Objects, Relations and Clusters for System Analysis

Joseph J Simpson
Mary J Simpson

Overview

- Definitions
- Types of cluster analysis
- Role of system organizing relationship
- Abstract Relation Type (ART)
- Augmented Model-Exchange Isomorphism (AMEI)
- Connection to classical system engineering methods and techniques

Systems and Clusters

- **A ‘construction-rule’ system definition**

  A relationship mapped over a set of objects

- **A ‘function-rule’ system definition**

  A constraint on variation

- **Cluster**

  A group of objects occurring closely together

- **Object-based cluster identification**

  Based on object attributes

- **Space-based cluster identification**

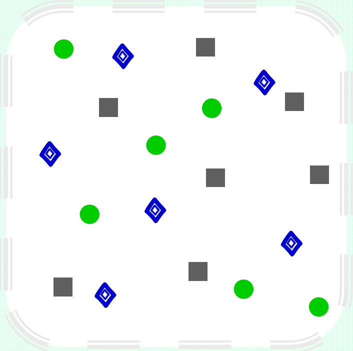
  Based on relation properties

Cluster Types

Object-Based Cluster

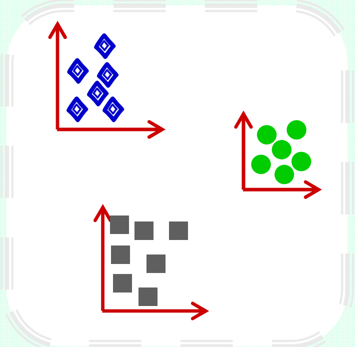
1

Identify, analyze objects



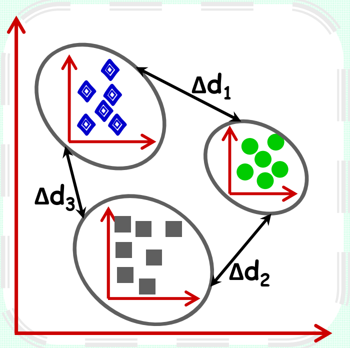
2

Determine cluster dimensions



3

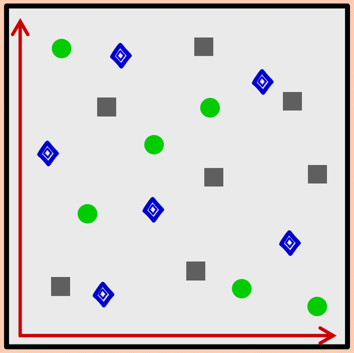
Analyze object clusters



Space-Based Cluster

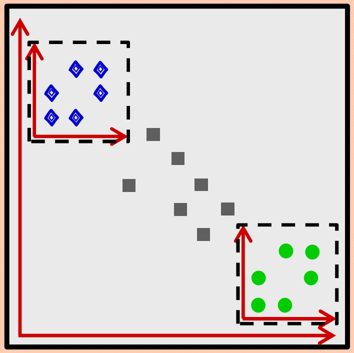
1

Identify analysis space using global system relation



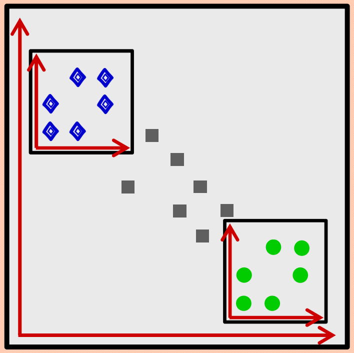
2

Identify subspace(s) of interest



3

Enumerate objects in subspace



Variable and Object Analysis

- **Variable analysis based on object properties**

  Degree of similarity among variables used to identify and describe the controlling object properties of interest

- **Object analysis based on class construction**

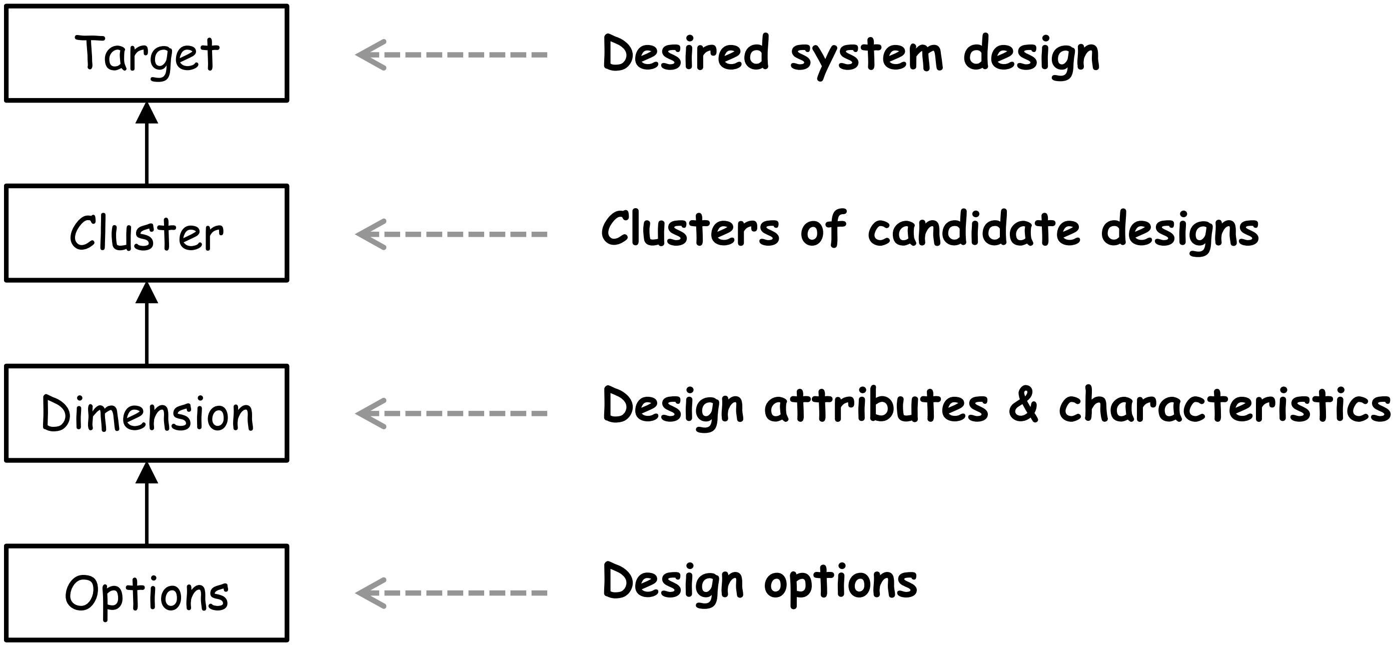
  The activity of identifying the general types into which the objects may be categorized or classed

Object analysis requires a large amount of, and greater depth of, contextual information.

As a result, it requires more specific application subject matter expertise than variable analysis.

Context for Cluster Application

Warfield's 'Four Level Inclusion Hierarchy for Design'



Target ← Desired system design

Cluster ← Clusters of candidate designs

Dimension ← Design attributes & characteristics

Options ← Design options

This represents a generalized 'included-in' relation, that becomes more specialized as the Target is achieved.

The following logical relation properties apply to 'included-in'

- ***Irreflexive***
- ***Asymmetric***
- ***Transitive***

Adapted from Figure 2.20, Warfield, 1994.

Logical Relation Properties

System Concepts SM

Hi-Level Logical Characteristics of Three Dyadic Relations - v1.1

Reflexivity *Involves one individual*	***Symmetry*** *Involves two individuals*	***Transitivity*** *Involves three (or more) individuals*
Reflexive A relation, R, is reflexive iff any individual that enters into the relation bears R to itself. ***Identical with; Divisible by**	**Symmetric** If any individual bears the relation to a second individual, then the second bears it to the first. ***Touching**	**Transitive** If any individual bears this relation to a second and the second bears it to a third, then the first bears it to the third. ***Greater than; North of; Included in**
Irreflexive A relation, R, is irreflexive iff no individual bears R to itself. ***Stand next to; Father of**	**Asymmetric** A relation, R, is asymmetrical iff, if any individual bears R to a second, then the second does not bear R to the first. ***North of; Heavier than; Child of**	**Intransitive** A relation, R, is intransitive iff, if any individual bears R to a second and the second bears R to a third, then the first does not bear R to the third. ***Father of; 2" taller than**
Nonreflexive A relation which is neither reflexive nor irreflexive is nonreflexive. ***Respecting; Killing**	**Nonsymmetric** A relation which is neither symmetrical nor asymmetrical is nonsymmetric. ***Likes; Seeing**	**Nontransitive** A relation which is neither transitive nor intransitive is nontransitive. ***Admiring; Fearing**

***Examples**

Adapted from *Predicate Logic* and *Handbook of Discrete & Combinatorial Mathematics*

The ART Construct

Abstract Relation Type (ART)

Prose Description (text, words)

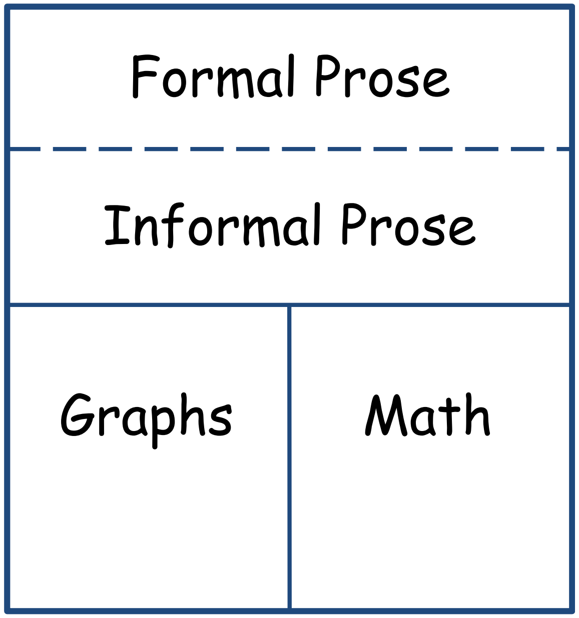
- Formal pattern
- Informal prose

Graphic Representation (directed graphs)

- Must have formal graphs
- Can also have informal graphs

Mathematics & Computer Representation

- Math equations
- Computer codes
- One or both



Formal Prose

Informal Prose

Graphs

Math

Augmented Model-Exchange Isomorphism

System Concepts

Abstract

Relation

Type

Reflected in

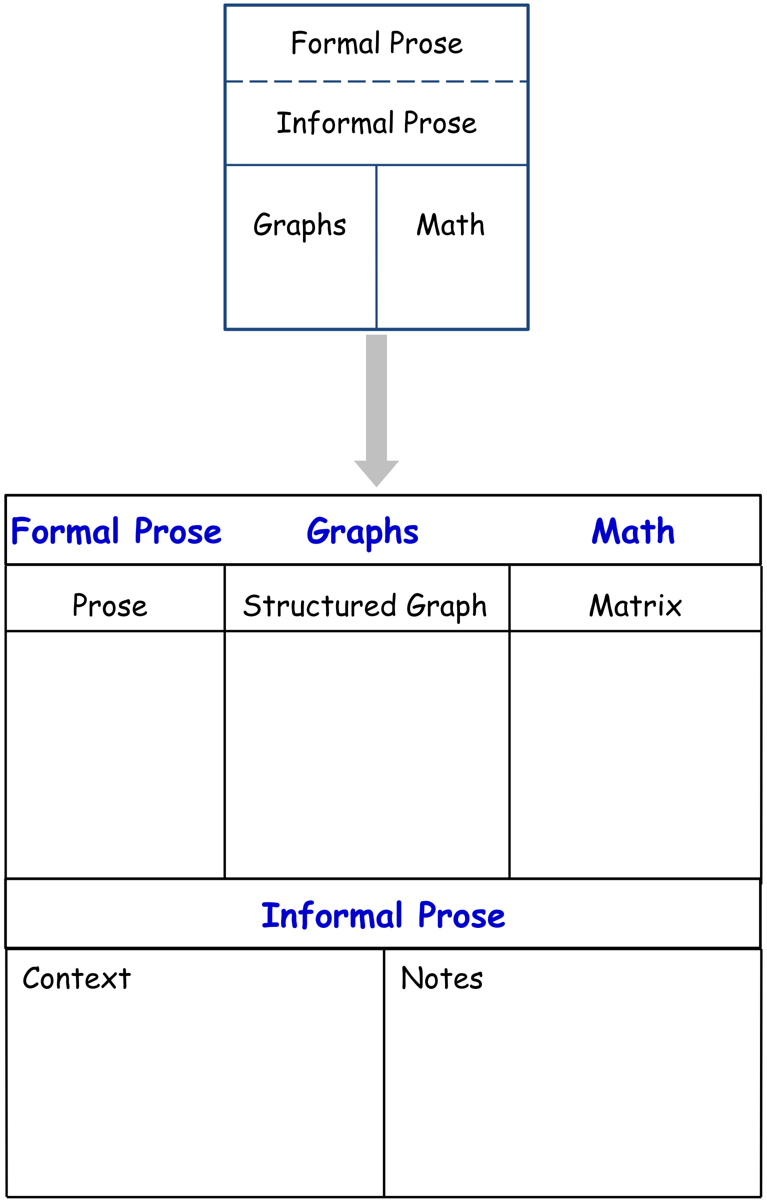
Augmented

Model

Exchange

Isomorphism

(AMEI)



Formal Prose

Informal Prose

Graphs	Math

Formal Prose	**Graphs**	**Math**
Prose	Structured Graph	Matrix
Informal Prose		
Context	Notes	

ART reflected in AMEI

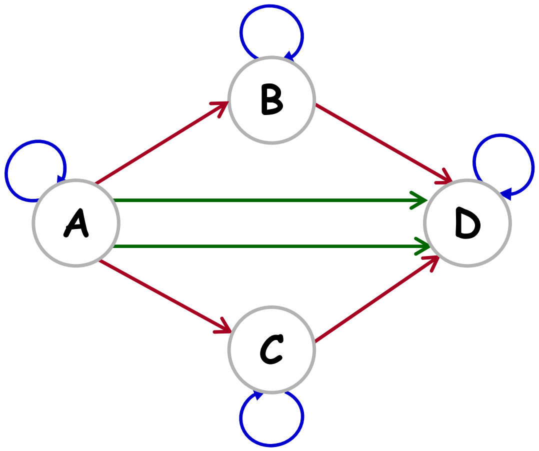
Prose

Relation 'Connected-to'

- **Reflexive**
- ***Asymmetric***
- **Transitive**

RAT-[1,2,1] v1.1

Structured Graph



Matrix

	A	B	C	D
A	1	1	1	1
B	0	1	0	1
C	0	0	1	1
D	0	0	0	1

Context

1. Directional connections
2. Single direction
3. Self-connection required

Notes

1. Shows transitive links

ART reflected in AMEI

System Concepts℠

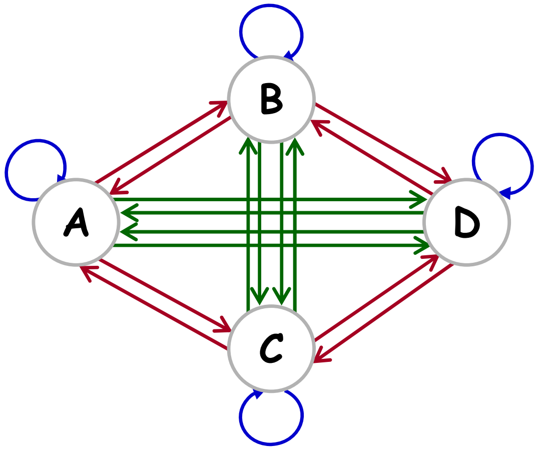
Prose

Relation 'Connected-to'

- **Reflexive**
- **Symmetric**
- **Transitive**

RST-[1,1,1] v1.1

Structured Graph



Matrix

	A	B	C	D
A	1	1	1	1
B	1	1	1	1
C	1	1	1	1
D	1	1	1	1

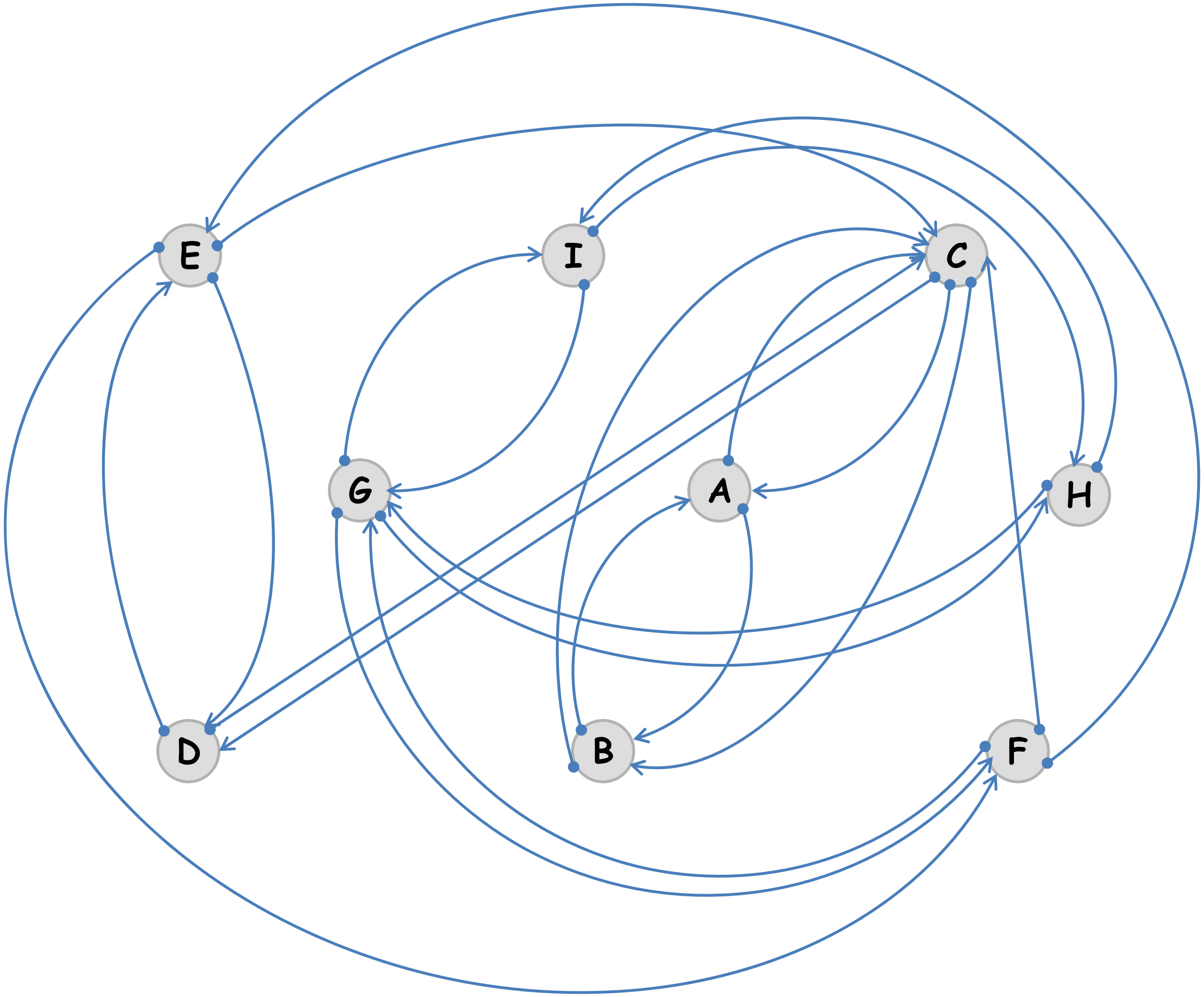
Context

1. Directional connections
2. Double directions
3. Self-connection required

Notes

1. Shows transitive links

Logical Properties?



Adapted from Figure 8.13, Hitchins, 2003.

Identify Clusters

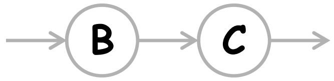
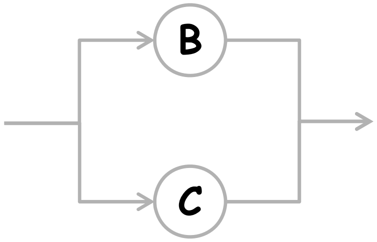
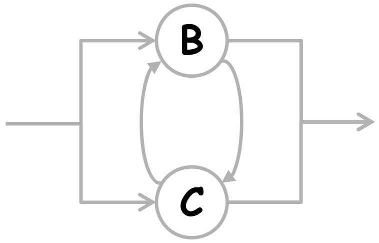
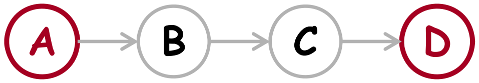
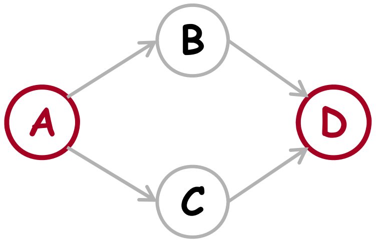
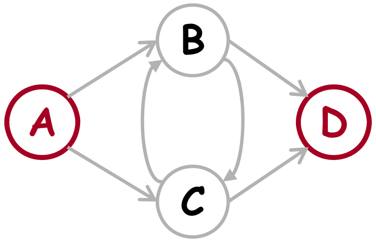
Disordered System Configuration

E	0	1	0	0	0	1	0	1
0	I	0	1	0	1	0	0	0
0	0	C	0	1	0	1	1	0
0	1	0	G	0	1	0	0	1
0	0	1	0	A	0	0	1	0
0	1	0	1	0	H	0	0	0
1	0	1	0	0	0	D	0	0
0	0	1	0	1	0	0	B	0
1	0	1	1	0	0	0	0	F

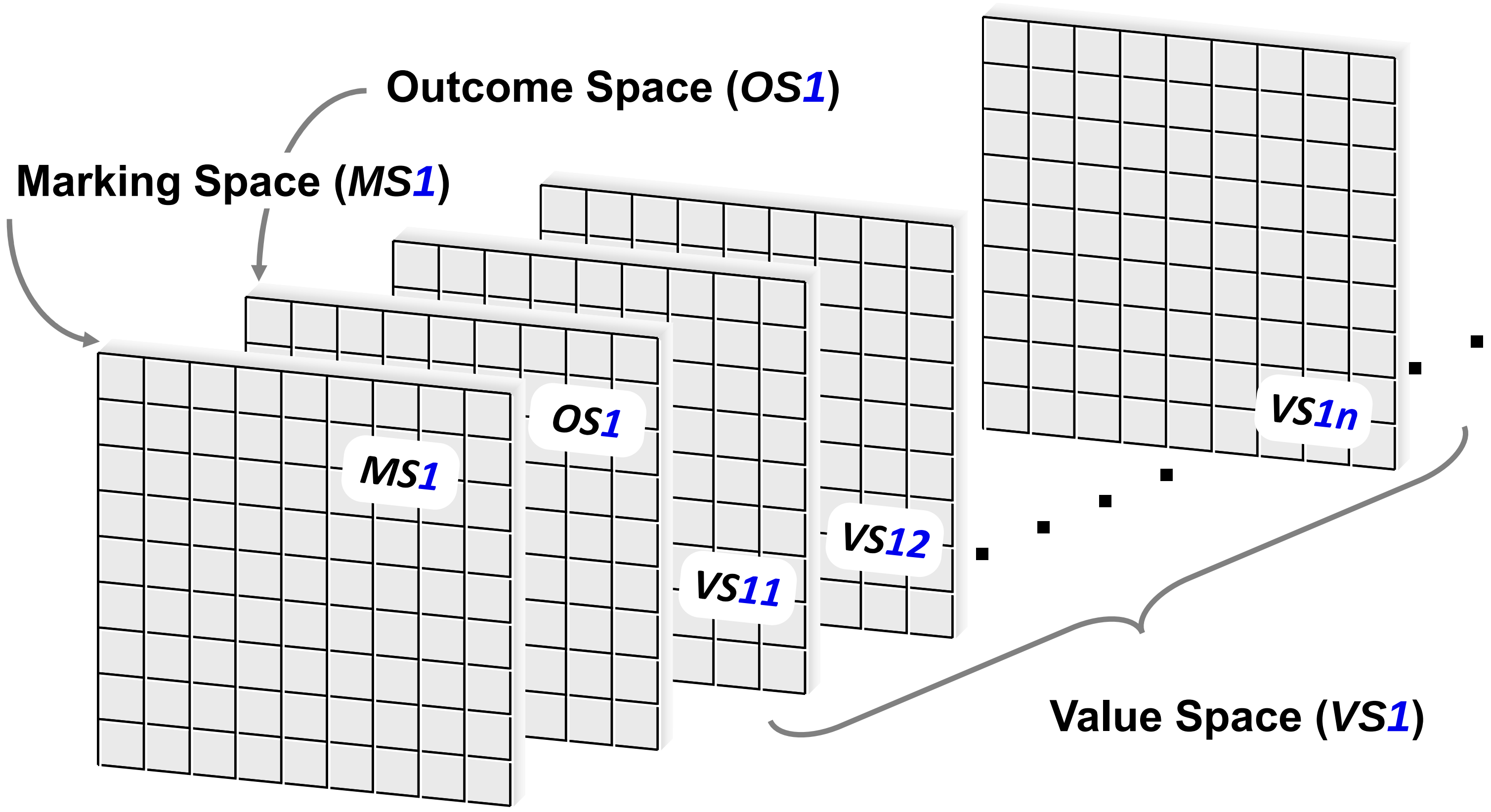
Ordered System Configuration

A	1	1	0	0	0	0	0	0
1	B	1	0	0	0	0	0	0
1	1	C	1	0	0	0	0	0
0	0	1	D	1	0	0	0	0
0	0	1	1	E	1	0	0	0
0	0	1	0	1	F	1	0	0
0	0	0	0	0	1	G	1	1
0	0	0	0	0	0	1	H	1
0	0	0	0	0	0	1	1	I

No Relationship!

Dependent (Series)	Independent (Parallel)	Interdependent (Coupled)
Eppinger's Representation		
		
Add Missing Vertices, Repair Malformed Arcs		
		
Matrix Forms		
$\begin{array}{ccccc} & A & B & C & D \\ A & 0 & 1 & 0 & 0 \\ B & 0 & 0 & 1 & 0 \\ C & 0 & 0 & 0 & 1 \\ D & 0 & 0 & 0 & 0 \end{array}$	$\begin{array}{ccccc} & A & B & C & D \\ A & 0 & 1 & 1 & 0 \\ B & 0 & 0 & 0 & 1 \\ C & 0 & 0 & 0 & 1 \\ D & 0 & 0 & 0 & 0 \end{array}$	$\begin{array}{ccccc} & A & B & C & D \\ A & 0 & 1 & 1 & 0 \\ B & 0 & 0 & 1 & 1 \\ C & 0 & 1 & 0 & 1 \\ D & 0 & 0 & 0 & 0 \end{array}$

ART ‘Spaces’



Outcome Space (*OS1*)

Marking Space (*MS1*)

Value Space (*VS1*)

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

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

Summary

- Relationships create systems
- Abstract Relation Types focus on relationships
- Relationship logical properties create classes of system types
- Classical systems engineering methods and techniques support clustering

Additional Information

Additional information is available

- http://systemsconcept.org/
- https://github.com/jjs0sbw

To join in the discussion and activity

Contact jjs0sbw@gmail.com

This presentation hits the highlights

More detail in the Thursday tutorial

Sign up for the email newsletter

Questions?

Types of Questions

A Good Question

I understand the question, **and** I have an answer.

An Excellent Question

I understand the question; I have an answer -

and charts!

An Interesting Question

I have no idea what you are talking about…

Backup Slides

Types of Set Definition

Set Definition by Extension

All set members are enumerated

Set Definition by Intention

A set is described by listing the defining properties of the members