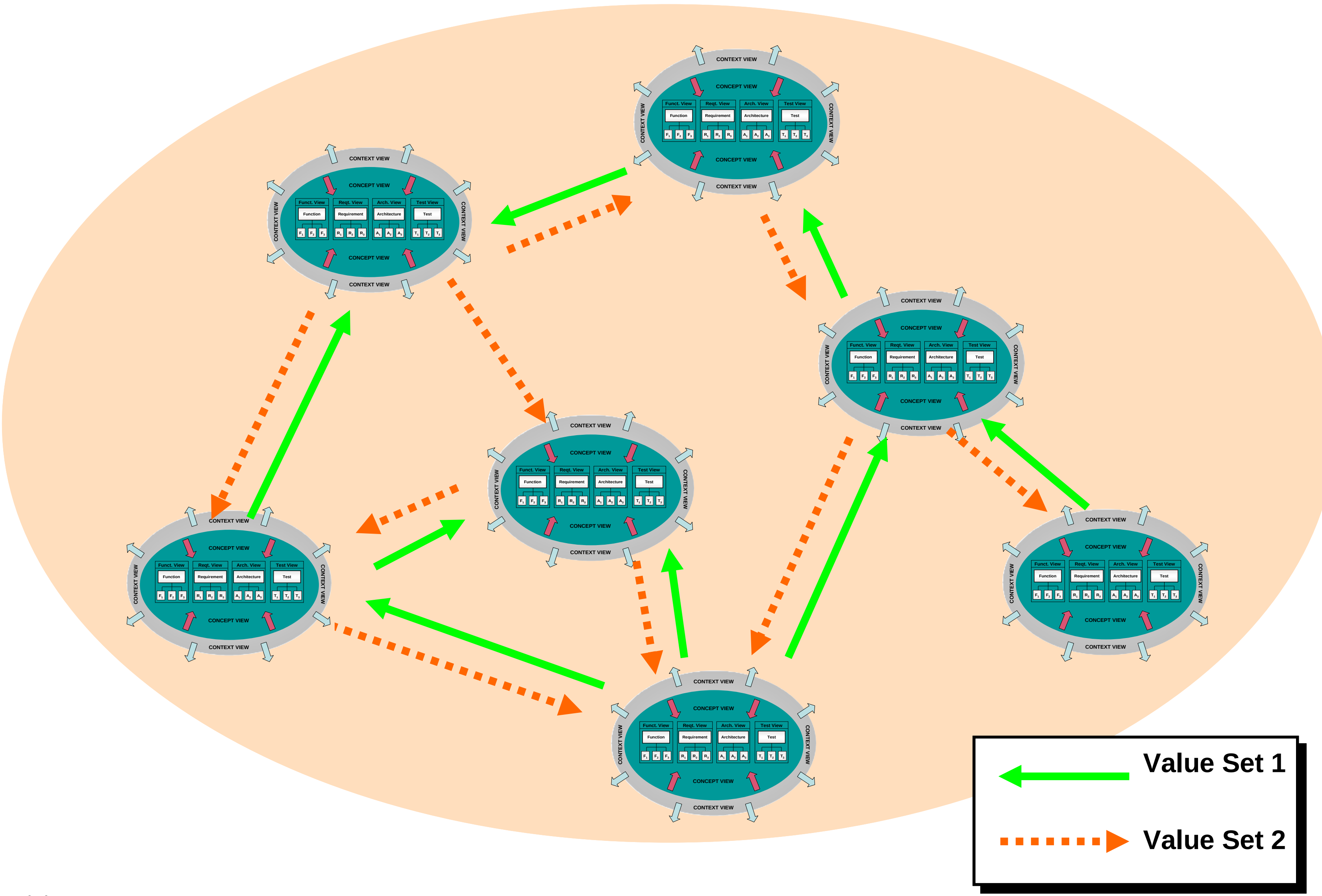


The FRAT Design Engine



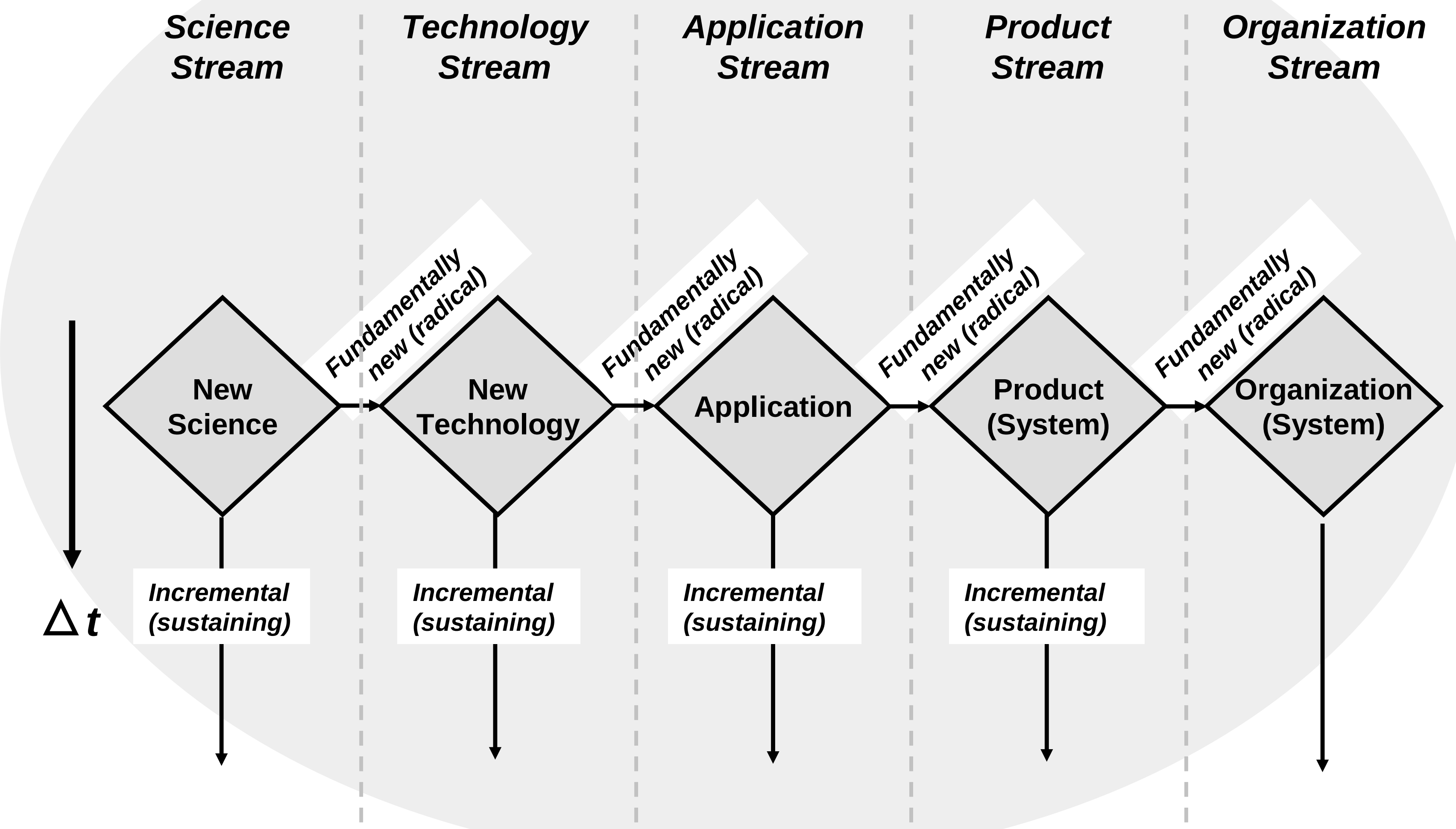
From "FRAT – A Basic Framework for Systems Engineering," Mar & Morais

Value Network in the Context View



Value "Streams of Change"

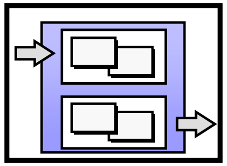
How the Problem Changes in the System Environment

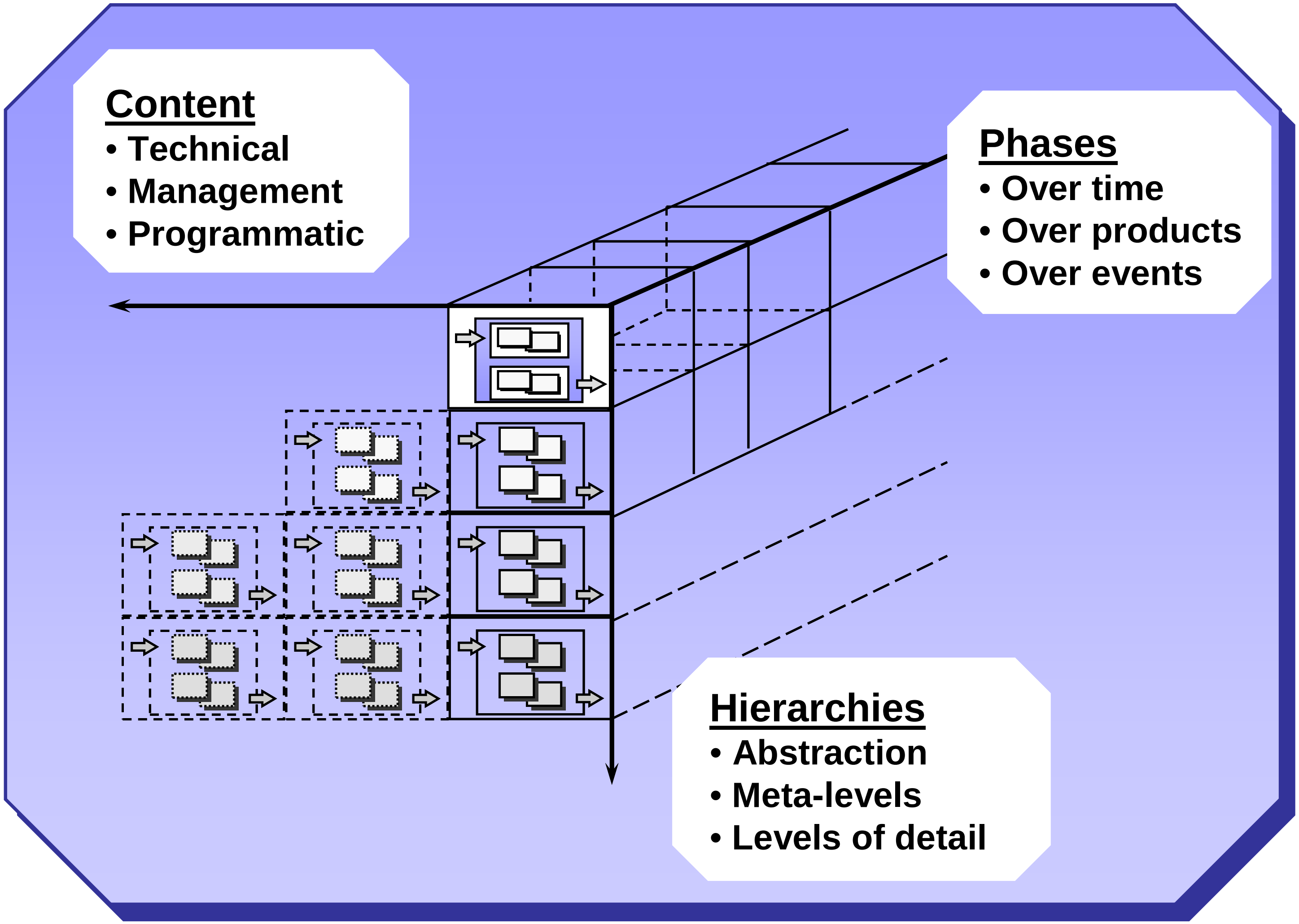


* Simpson, Joseph J., "Innovation and Technology Management," INCOSE 12th Annual International Symposium, Las Vegas, NV, 2002

CCFRAT Approach

Phases, Hierarchies, Content





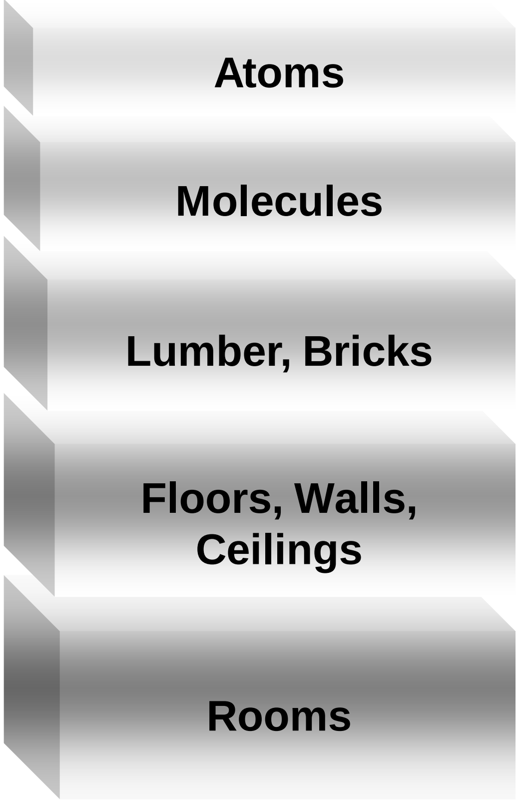
Pick One Aspect from Each Axis

‘Abstraction Stacks’

A House Consists of:

*“... house consists of several thousand pounds of carbon, some complex polymers, about 5000 bricks, two bathrooms, and 13 ceilings...”

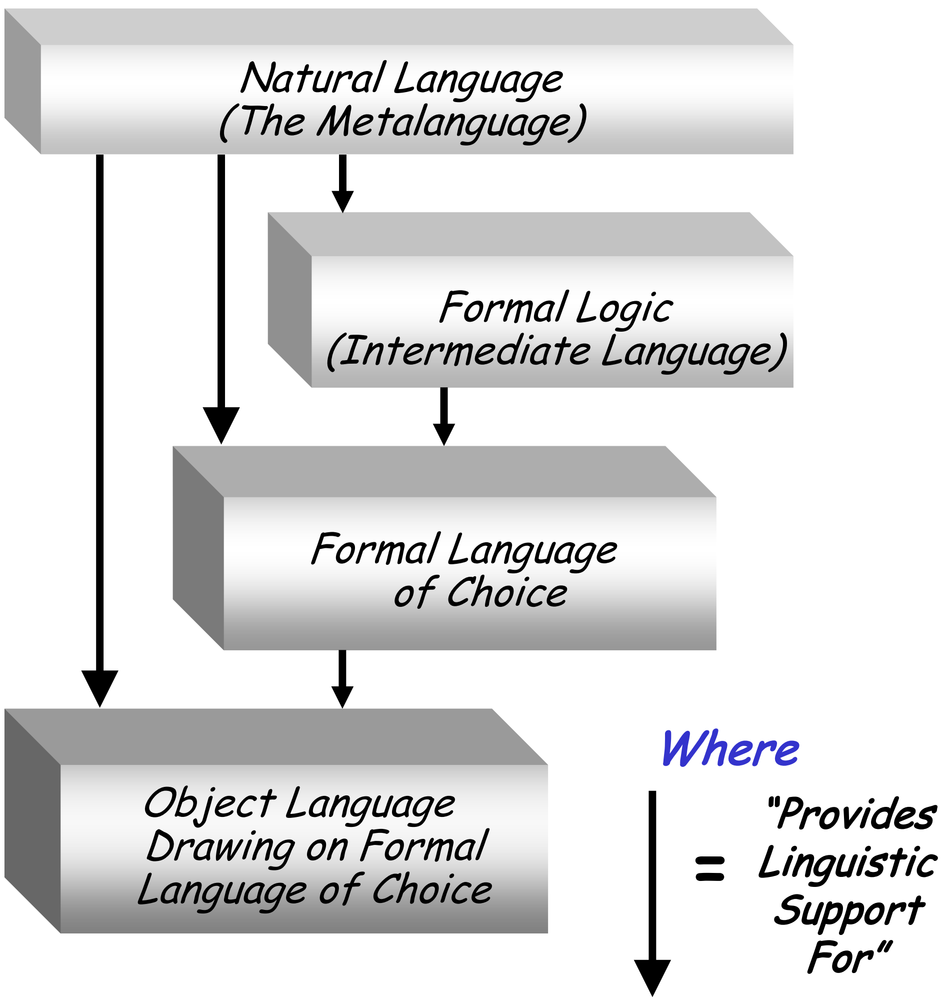
Use of Abstraction ‘Stacks’



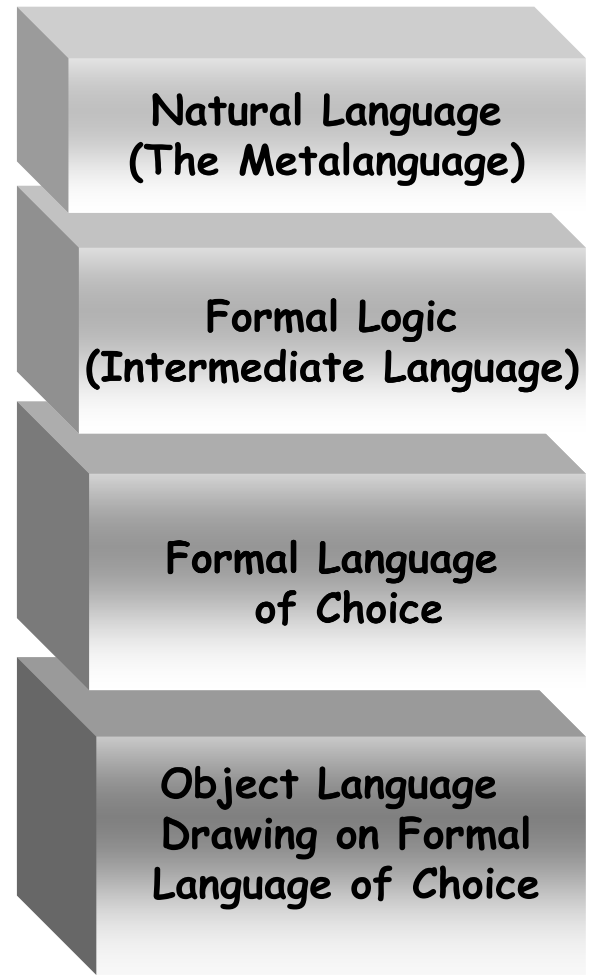
* From Chapter 12, What do Classes Represent?, *The C++ Programming Language, 2nd Edition,* Bjarne Stroustrup, 1991.

Warfield, S&T Language Design Capability

Initial Inter-Relationships Defined: "Pattern of Infrastructure"

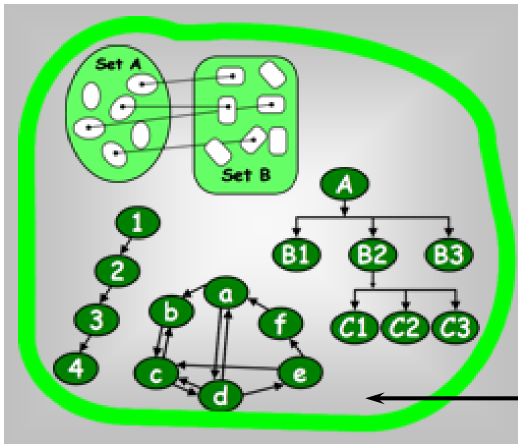


In Terms of an Abstraction Stack

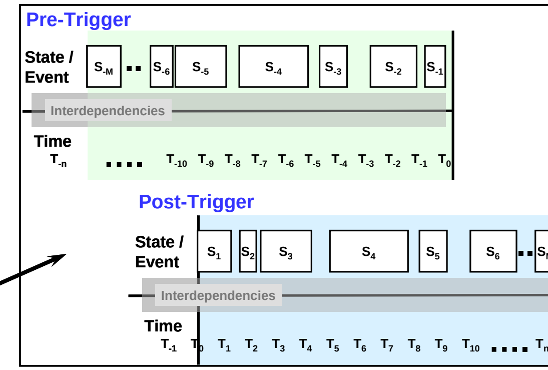


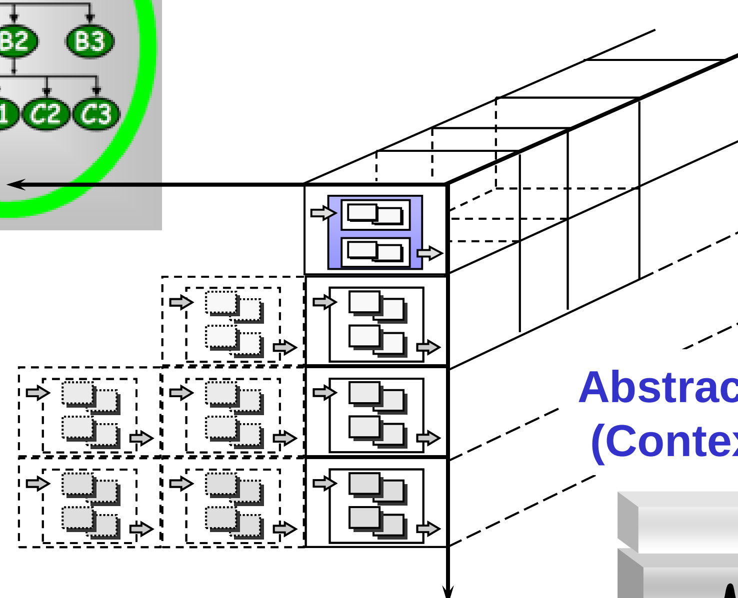
Expanded CCFRAT Approach

Relationships (Mapping/Order/Constraint Logic)

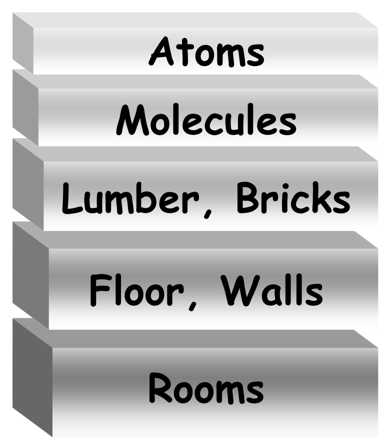


Process Mechanisms (Precedence/Timing Logic)





Abstraction 'Stacks' (Contextual Logic)



Meta-Levels

Systems Meta-Levels

	Being's Meta-levels	Bateson's Series	Modalities of Being-in-the-World	Associated Cognitive Abilities	Systems Meta-levels
5	Being meta-level 5 ULTRA	This step into non-Being is ultimately unthinkable	Empty Handedness Emptiness or Void	Cognitive Inability	Rules For Developing Rules
4	Being meta-level 4 WILD	Learning ^{4} to learn to learn to learn	Out-of-Hand	Encompassing	Rules For Developing Frameworks
3	Being meta-level 3 HYPER	Learning ^{3} to learn to learn	In-Hand	Bearing	Architectural Frameworks
2	Being meta-level 2 PROCESS	Learning ^{2} to learn	Ready-to-Hand	Grasping	Architectural System Schema
1	Being meta-level 1 PURE	Learning ^{1} as an ideal gloss	Present-at-Hand	Pointing	Conceptual System Schema
0	Being meta-level 0 ENTITY	Concrete Instances ^{0} of learning in world	Orientation toward Things	Thing	Single Physical Instance

- Table from Palmer, Kent D., "Meta-systems Engineering,"
- 10th Annual Symposium of INCOSE, 2000

Basis for SE Language

A formal systems engineering language will greatly reduce complexity in the practice of systems engineering, as well as facilitate the solution of even more complex problems. The initial basis for formal SE language development includes:

- **Robust set of system meta-levels**
- **Clear set of conceptual transforms to reduce system complexity**
- **Applicable combination of CCFRAT methods**
- **Well-defined system meta-level heuristics**

A **formalized set** of system meta-levels and meta-level transforms must be created in a structured fashion to support the development of a systems engineering language.

Summary and Conclusions

Increasing system and environmental complexity can be measured and managed.

Systems engineering processes and principles provide a logical framework for evaluation of system complexity.

As product systems grow in size and complexity, system engineering must find and utilize the proper abstraction frame which reduces the system complexity and retains the proper level of system analysis.

The CCFRAT concepts and methods combined with well-defined meta-levels provide a foundation for a specialized systems engineering language.

Questions?