Complexity Reduction

A Pragmatic, Semantic Approach

Empowering Collaborative Intelligence

Joseph J Simpson
Mary J Simpson
System Concepts LLC
http://www.systemsconcept.org

August 17th, 2011

Overview

Introduction and Definitions

Scientific Foundations

- Scientific Method (Abduction, Induction, Deduction)
- Types of Truth
- Types of Systems
- Types of Complexity

Historical Systems Engineering (SE) Practices

- SE Method, N Squared Chart (N^2C)
- SE Method, Design Structure Matrix (DSM)
- SE Method, Interpretive Structural Modeling (ISM)

Recent Developments

- Abstract Relation Type (ART)
- ART for N^2C
- ART for DSM
- Concept Cube SM

Summary and Questions

Introduction

- Primary goal is to enable group learning
- Learning and discovery reduce cognitive complexity
- Groups are collections of intelligent agents both human and non-human
- Some systems engineering (SE) techniques support structured group learning for humans
- Other SE techniques include the use of intelligent agents
- Evolutionary programming is used to add intelligent agents to classical SE techniques

Scientific

Foundations

Scientific Method

Inference

Reduces uncertainty and develops clear objective views of the world

Abduction

Generates theories, conjectures, hypotheses and explanations that have not yet been verified by induction or deduction

Induction

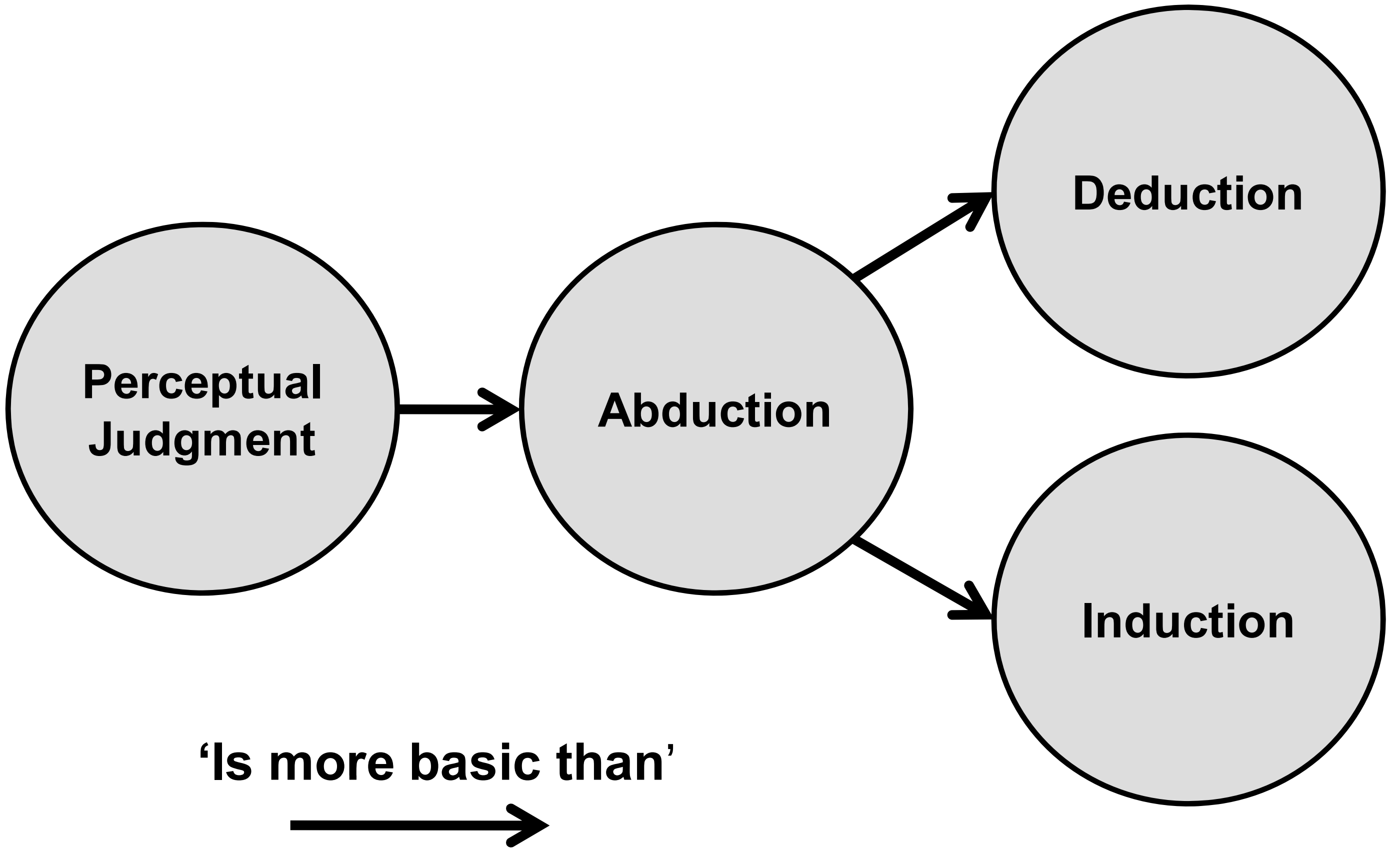
Generates conclusions based on observations or experimental data

Deduction

Process of formal, mathematical reasoning that produces a conclusion based on a set of assumptions that are given as true

C.S. Peirce, Four Types of Inference

“Every perceptual judgment shades into an abductive inference or hypothesis, and further elucidation of the meaning must involve phases corresponding to deduction and induction.”*



*Manley Thompson, *The Pragmatic Philosophy of C S. Peirce*, 1953, page 250.

Figure adapted from Chapter 2 Universal Priors to Science, *A Science of Generic Design, Managing Complexity through Systems Design*, John N Warfield, 2003.

Peirce, Scope of Inference Structure

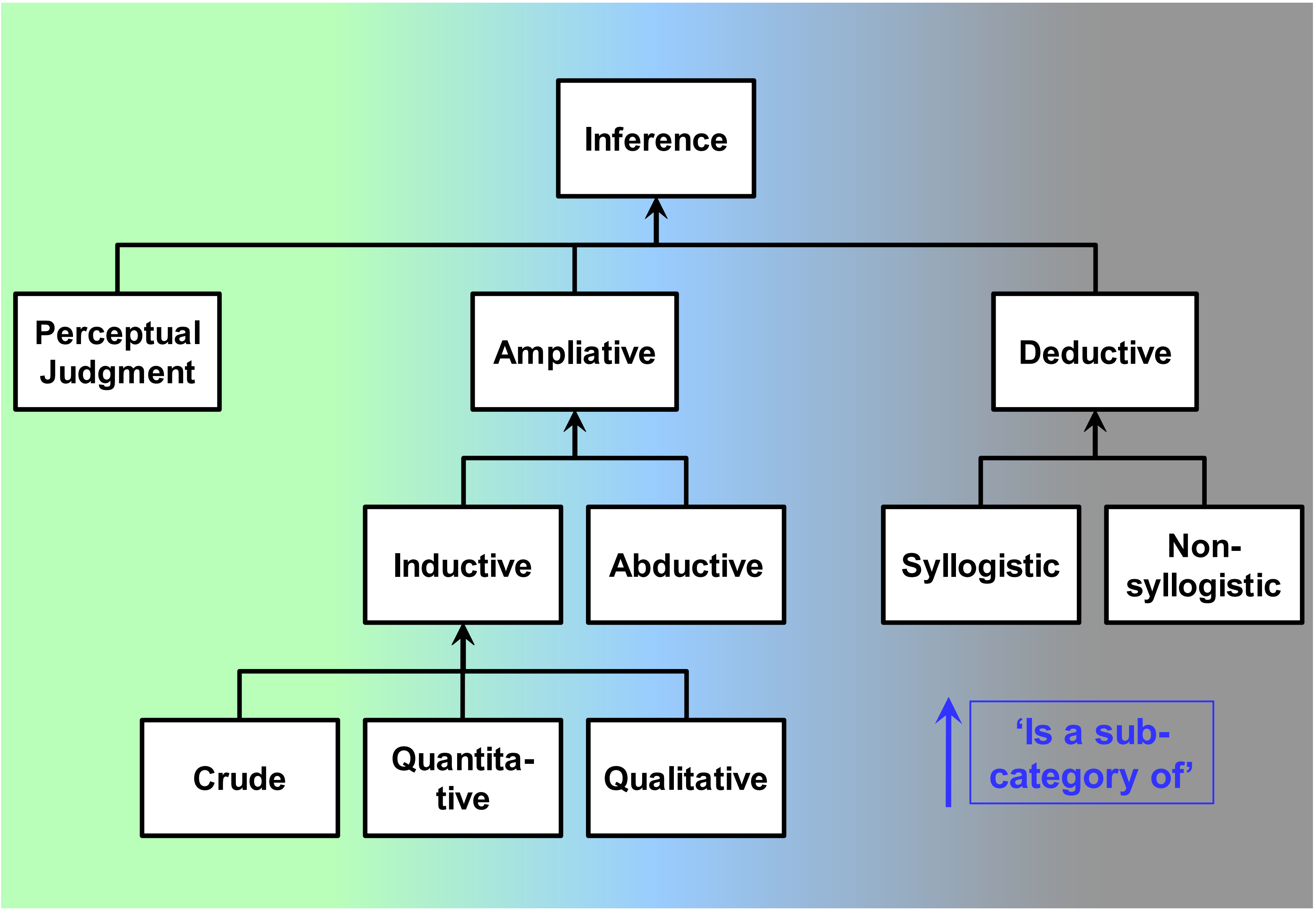


Figure adapted from Chapter 2 Universal Priors to Science, *A Science of Generic Design, Managing Complexity through Systems Design*, John N Warfield, 2003.

Scientific Method & Model Building, AD Hall

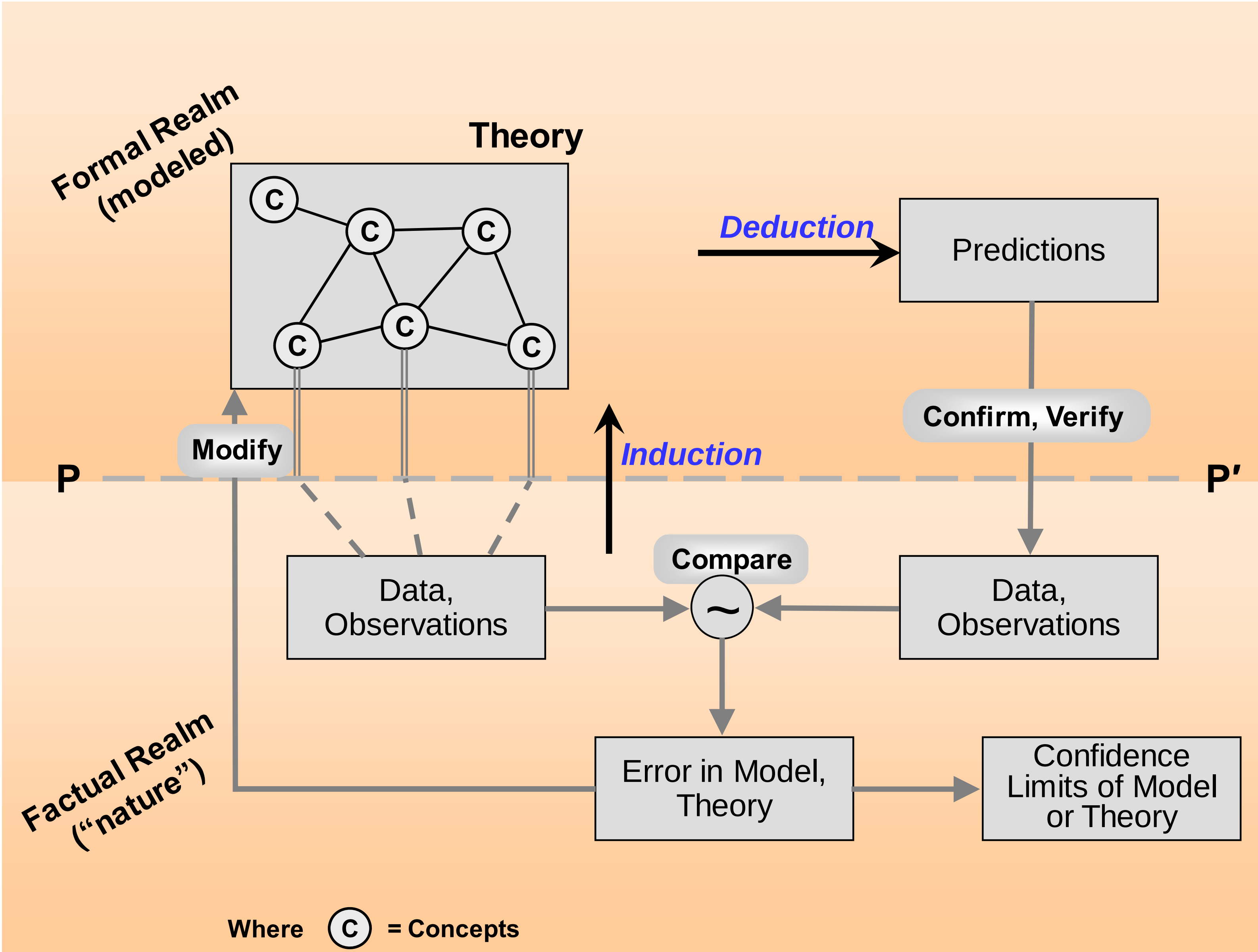


Figure adapted from Chapter 4 The Basis of Systems Methodology in Scientific Method, *Metasystems Methodology*, Arthur D. Hall, 1989.

Scientific Method-Evolutionary Computation, Fogel

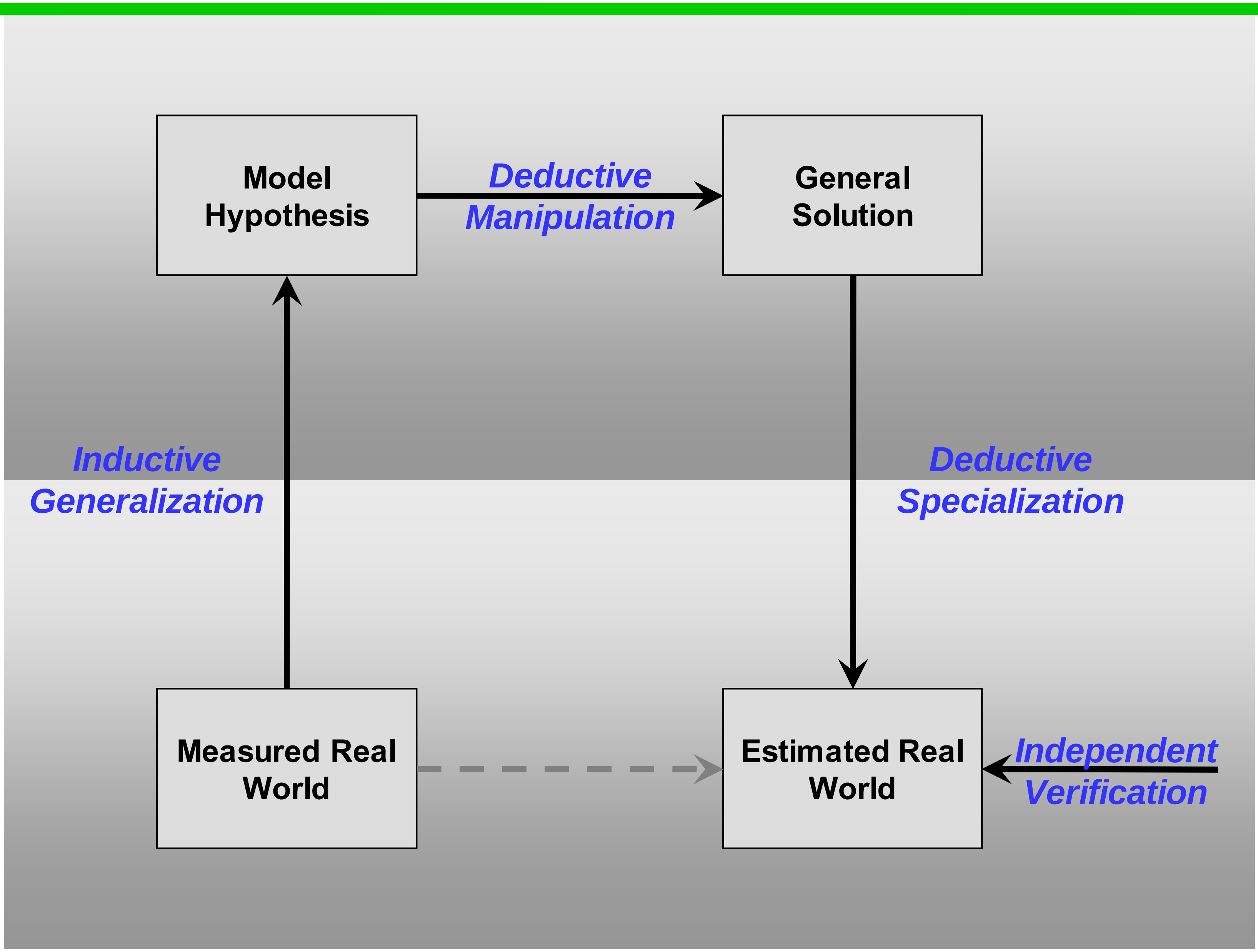


Figure adapted from Chapter 1 Defining Artificial Intelligence, *Evolutionary Computation, Toward a New Philosophy of Machine Intelligence*, David B. Fogel, 2006.

Types of Truth

Three types of statements (or sentences) are considered in this discussion: formal, factual and value.

Formal

Truth of formal sentences depends only on the form of the logical connectives in the sentence.

Factual

Truth of factual sentences depends on the state of the real world, and how well the synthetic statement describes the real world.

Value

Truth of a value sentence is determined in a number of ways, including the methods applied to formal and factual sentences.

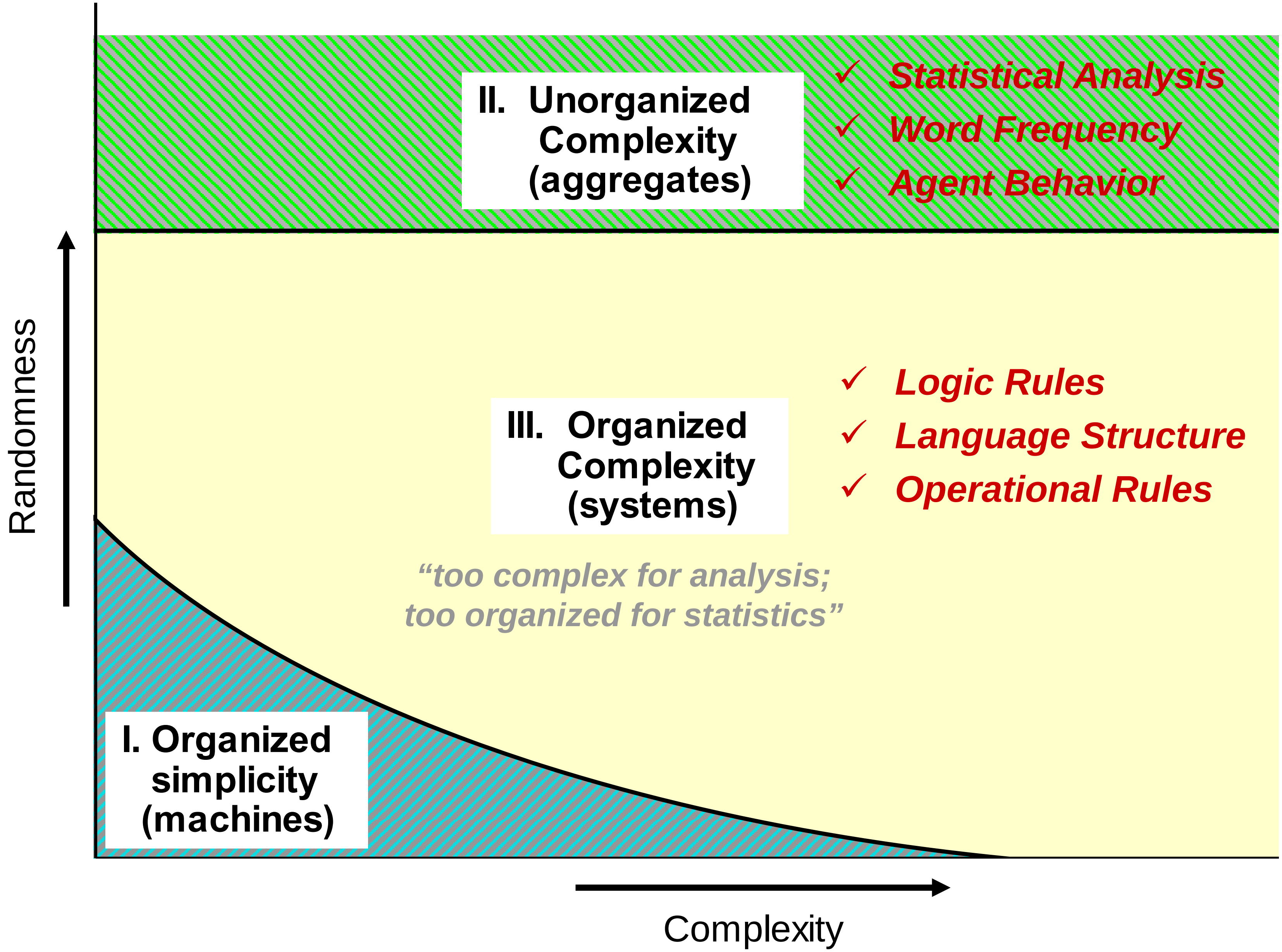
Value sentence truth expands to ethics, culture, & prediction.

Types of Truth

	Formal	Factual	Value
Examples **(a)** **(b)** **(c)**	A is B A is not B If A then B	John is a man. John is not a man. All swans are white.	John is a good man. John ought to pay five dollars. John values gold.
These sentences tell us:	What is possible or impossible	What is true or false in the world.	What ought or ought not to be.
Evidence relevant to their truth:	Meaning of words, esp. "is", "not", "and", "if then".	Meanings and observations.	Meanings, observations and value/ethical systems.
Known by experience?	No	Yes	Some types yes, some types no.
Are they certain?	Yes	No	No
Required to validate inferences:	A proof is necessary & sufficient.	Observational data and predictive success.	No one kind of argument; many desired things.

Adapted from "Metasystems Methodology" A.D. Hall. III

Types of Systems



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

Types of Complexity

Complexity is defined as the measure of the difficulty, effort and/or resources required for one system to effectively observe, communicate and/or interoperate with, another system.

Range of Complexity Types

Cognitive/ Perceptual ••• Behavioral ••• Organic ••• Computational

[Everything Else]

Types of Complexity

Cognitive Complexity

"… is that sensation experienced in the human mind when, in observing or considering a system, frustration arises from lack of comprehension of what is being explored." [Warfield]

Relativistic View of Complexity

"… the complexity of a given system is *always* determined by some other system with which the given system operates." [Casti]

Computational Complexity

Computational complexity is generally associated with well defined algorithmic problems and the efficient solutions for these stated algorithmic problems.

Observations re. Types of Complexity

Human Cognitive Complexity	Computer Computational Complexity
Communication in common context (Law 1)	Communication with common syntax (Symbols)
Can communicate with nonsense symbols (Law 2)	Cannot communicate with nonsense symbols
Limited by short-term memory recall	Limited by computation time and memory size
Three types of language components: • Prose • Mathematics • Graphics	Single language component: • Symbols (0,1) (binary logic)

Historical

Systems Engineering (SE)

Practices

SE Method: N-Squared Charts

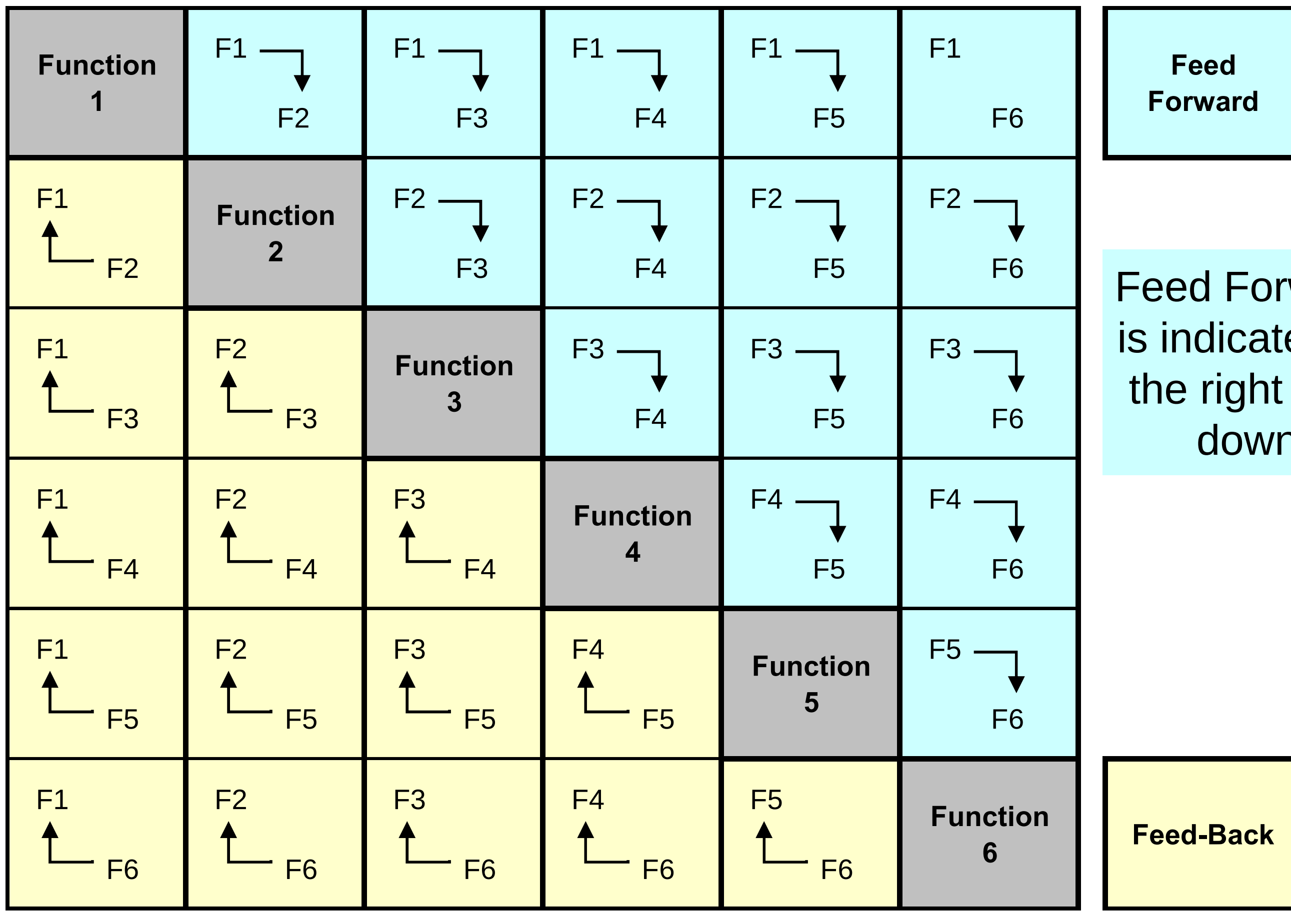
Initially developed as a systems analysis and interface communication tool for software design and development.

- Proved to be well suited for the grouping of alternative system configurations
- Mostly used to communicate system structure
- Based on analysis by human experts
  - Analyze and define interfaces
  - Present dependency, interdependency and sequence
  - Evaluate clustering and parallelism
  - Address interrelationships inherently without bias
- Recognize interface patterns and signatures (functionally-bound blocks, nodes, supported nodes, disjoints, etc.) thru use of human pattern recognition

SE Method: N-Squared Charts

Functions, elements are placed on the diagonal

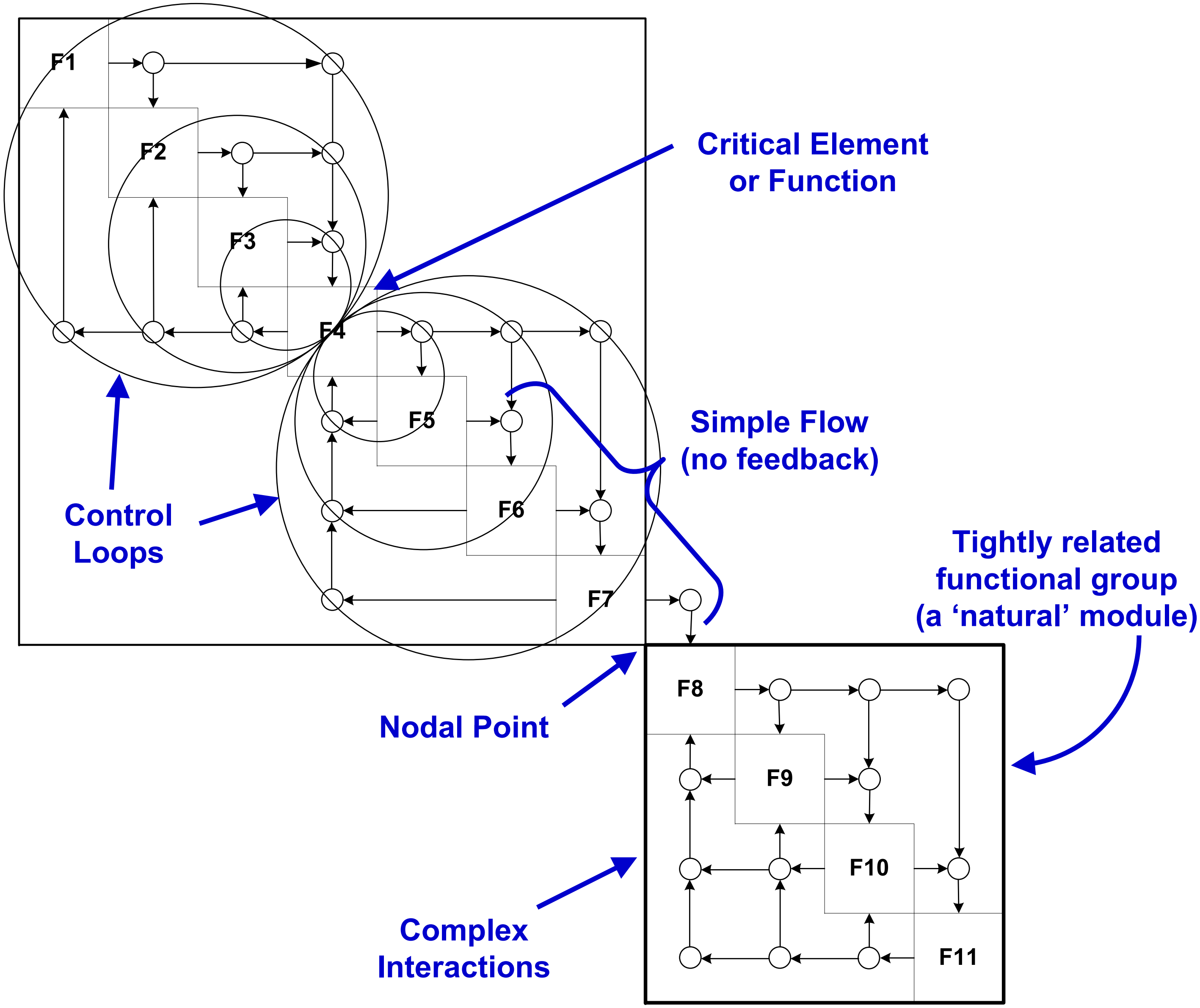
Feed-Back is indicated to the left and up



Feed Forward is indicated to the right and down

Interactions or interdependencies between functions, elements are indicated in the remaining squares

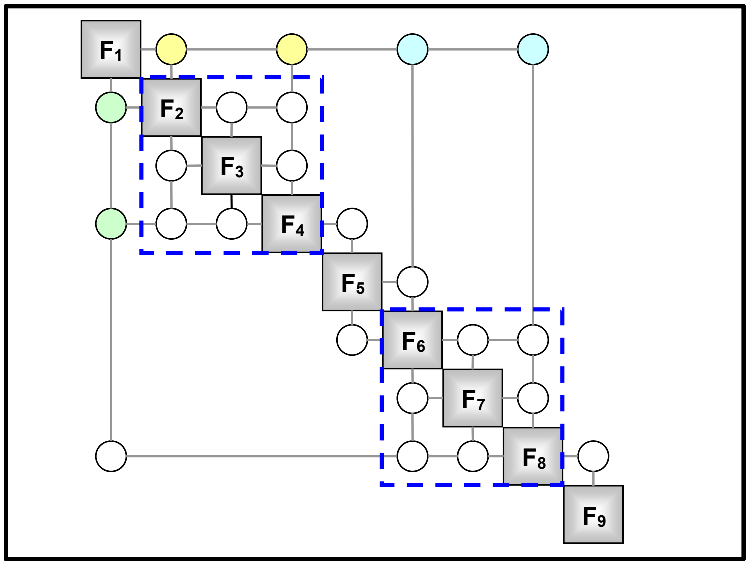
SE Method: N-Squared Charts



Adapted from "The N2 Chart, R. Lano, Copyright 1977, TRW Inc.

SE Method: N-Squared Charts

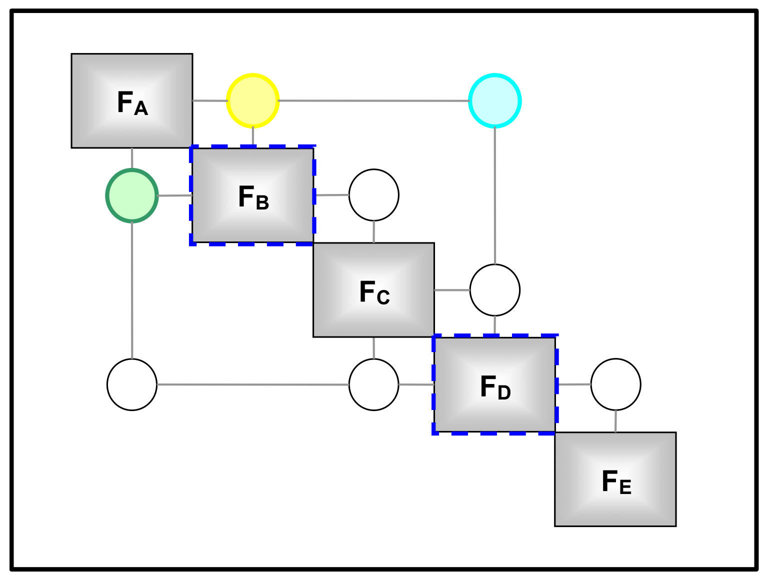
Nine Functions and Twenty-Three Nodes



Five Functions and Eight Nodes

Where:

$F_A = F_1$
$F_B = F_2$, F_3 AND F_4
$F_C = F_5$
$F_D = F_6$, F_7 AND F_8
$F_E = F_9$



SE Method: Design Structure Matrix (DSM)

First developed to address the reduction of computational complexity in sets of linear equations.

- Recognized as a powerful graphical tool to communicate large amounts of detailed information to groups of diverse individuals
- Developed to facilitate the solution of large-scale systems problems using graphical communication techniques coupled with a set of computational rules and techniques
- Designed to identify system clusters and the highest value system configurations
- Has developed into a number of different, varying techniques each with unique syntax and semantics

Identification of feasible, alternative system configurations has proven highly complex, and remains the highest barrier to effective implementation

SE Method: Directional DSM

Items are placed on the diagonal

Rows with no 'local' interaction are allowed

Item1					
X	**Item2**				
X	**X**	**Item3**			
	X		**Item4**		
		X		**Item5**	
X	**X**	**X**	**X**	**X**	**Item6**

"Feed Forward" is **maximized**

Direction differs from one technique to another

Output

"Feed-Back" is **minimized**

Direction differs from one technique to another

Input

Interactions or interdependencies between items are indicated in the remaining squares

SE Method: Balanced DSM

Items are placed on the diagonal

Rows with no 'local' interaction are allowed

Item1					
X	Item2	X	X		
	X	Item3	X		
	X	X	Item4	X	X
			X	Item5	X
			X	X	Item6

Links are balanced

Links are balanced

Interactions or interdependencies between items are indicated in the remaining squares

SE Method: Interpretive Structural Modeling

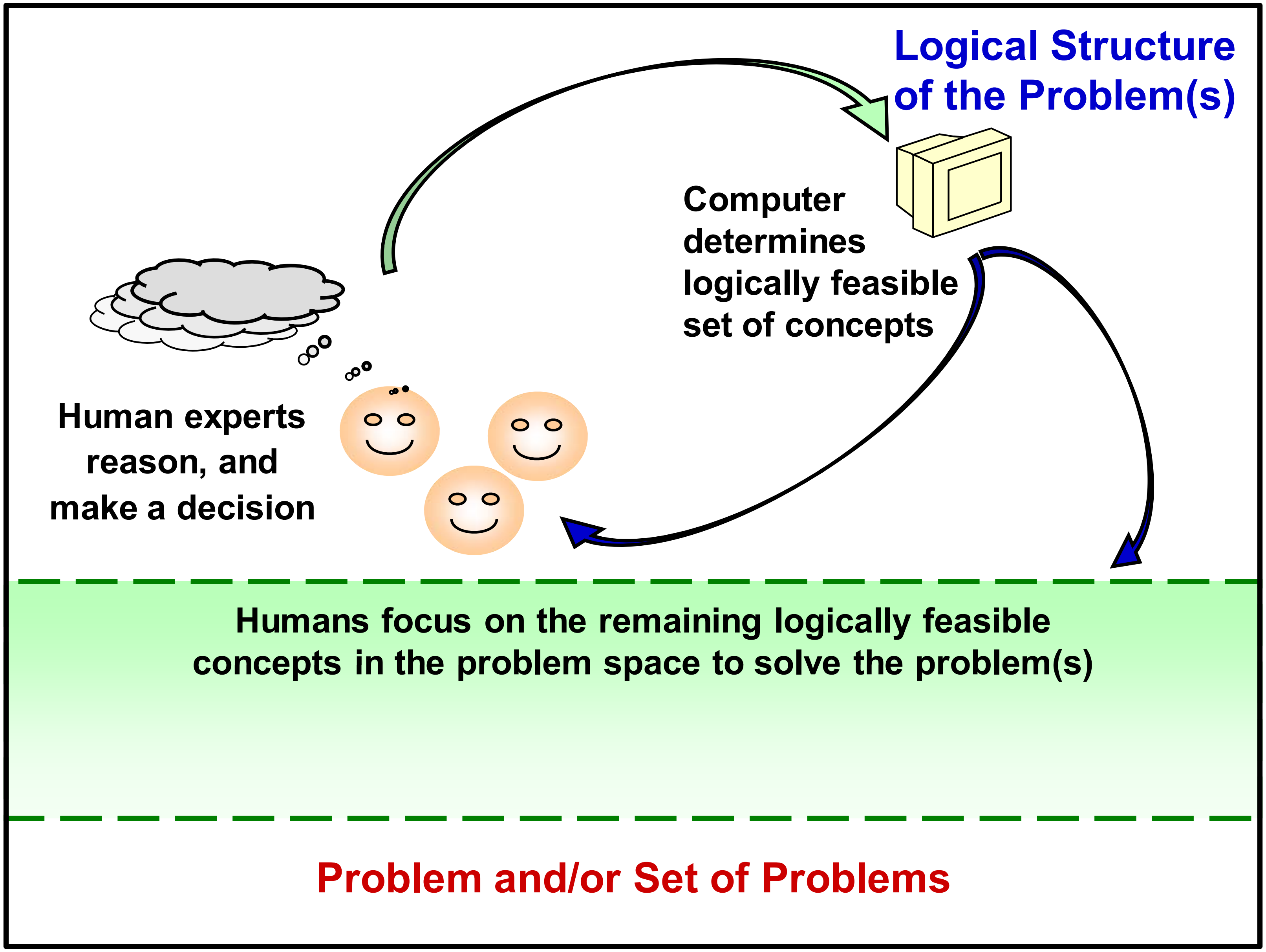
Developed by John N Warfield, to address the reduction of cognitive complexity found in socio-technical systems. Warfield's science encompasses:

- **Behavioral pathologies** of individuals, groups, organizations
- **Logic and language** (formal logic, mathematics of structure, structural modeling, and languages)
- **Collaborative abduction** (generation of structural hypotheses; group abduction or hypothesis generation in a group process)

Adds intelligent computer-based agent to process

Applies to ideas and concept development

SE Method: ISM

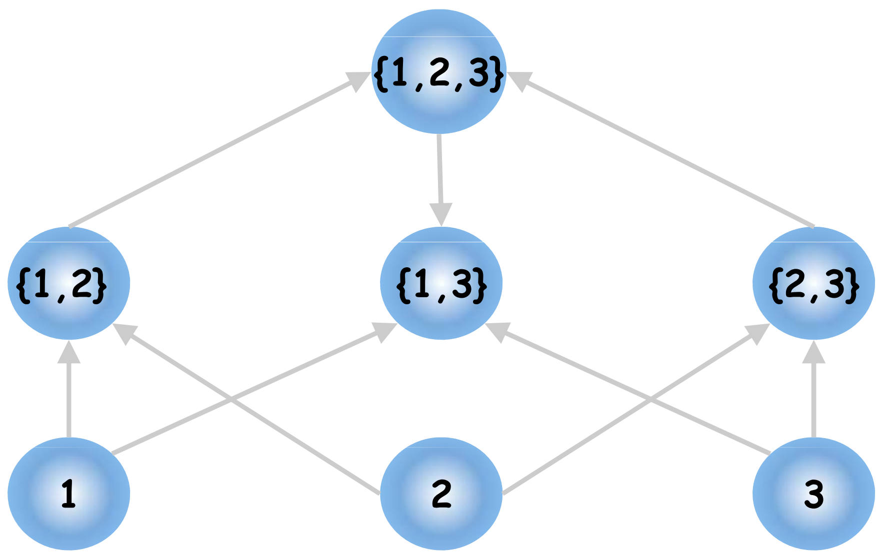


SE Method: ISM, Law of Triadic Compatibility

Quantifies the limitations of short-term memory as they relate to human decision making

The human mind can recall and operate with seven concepts:

- Three elements
- The four combinations associated with three elements



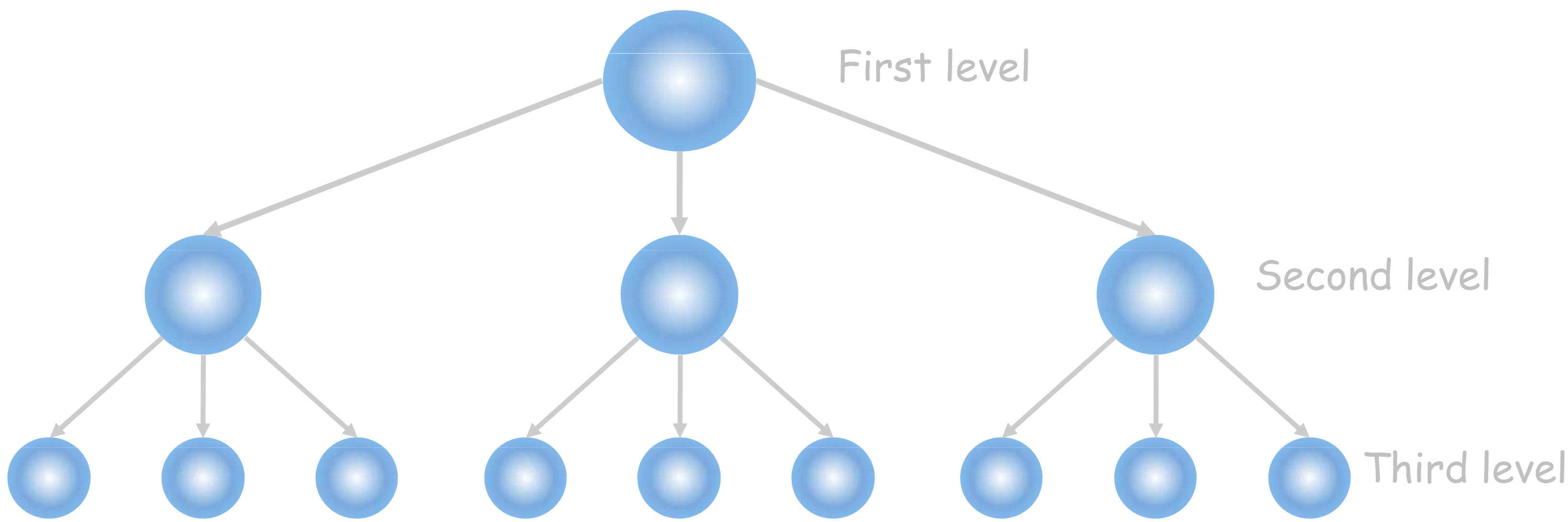
The human mind is compatible with the need to explore interactions among a set of three elements

Capacity cannot be presumed for a set that both has four members, and for which those members interact

SE Method: ISM, Principle of Division by Threes

Iterative division of a concept as a means of analysis is mind compatible if each division produces at most three components, thereby creating a ‘tree’ with

- One element at the top
- At most three elements at the second level
- At most nine elements at the third level
- And so on…



Recent Developments

Abstract Relation Type (ART)

Based on the Abstract Data Type concept

- Similar structure
- Similar behavior
- Programming language independent

Abstract Relation Types

- Focused on global system organizing relationship
- Similar structure
- Similar semantics
- Specific system type independent

Used as basic components of a pattern language

Abstract Relation Type (ART)

Has three main parts:

Prose Description (text, words)

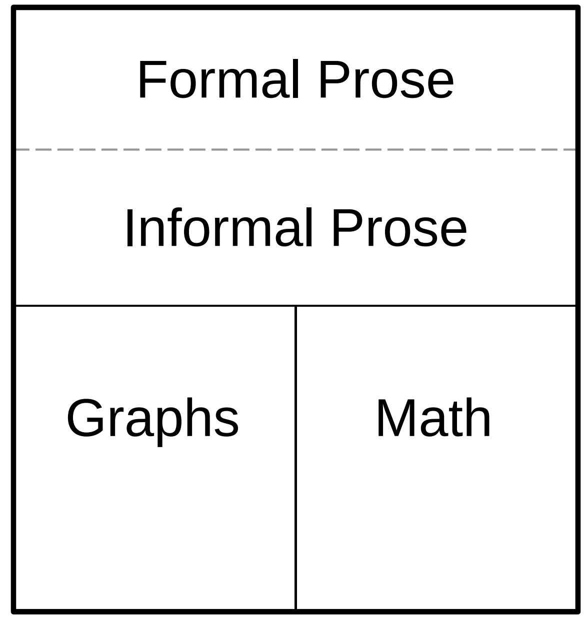
- Formal pattern
- Informal prose

Graphic Representation (directed graphs)

- Must have formal graphs
- Can also have informal graphs

Mathematics & Computer Representation

- Math equations
- Computer codes
- One or both



Abstract Relation Type, Setting the Context

Focus on system organizing relationship

- Natural language **relationship** (meta-language)
  Relationship types
  - Definitive
  - Competitive
  - Influence
  - Temporal
  - Spatial
  - Mathematical
- Mathematical **relation** (object language)
  Relation types
  - Universal
  - Reflexive
  - Transitive
  - Symmetric
  - Hybrid

Abstract Relation Type (ART) Example

Marking Space

Represents system structure

Value Space

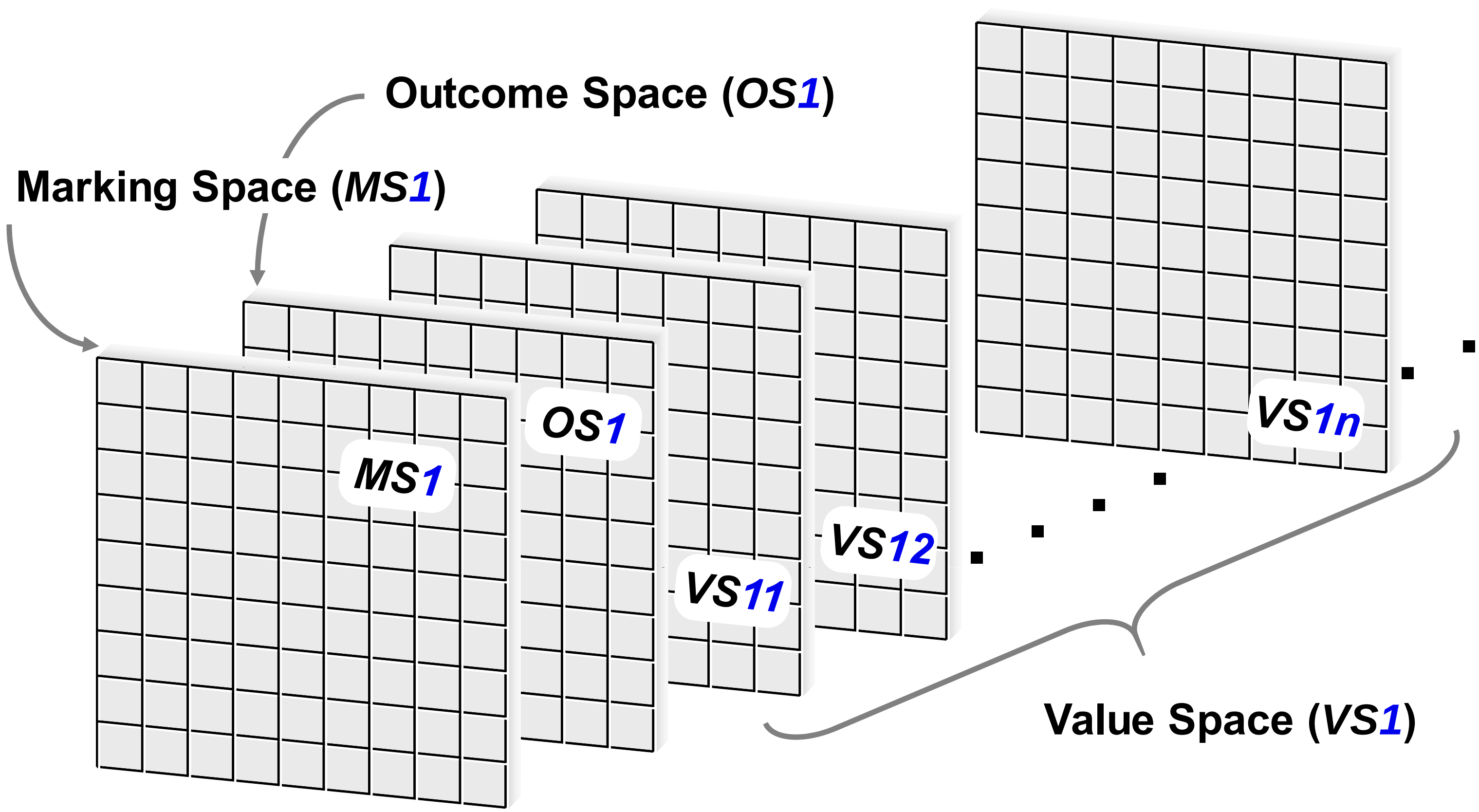
Represents system value configuration(s)

Outcome Space

Represents aggregation of system structure and system value(s)

Evolutionary algorithms are used to search the Outcome Space to find the best-fit solution

Abstract Relation Type (ART)



Abstract Relation Type (*ART*) ≡ Ƒ [*MS, OS*]

Outcome Space (*OS*) ≡ Ƒ [VS_1, VS_2, …VS_n, VS_{n+1}, …]

Abstract Relation Type (ART), Example

Marking Space

	A	B	C	D	E	F
A	0	0	0	0	0	1
B	0	0	0	1	0	0
C	1	1	0	0	0	1
D	0	0	1	0	0	0
E	0	1	0	0	0	0
F	0	1	0	0	1	0

Represents the System Structure [for each object A through F, a one in the cell shows location of an interface]

X

Value Matrix

0	1	2	3	4	5
1	0	1	2	3	4
2	1	0	1	2	3
3	2	1	0	1	2
4	3	2	1	0	1
5	4	3	2	1	0

Number Field reflects the same properties as the organizing relationship

=

Outcome Space

0	0	0	0	0	5
0	0	0	2	0	0
2	1	0	0	0	3
0	0	1	0	0	0
0	3	0	0	0	0
0	4	0	0	1	0

Reflects the relative value of the system structure

ART for N^2C

0	0	1	0	0	0	1	0	1
0	0	0	1	0	1	0	0	0
0	0	0	0	1	0	0	1	0
0	1	0	0	0	1	0	0	1
0	0	1	0	0	0	0	1	0
0	1	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0
1	0	1	1	0	0	0	0	0

No Obvious Pattern; Unordered

Ordered; Obvious Patterns

0	1	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0
0	0	1	1	0	1	0	0	0
0	0	1	0	1	0	1	0	0
0	0	0	0	0	1	0	1	1
0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	1	0

ART for N^{2}C, Evolutionary Algorithm Applied

System Concepts[SM]

0	1	2	3	4	5	6	7	8
1	0	1	2	3	4	5	6	7
2	1	0	1	2	3	4	5	6
3	2	1	0	1	2	3	4	5
4	3	2	1	0	1	2	3	4
5	4	3	2	1	0	1	2	3
6	5	4	3	2	1	0	1	2
7	6	5	4	3	2	1	0	1
8	7	6	5	4	3	2	1	0

0	0	1	0	0	0	1	0	1
0	0	0	1	0	1	0	0	0
0	0	0	0	1	0	0	1	0
0	1	0	0	0	1	0	0	1
0	0	1	0	0	0	0	1	0
0	1	0	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0
1	0	1	1	0	0	0	0	0

Unordered Marking Space; No Obvious Pattern

0	1	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0
0	0	1	1	0	1	0	0	0
0	0	1	0	1	0	1	0	0
0	0	0	0	0	1	0	1	1
0	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	1	0

Ordered Marking Space; Obvious Patterns

ART for DSM

- System structure is separated from system value
- Marking space is based on Warfield's Mathematics of Structure – augmented Boolean mathematics
- Value space is based on Hitchins' Automated N Squared chart
- Evolutionary algorithm ART method generates candidate solutions automatically, reducing the cost of human systems analysis and creating solutions not seen by the human analyst

ART for DSM

Initial ART Marking Space

0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	1
0	0	0	0	0	1	0	1	0	0	1	0
0	1	0	0	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	1	0	0	0	1	0	0
0	1	1	0	0	1	0	0	0	0	1	1
0	1	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	1	1	0

Final ART Marking Space

0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	1	0	0	0	0
1	1	0	1	1	0	1	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	1	1	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0	1	0
0	0	0	0	1	0	1	0	1	0	0	0
0	0	1	1	0	0	0	1	0	1	0	0
1	0	0	1	0	0	0	0	0	0	0	0

SE Method: Concept Cube SM

A Generic Concept Cube Approach

- Designed to organize any given concept so that human experts can recognize and evaluate a concept
- Adapts basic aspects of Formal Concept Analysis
- Incorporates Warfield's "Law of Triadic Compatibility" (LTC) and "Principle of Division by Threes" (PDT)
- Main concept assigned as Level 0 (L0)
- Three sub-concepts assigned as Level 1 (L1)
- Standard numbering scheme (level, parent, local):

L**0** -> C**0**.0

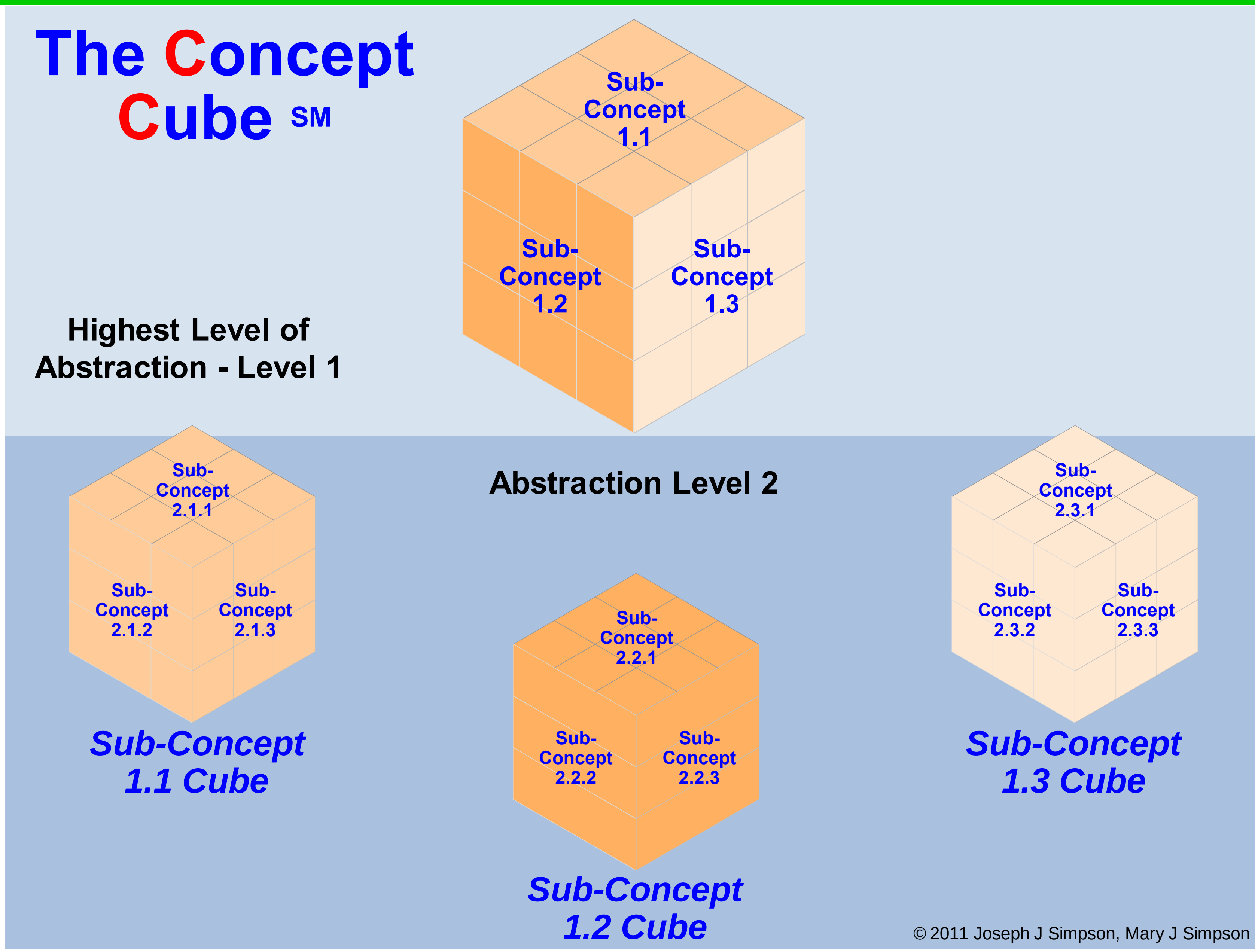
L**1** -> SC**1**.1; SC**1**.2; SC**1**.3

L**2** -> SC**2**.1.1; SC**2**.1.2; SC**2**.1.3

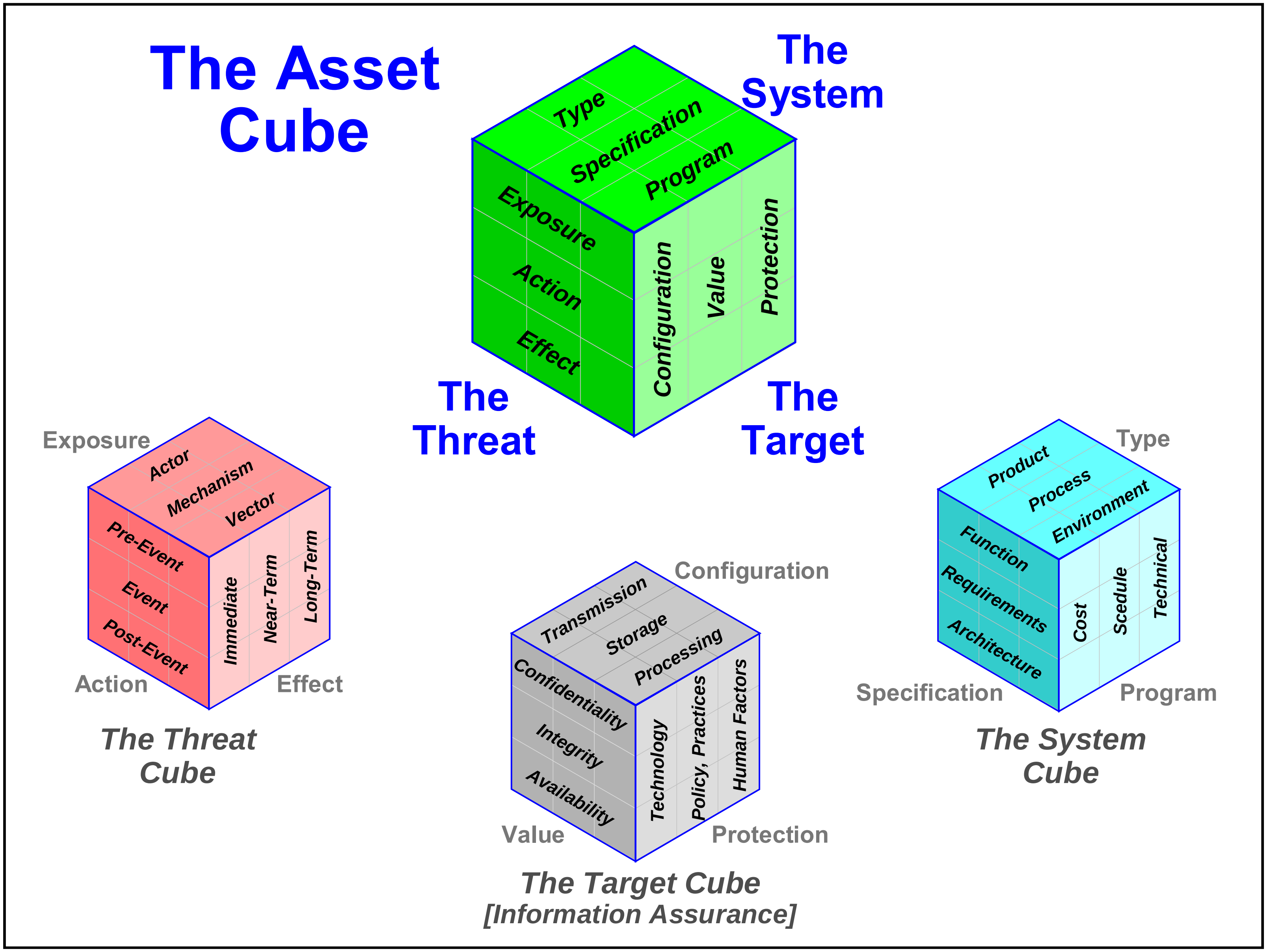
SC**2**.2.1; SC**2**.2.2; SC**2**.2.3

SC**2**.3.1; SC**2**.3.2; SC**2**.3.3

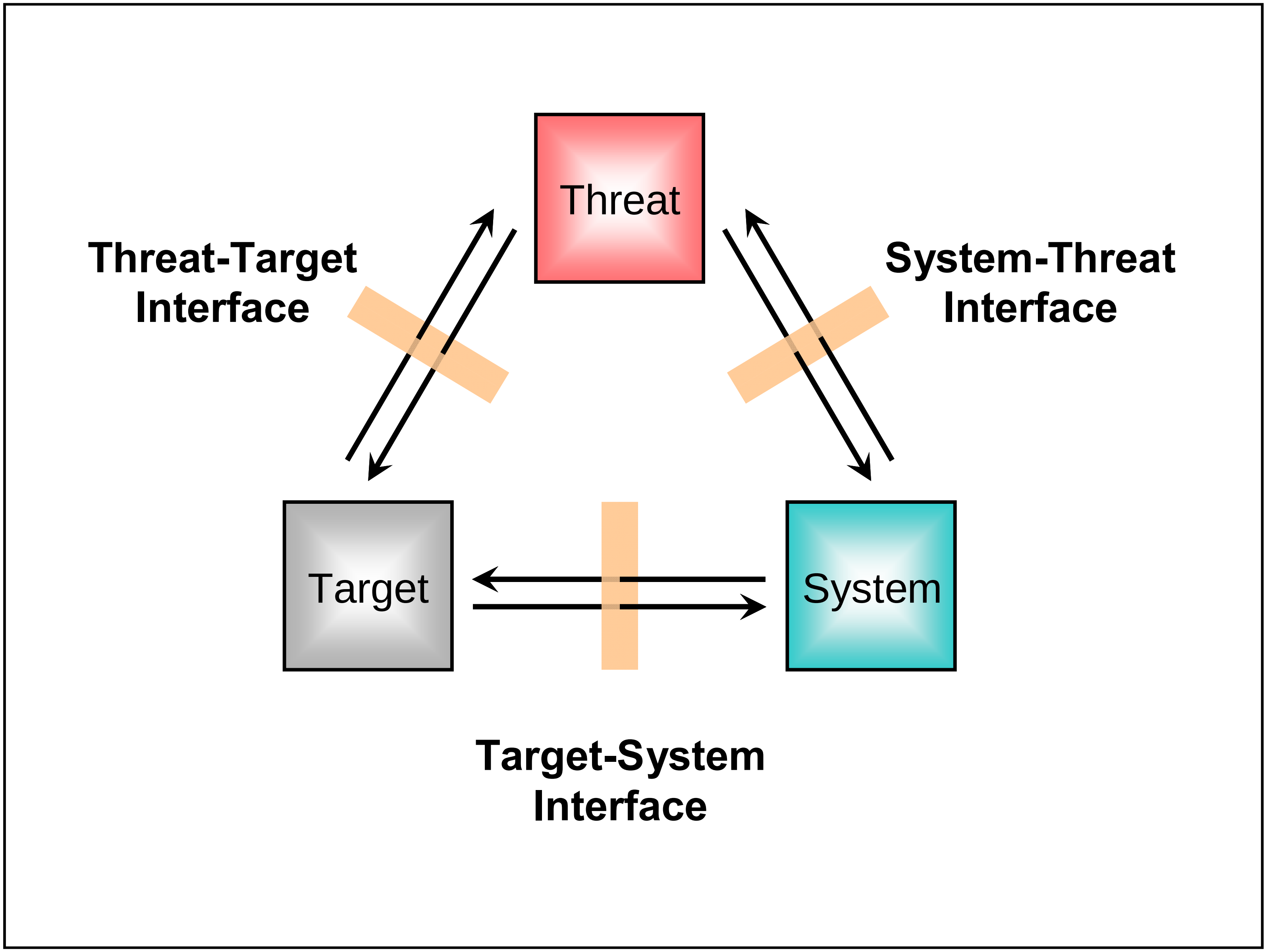
The Concept Cube SM



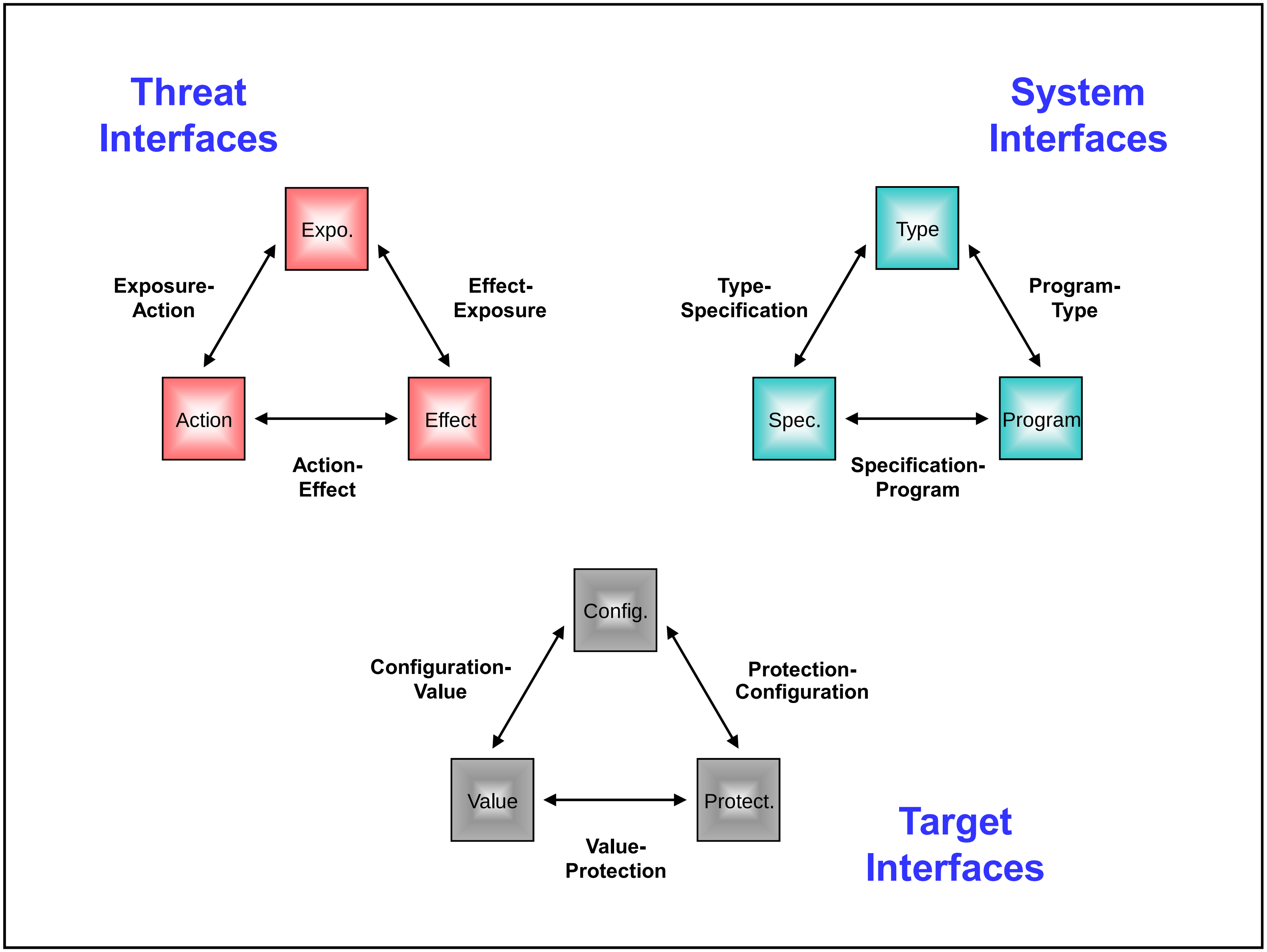
Concept Cube Example, The Asset Cube



Example, The Asset Cube – Level 1 Interfaces



Example, The Asset Cube – Level 2 Interfaces



Summary and Questions

Summary

ART approach facilitates reduction of cognitive complexity

- Encodes typical systems engineering techniques in clearly defined processes and patterns
- Supports the application of computing resources to evaluate alternative system representations
- Separates system structure from system value and system semantics, & supports detailed communication of both the system organizing structural relationship, and system interface value relationships
- Connects these relationships to mathematical relations **that are the basis** for the computation and analytical portions of the cognitive complexity reduction
- Contributes to greater relative density of information transfer due to its use of a formal approach to prose, mathematics, and structural graphics

Early indications show ART evolutionary computation techniques will directly reduce computational complexity associated with the solution of complex system problems

Discussion

- **Questions?**
- **Comments?**