Development and Application of Abstract Relation Types (ART) for Use in System and System-of-Systems Design and Evaluation

Joseph J. Simpson

Dr. Cihan Dagli

Dr. Ann Miller

UMR UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.

Presentation Overview

1- Introduction

2- Complex System Architecting

3- Abstract Relation Type (ART) Definition

4- Binary Relation

5- Context and Concept

6- Functional Hierarchal Abstract Relation Type (HART)

7- Physical Hierarchal Abstract Relation Type (PART)

8- System Evaluation Using ART

9- Summary and Conclusions

Introduction

Motivation for ART development:

- ***Develop standard mechanisms to describe systems***
- ***Develop standard system computing operations***
- ***Provide basis for "system engineering language."***

Wide range of similar implementations:

- ***Warfield's mathematics of structure***
- ***Friedman's Constraint Matrix***
- ***Design Structure Matrices (DSM)***
- ***Dependency Structure Matrices (DSM)***
- ***N Squared Diagrams (Many subtypes).***

Complex Systems Architecting

- **Attributes for Complex Systems**
  - Interdependent
  - Independent
  - Distributed
  - Cooperative
  - Competitive
  - Adaptive

Complex Systems Architecting

- ***Seamless integration and dynamic adaptation to changing environments are common characteristics.***
- ***These characteristics are true for both defense and commercial systems***
- ***Resulting systems are complex; their behavior can be understood through Computational Intelligence, Artificial Life approaches and Complexity Theory***

Complex Systems Architecting

- ***These attributes are being used to create new system definitions***
  - ***Systems of Systems (Interdependent)***
  - ***Family of Systems (Independent)***
  - ***Galaxies of Systems (Distributed)***
  - ***Intelligent Enterprise Systems (Cooperative, Competitive and Adaptive)***
- ***ART are used as a common conceptual building block for complex systems.***
- ***ART provide common context and computational constructs upon which executable computer programs are constructed***

Abstract Relation Types (ART)

ART are based on two primary ideas and constructs:

1- Warfield's methods for structuring complex systems

2- Abstract data types

Warfield's methods include:

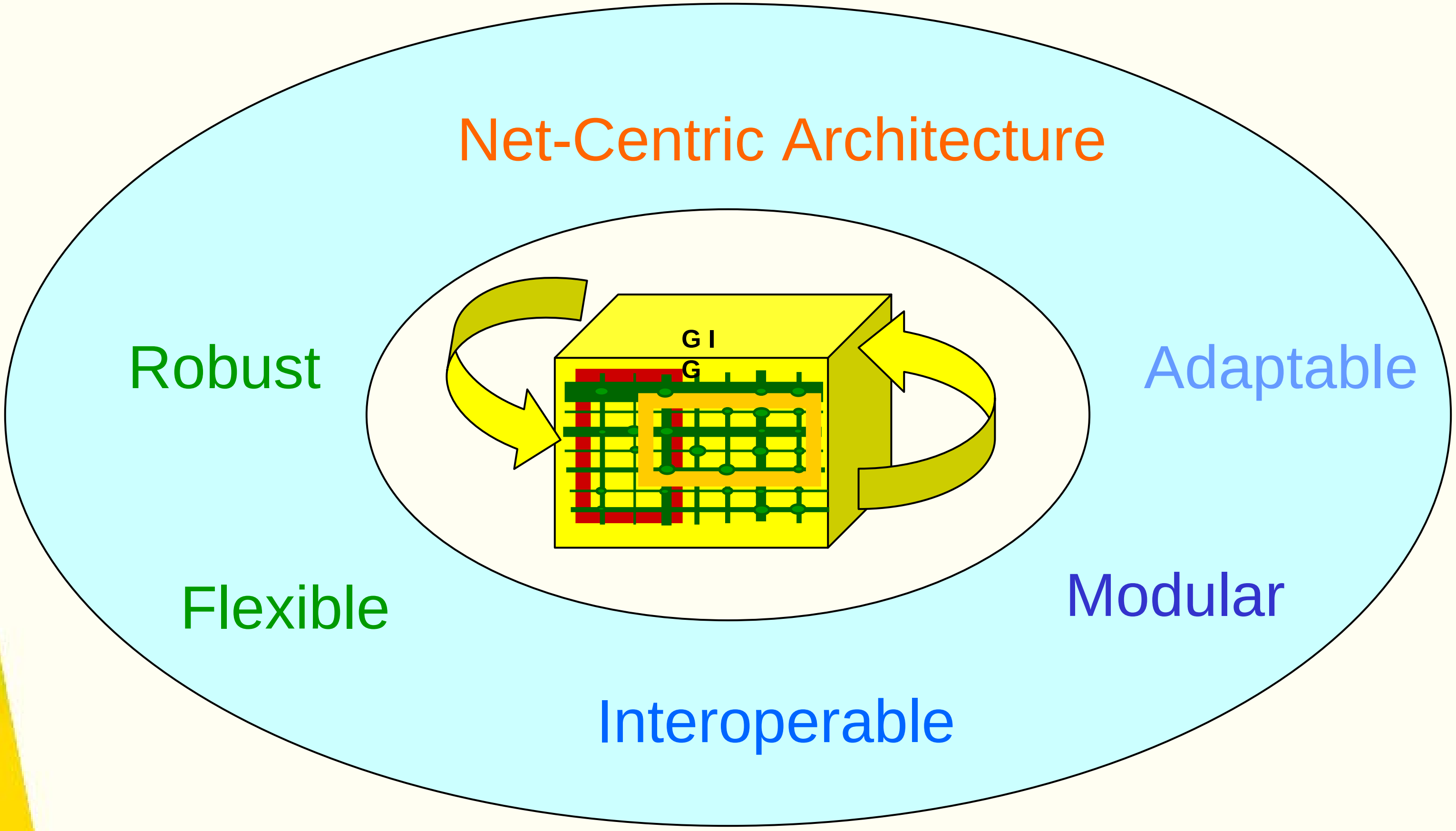
- Natural language contextual binary relations

- Binary matrices and matrix operations

Abstract data types (ADT) are used to organize the structure and specify the operations associated with a specific ADT. Each ADT is independent of the specific real data type that is used to implement the ADT.

ART are not independent of the specific real relation and are used to specify the ART structure as well as specify and organize the ART operations.

Net-Centric System



Net-Centric Architecture

Robust

Adaptable

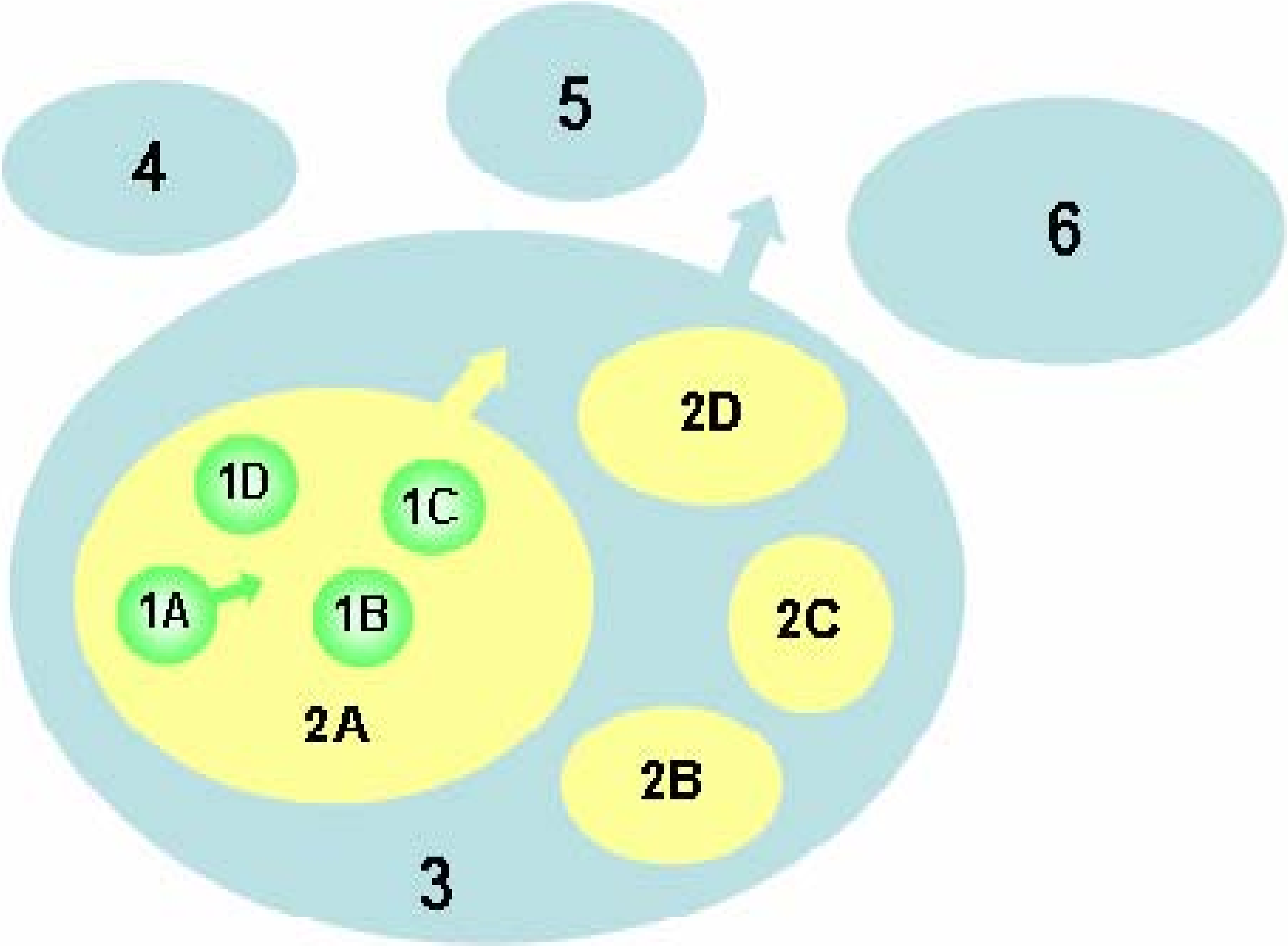
Flexible

Modular

Interoperable

An Evolving Net-Centric Architecture

For the set of systems 1A, 1B, 1C, 1D, 2A, 2B, 2C, 2D, 3, 4, 5, and 6



Example context for System 1A (C1A) with attributes *a, b, c, d, e, f, g,* and *h* can be considered

		Physical Attributes				***Logical Attributes***			
	C1A	***a***	***b***	***c***	***d***	***e***	***f***	***g***	***h***
Systems	**1B**	x		x	x	x	x		
	1C	x	x	x	x	x		x	x
	1D		x	x		x		x	

Where

a = radio frequency connection
b = internet physical port
c = electrical power type
d = heat generation

e = IP routing
f = ftp information service
g = rlogin operations service
h = http network service

Binary Matrix: Set A X Set B

Set B

Set A

A	Youthful	Young	Middle-Aged	Mature	Old
HS	Excellent	Good	Average	Below Average	Poor
HH	Excellent	Good	Average	Below Average	Poor
JR	Minor	Below Average	Average	Above Average	Major
FR	Minor	Below Average	Average	Above Average	Major

A	0	0	0.25	0.50	0.25
HS	0	0.50	0.50	0	0
HH	0.25	0.50	0.25	0	0
JR	0.50	0.50	0	0	0
FR	0.25	0.50	0.25	0	0
RATE	Low	[✓ Mod-Low]	Moderate	Mod-High	High

Set A (Relations) x Set B (Variables)

Variables

Relations

	X	x'	Z	z'	r	r'	S	s'	V	d
1	●		●		●					
2	●	●	●	●		●				
3	●				●		●			
4			●		●		●			
5	●		●				●			
6	●	●	●	●				●		
7		●		●					●	
8		●		●						●

Where

1: $r = (X^2 + Z^2)^{1/2}$

2: $r' = (X'X + Z'Z)(X^2 + Z^2)^{1/2}$

3: $X = r \cos S$

4: $Z = r \sin S$

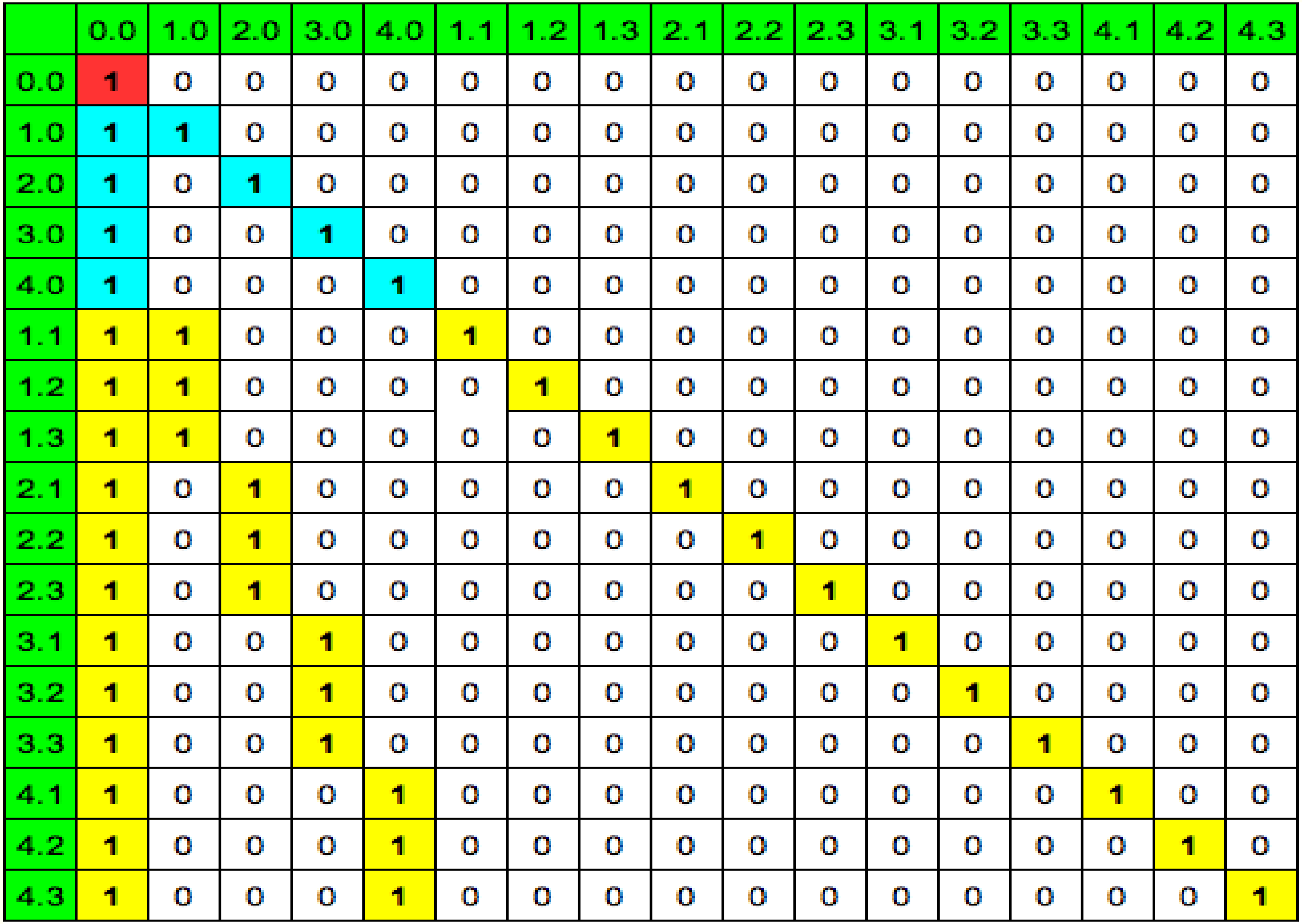
5: $S = \tan^{-1} (Z/X)$

6: $s' = (Z'X - X'Z)(X^2 + Z^2)^{-1}$

7: $r = (X^2 + Z^2)^{1/2}$

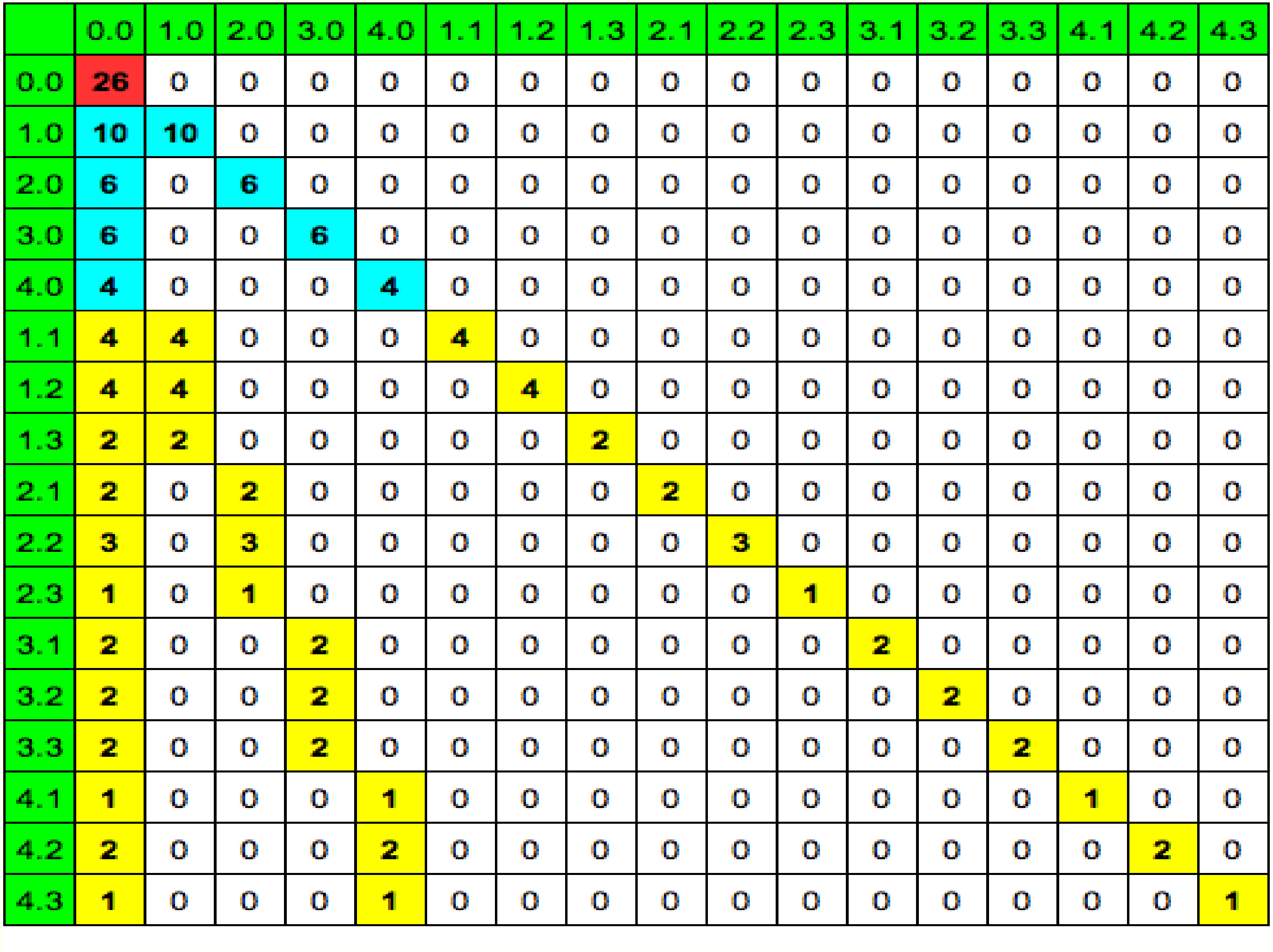
8: $Z = -X' \tan d$

HART Binary Matrix



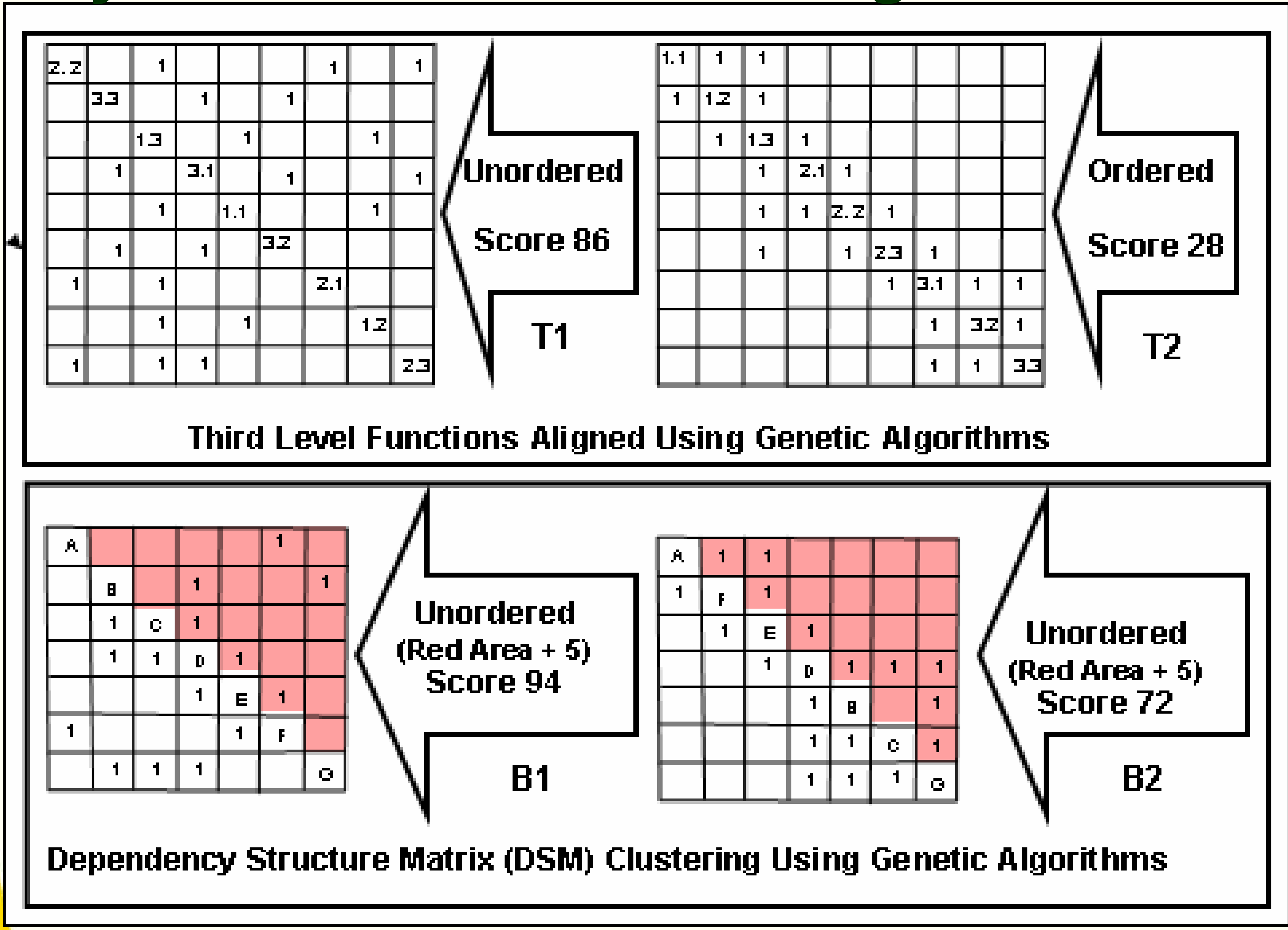
	0.0	1.0	2.0	3.0	4.0	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2	3.3	4.1	4.2	4.3
0.0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4.0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1.1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1.2	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1.3	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
2.1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
2.2	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
2.3	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
3.1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
3.2	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
3.3	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
4.1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
4.2	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
4.3	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

PART Binary Matrix



	0.0	1.0	2.0	3.0	4.0	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2	3.3	4.1	4.2	4.3
0.0	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.0	10	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.0	6	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.0	6	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0
4.0	4	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0
1.1	4	4	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
1.2	4	4	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0
1.3	2	2	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
2.1	2	0	2	0	0	0	0	0	2	0	0	0	0	0	0	0	0
2.2	3	0	3	0	0	0	0	0	0	3	0	0	0	0	0	0	0
2.3	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
3.1	2	0	0	2	0	0	0	0	0	0	0	2	0	0	0	0	0
3.2	2	0	0	2	0	0	0	0	0	0	0	0	2	0	0	0	0
3.3	2	0	0	2	0	0	0	0	0	0	0	0	0	2	0	0	0
4.1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
4.2	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2	0
4.3	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1

System Evaluation Using ART



2.2		1				1		1
	3.3		1		1			
		1.3		1			1	
	1		3.1		1			1
		1		1.1			1	
	1		1		3.2			
1		1				2.1		
		1		1			1.2	
1		1	1					2.3

Unordered

Score 86

T1

1.1	1	1						
1	1.2	1						
	1	1.3	1					
		1	2.1	1				
		1	1	2.2	1			
		1		1	2.3	1		
					1	3.1	1	1
						1	3.2	1
						1	1	3.3

Ordered

Score 28

T2

Third Level Functions Aligned Using Genetic Algorithms

A					1	
	B		1			1
	1	C	1			
	1	1	D	1		
			1	E	1	
1				1	F	
	1	1	1			G

Unordered
(Red Area + 5)
Score 94

B1

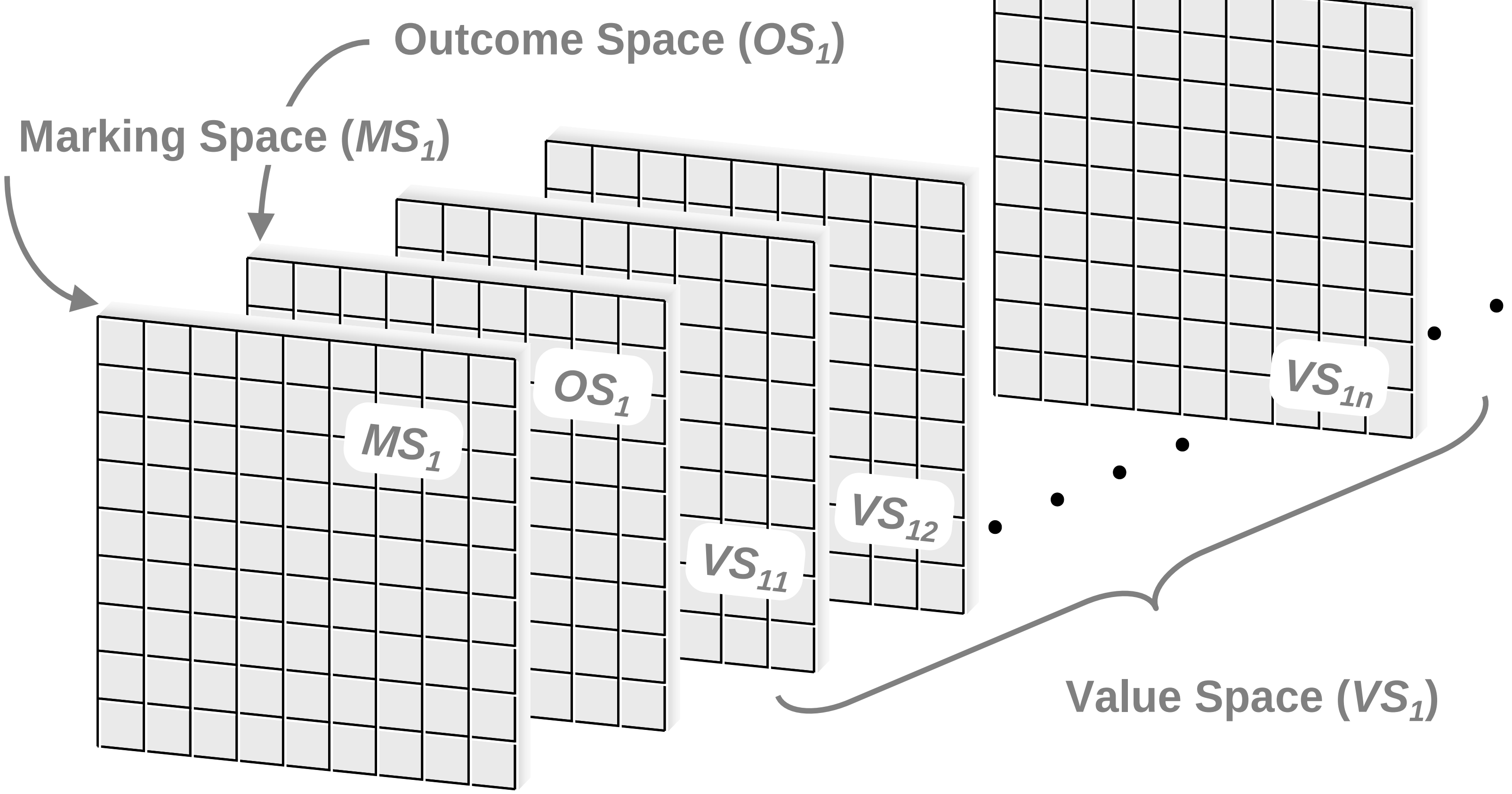
A	1	1				
1	F	1				
	1	E	1			
		1	D	1	1	1
			1	B		1
			1	1	C	1
			1	1	1	G

Unordered
(Red Area + 5)
Score 72

B2

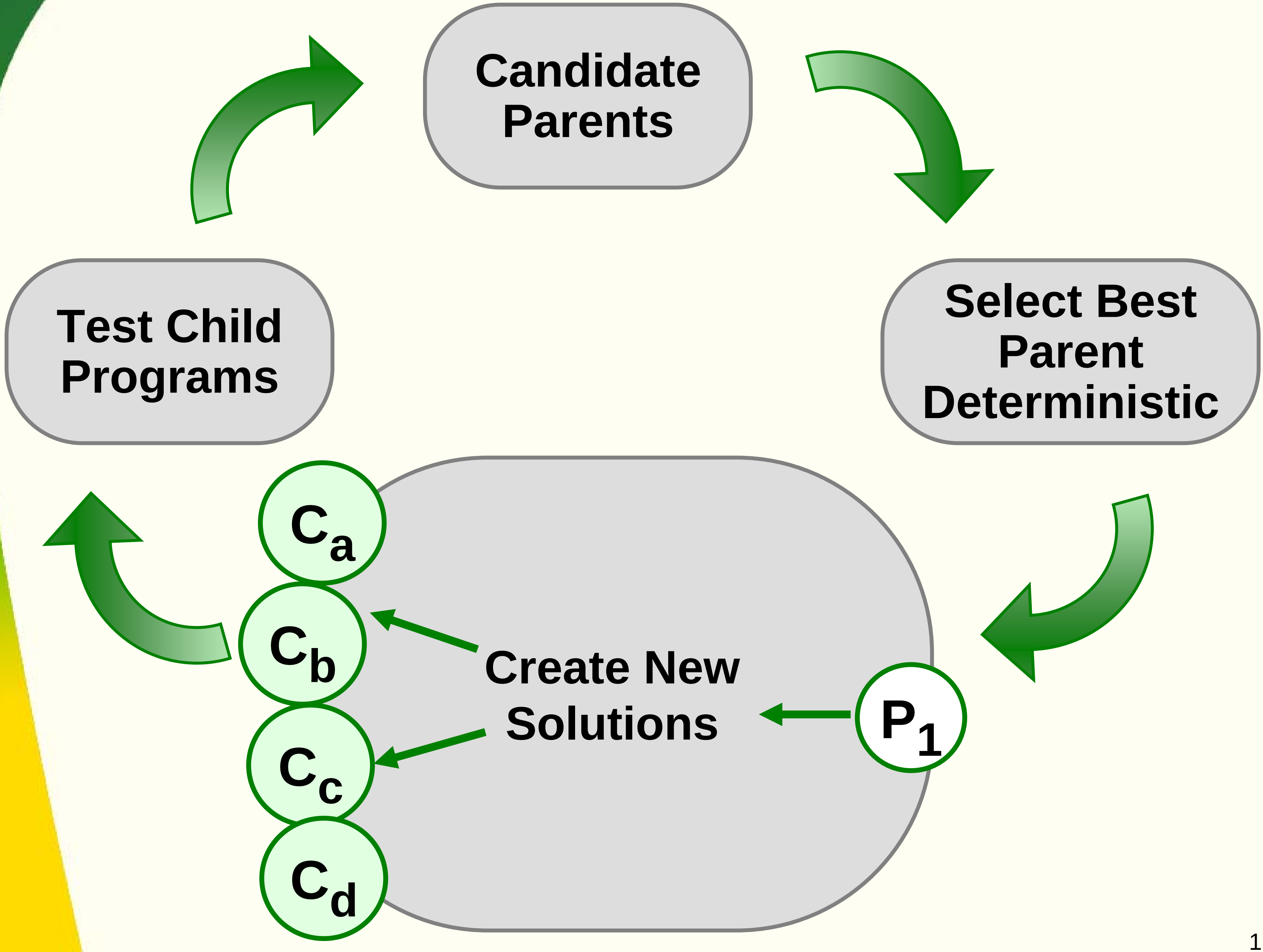
Dependency Structure Matrix (DSM) Clustering Using Genetic Algorithms

Evolutionary Strategy



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

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



Candidate Parents

Select Best Parent Deterministic

Create New Solutions

Test Child Programs

Evolutionary Strategy

- Continue Research to Determine:
  - Best mix of parent and children in a population
  - Specific approaches to calculation
  - Application to relation types
  - Effects of matrix scale
  - Prepare “known solutions” for test

Summary and Conclusions

- Systems engineering needs a standard system computational approach
- ART have been designed and developed as a fundamental component of system computation
- A library of well defined ART provides the basis for clear formulation and communication of systems and system attributes
- ART provide a set of executable system patterns.

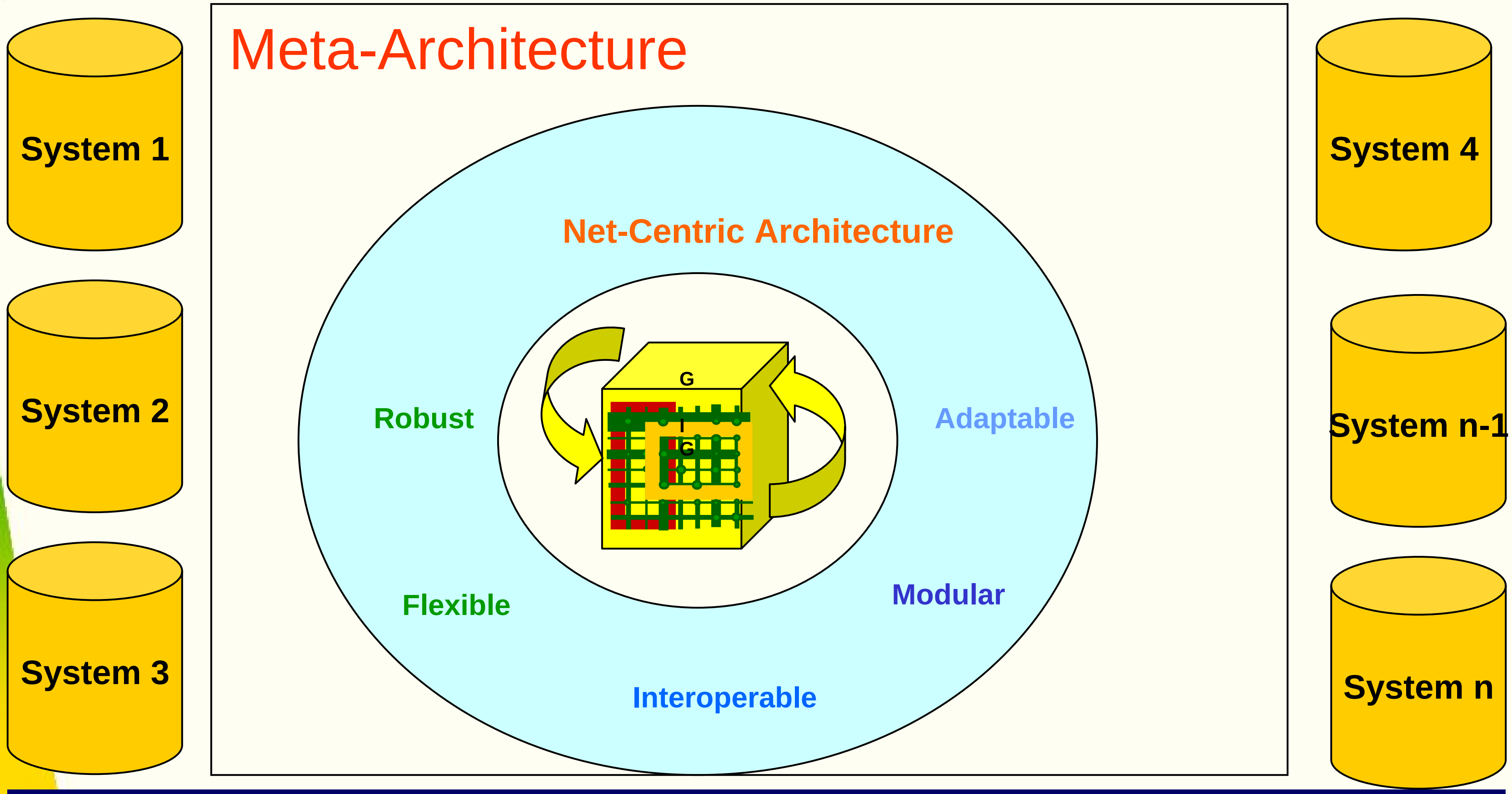
- Further work is needed to fully develop the ART of systems engineering

Questions

Functional Hierarchy ART (HART)

Number	Function Name
0.0	Execute Task
1.0	Prepare for Task Execution
1.1	Conduct Initial Task Briefing
1.2	Prepare Personnel for Task
1.3	Prepare Equipment for Task
2.0	Transit to Task Execution Area
2.1	Prepare for Transit
2.2	Load Unit on Transport
2.3	Move to Task Area
3.0	Perform Task
3.1	Evaluate Task Area
3.2	Execute Task Objectives
3.3	Evaluate Task Effectiveness
4.0	Return From Task Area
4.1	Transit to Initial Area
4.2	Evaluate Unit Status
4.3	Record Lessons Learned

Complex Systems Architecting



System 1

System 2

System 3

Meta-Architecture

Net-Centric Architecture

Robust

Adaptable

Flexible

Modular

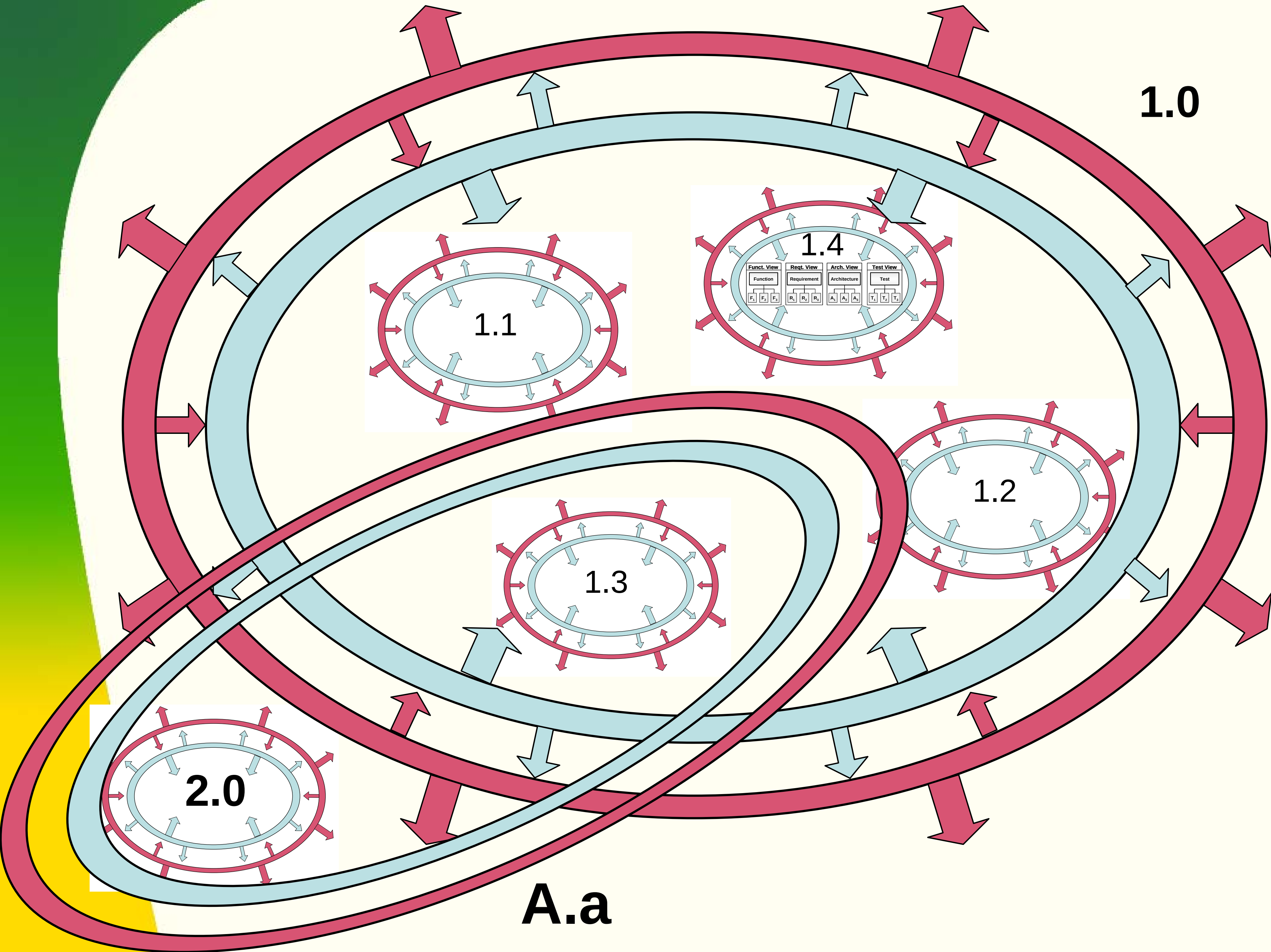
Interoperable

System 4

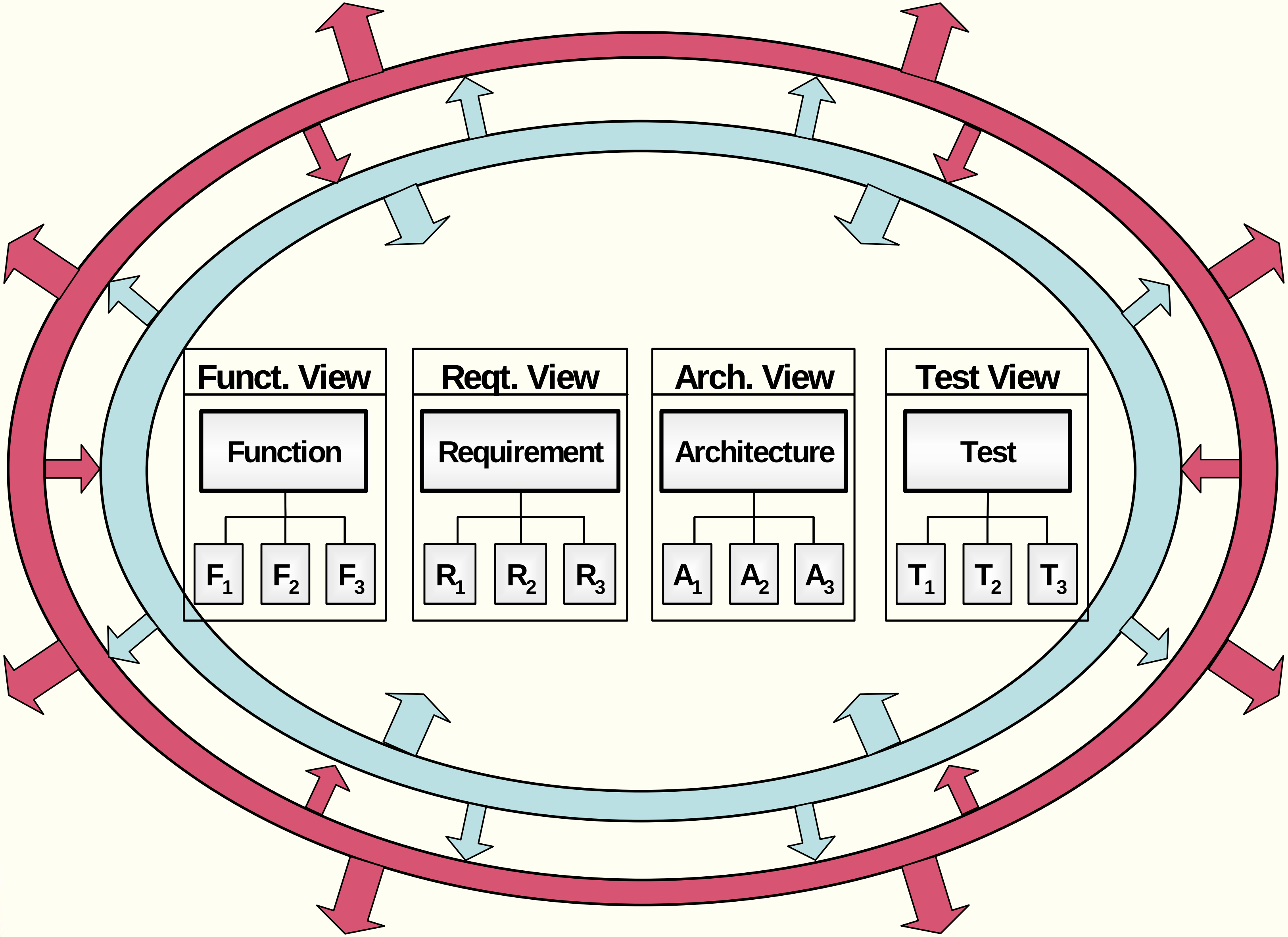
System n-1

System n

Dynamically Changing Meta-Architecture for Complex Systems


1.0
1.4
Funct. View
Function
F_1
F_2
F_3
Reqt. View
Requirement
R_1
R_2
R_3
Arch. View
Architecture
A_1
A_2
A_3
Test View
Test
T_1
T_2
T_3
1.1
1.2
1.3
2.0
A.a

CCFRAT System Model



Funct. View

Function

F_1 F_2 F_3

Reqt. View

Requirement

R_1 R_2 R_3

Arch. View

Architecture

A_1 A_2 A_3

Test View

Test

T_1 T_2 T_3