25th anniversary
annual INCOSE
international symposium
Seattle, WA
July 13 - 16, 2015

Guidelines for creating systems

Dr Joseph E. Kasser
Temasek Defence Systems Institute
National University of Singapore

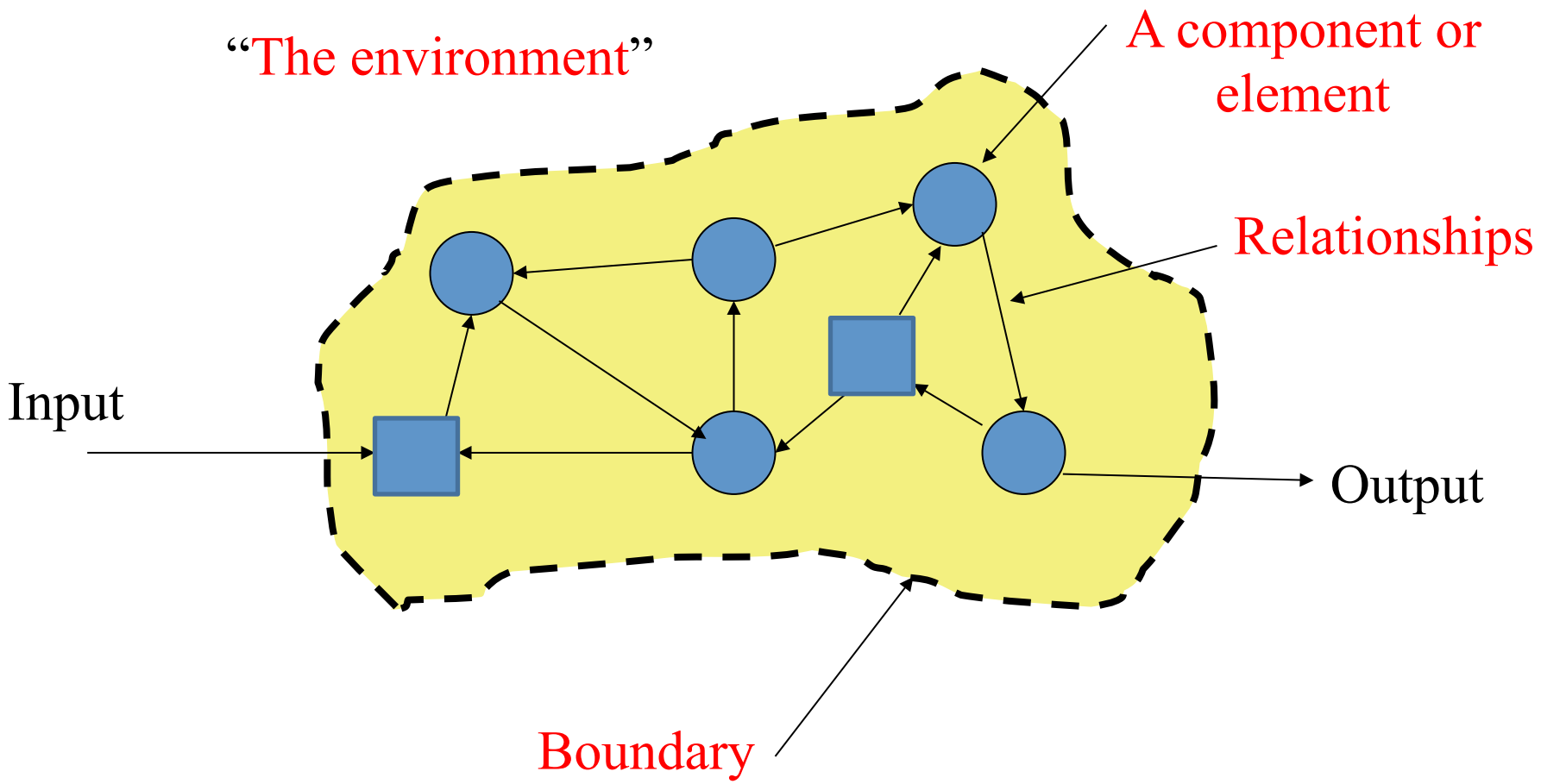
Topics

- Mind the gap
- The problem
- Rules for creating systems
- Summary
- Questions and comments

Mind the Gap

- Systems engineering education tends to assume the system exists and go on from there
- There is a need for guidance on creating the system
- This presentation addresses that need

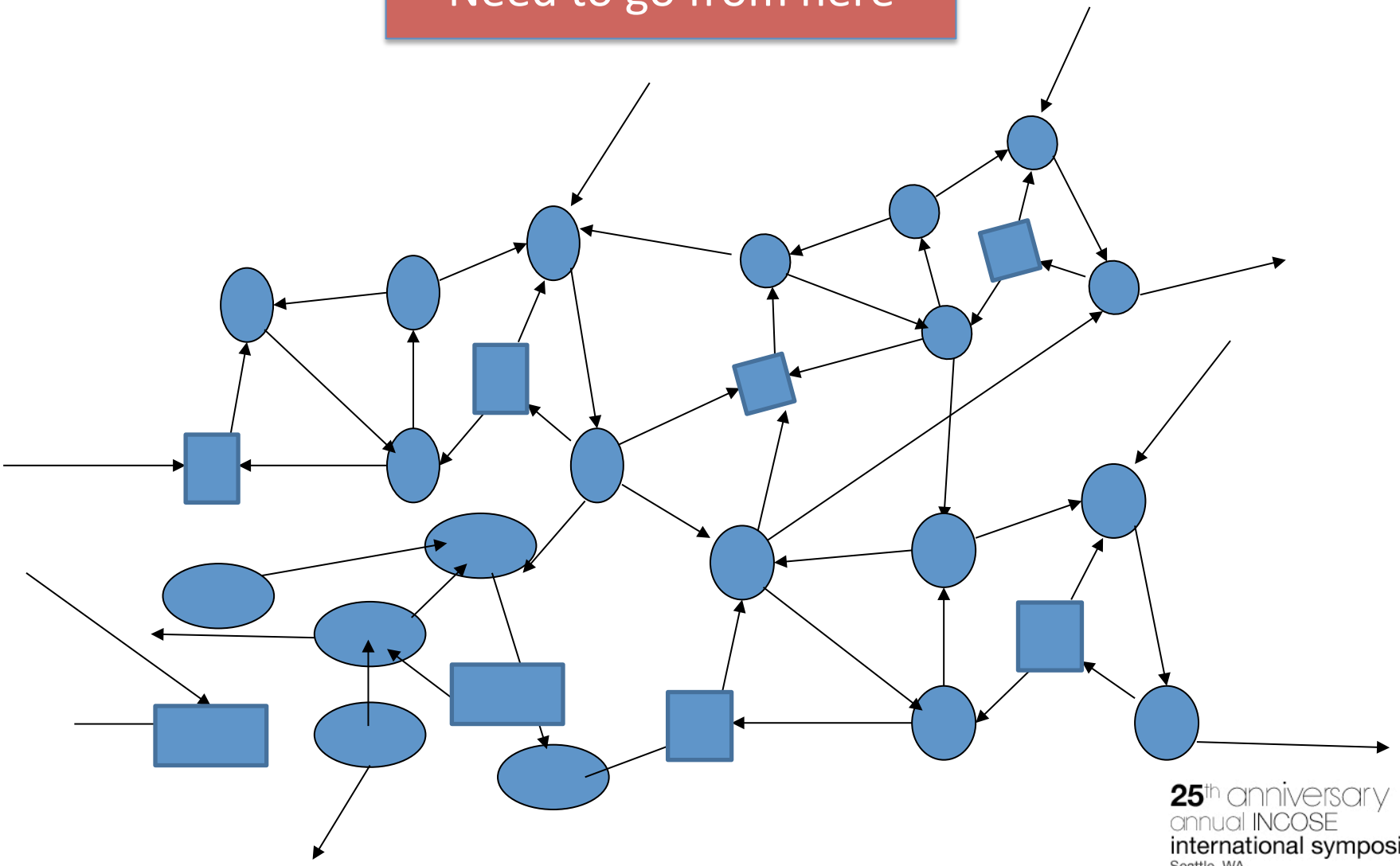
Common Elements of a System*



* Flood and Jackson, 1991

The Situation

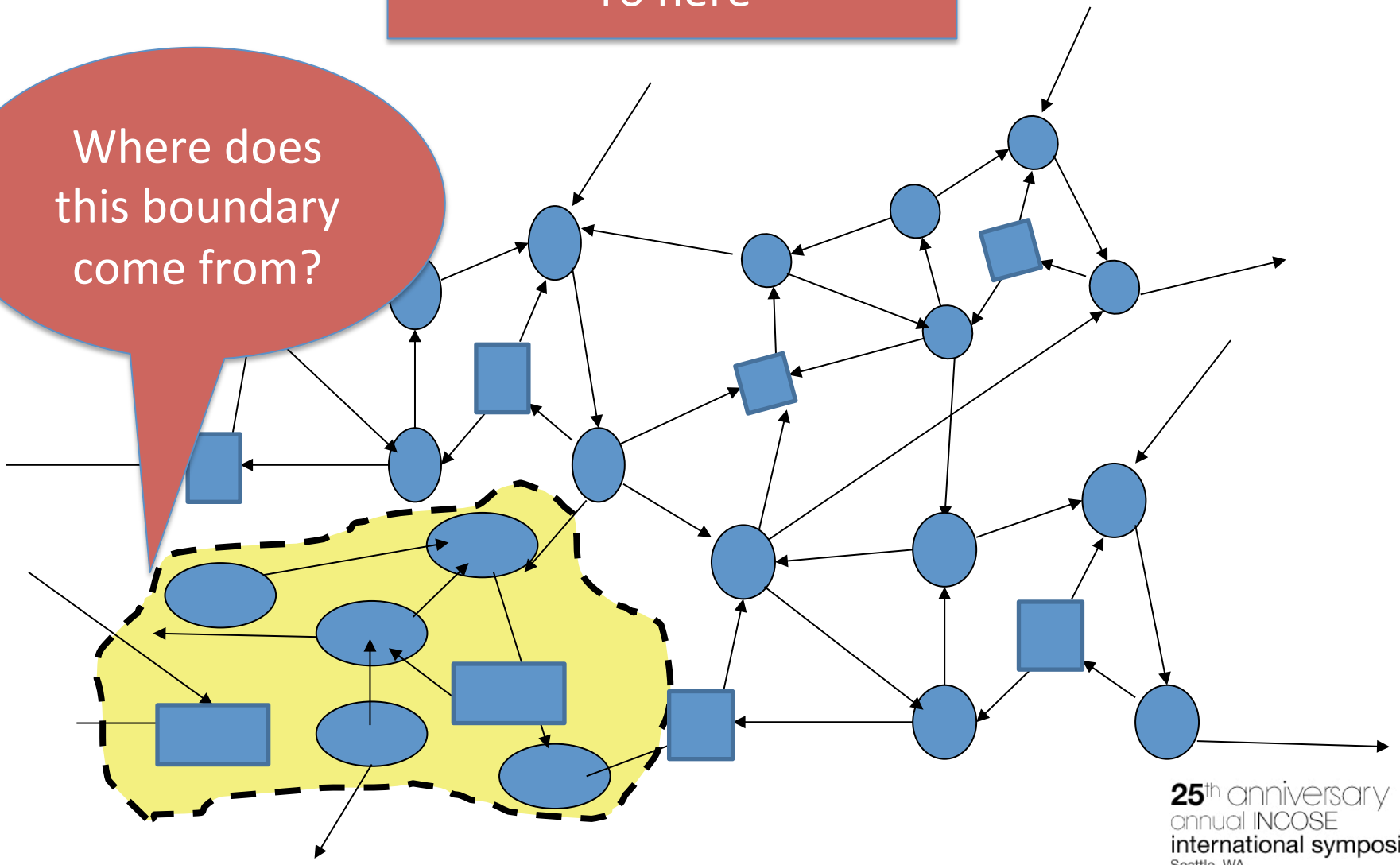
Need to go from here



The System of Interest (SOI)

To here

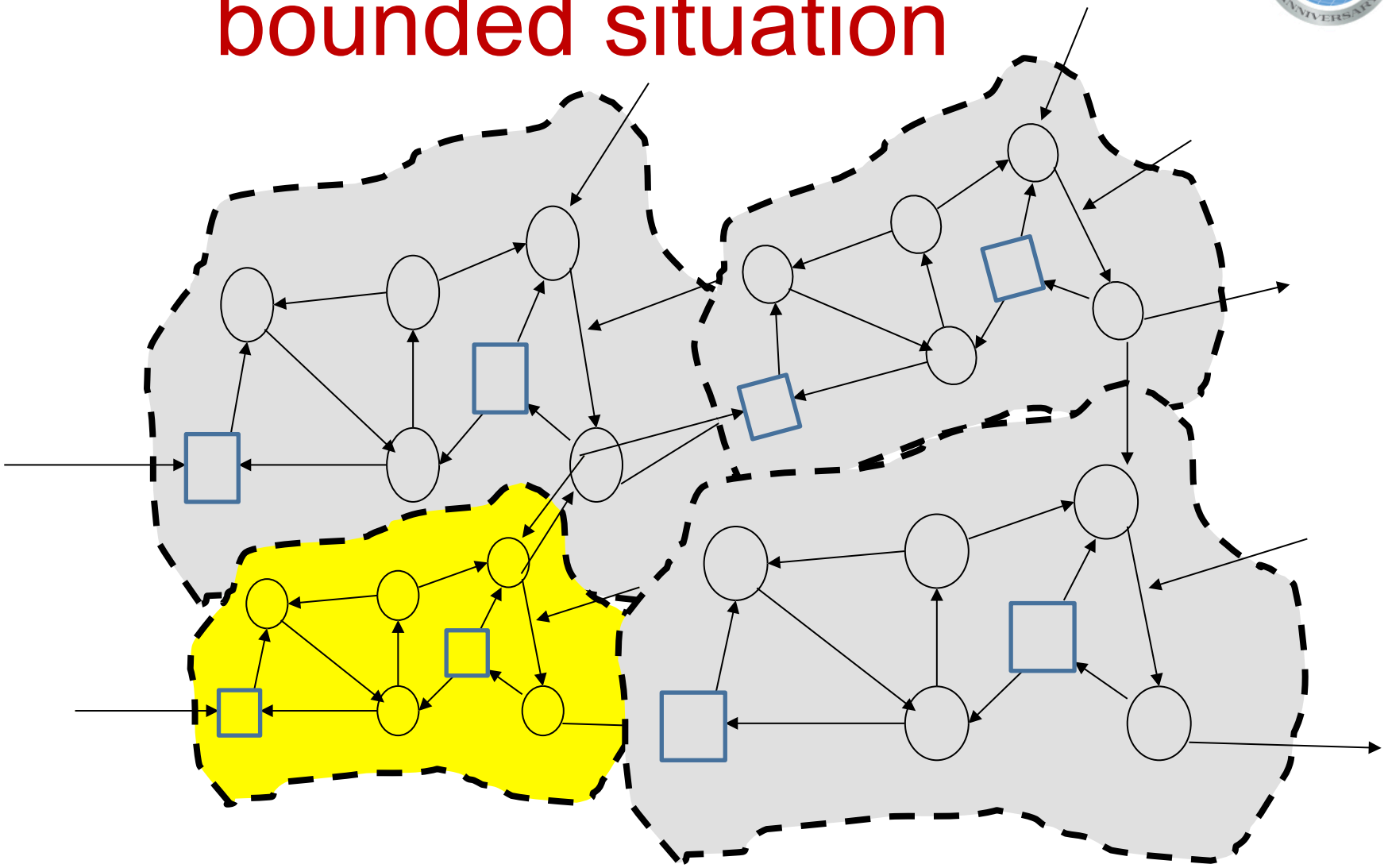
Where does this boundary come from?



Partition the situation

- The act of drawing the system boundary creates the system
  - (Beer, 1994; Churchman, 1979: page 91)
  - ***Inside the boundary***
    - Part of the System of Interest (SOI)
    - Partitioned into subsystems or components
      - people, technology, processes, doctrine, etc.
  - ***Outside the boundary***
    - The context, metasystem or environment
    - Partitioned into adjacent systems

Big Picture Perspective: a bounded situation



Problem formulation template*

1. ***The undesirable situation***
   - Perceived from the Holistic Thinking Perspectives (HTP)**
2. ***The Feasible Conceptual Future Desired Situation (FCFDS)***
   - Perceived from the HTPs
3. ***The problem***
   - How to convert the FCFDS to reality
4. ***The solution*** that remedies the undesirable situation has to be interoperable with evolving adjacent systems over the operational life of solution and adjacent systems
   - The solution is made of two interdependent parts
     a. the SDP or transition process that converts the undesirable situation to a desirable situation, and
     b. the solution system operating in the context of the desirable situation.

* Kasser, 2015, Perceptions of Systems Engineering

* Kasser, 2013, Holistic Thinking: creating innovative solutions to complex problems

Formulation of problem

- The Undesirable Situation
  - The need to define the system boundary/boundaries
- FCFDS
  - The boundaries of the system are defined
- The Problem (Well-structured)
  1. What to do?
     - The Gap
  2. How to do it?
     - Unknown to many people
- The Solution
  - The FCFDS

Process/rules*-1

1. **Examine the undesirable situation from several different perspectives**
2. Develop an understanding of the situation
3. Create the FCFDS containing the SOI
4. Use the principle of hierarchies to abstract out the complexity
5. Abstract out the parts of the situation that are not pertinent to the problem
6. Partition the FCFDS into the SOI and adjacent systems
7. Optimize the interfaces
8. Partition the SOI into subsystems

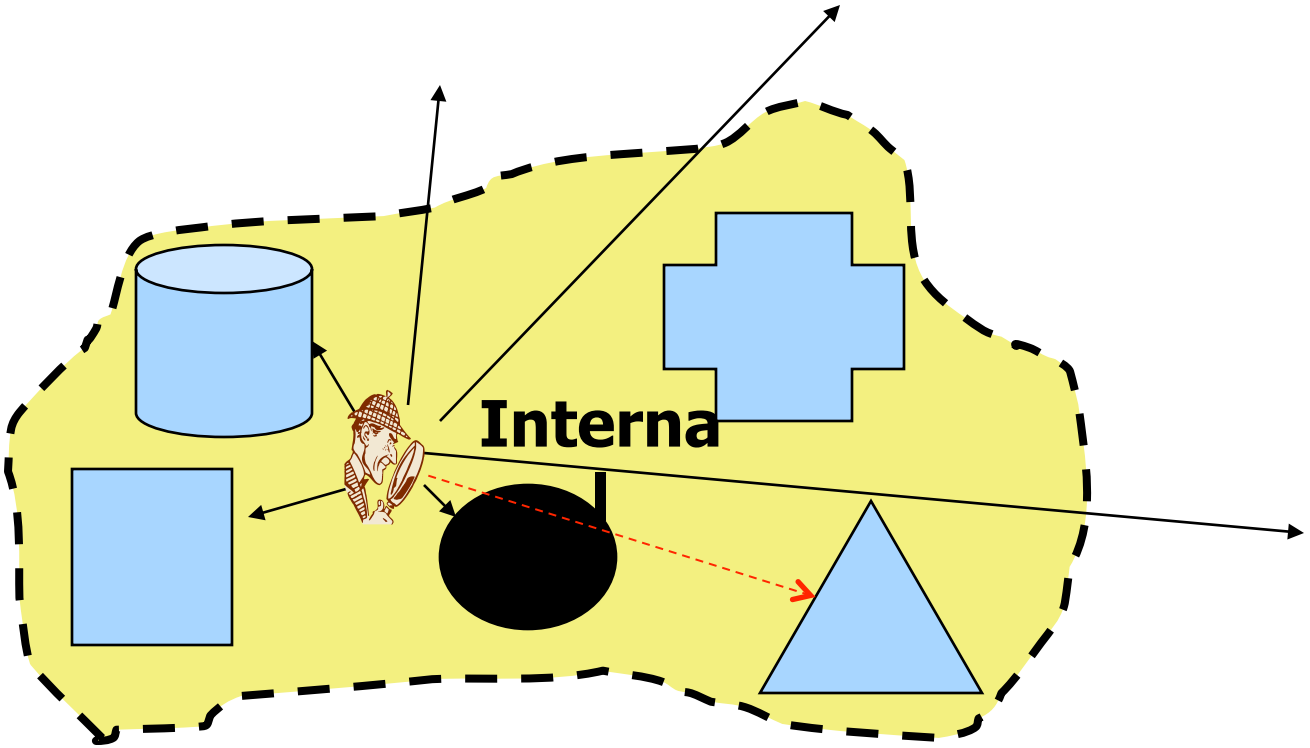
*The activities should be performed in an iterative sequential parallel manner not in a sequential manner

Examine the undesirable situation from several different perspectives

- Perceive situation from multiple perspectives to avoid incorrect conclusions
  - HTPs*
- Identify
  - Entities in the situation
  - Relationships between entities
    - Causal loops
    - Models and simulations

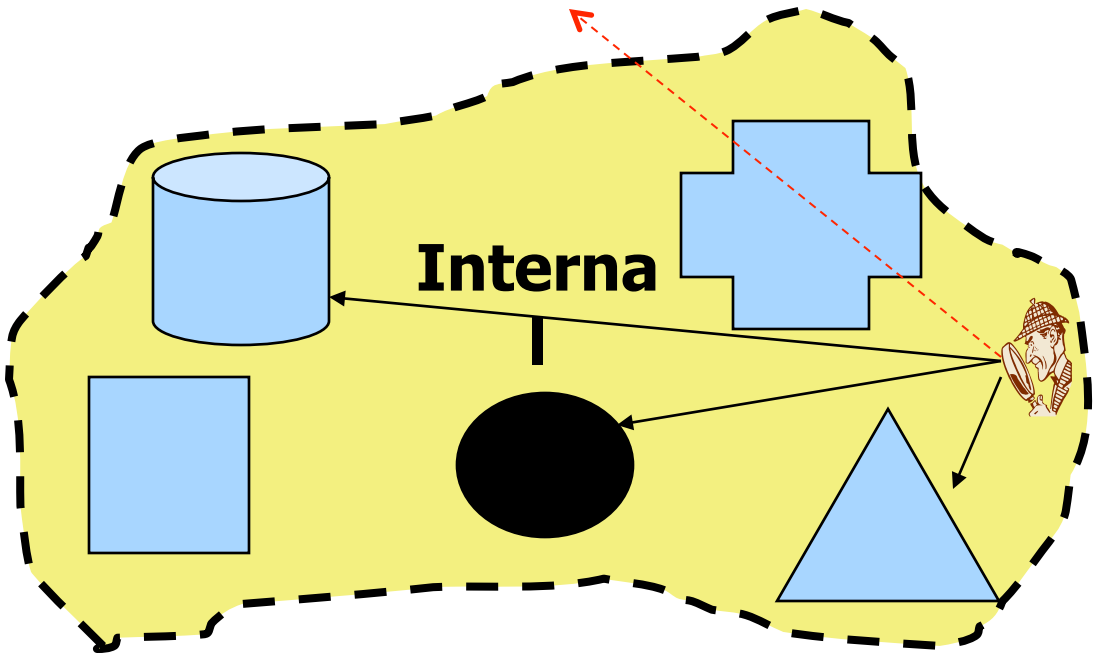
* Kasser, 2015, Holistic Thinking: creating innovative solutions to complex problems

Perspectives of a problem-1



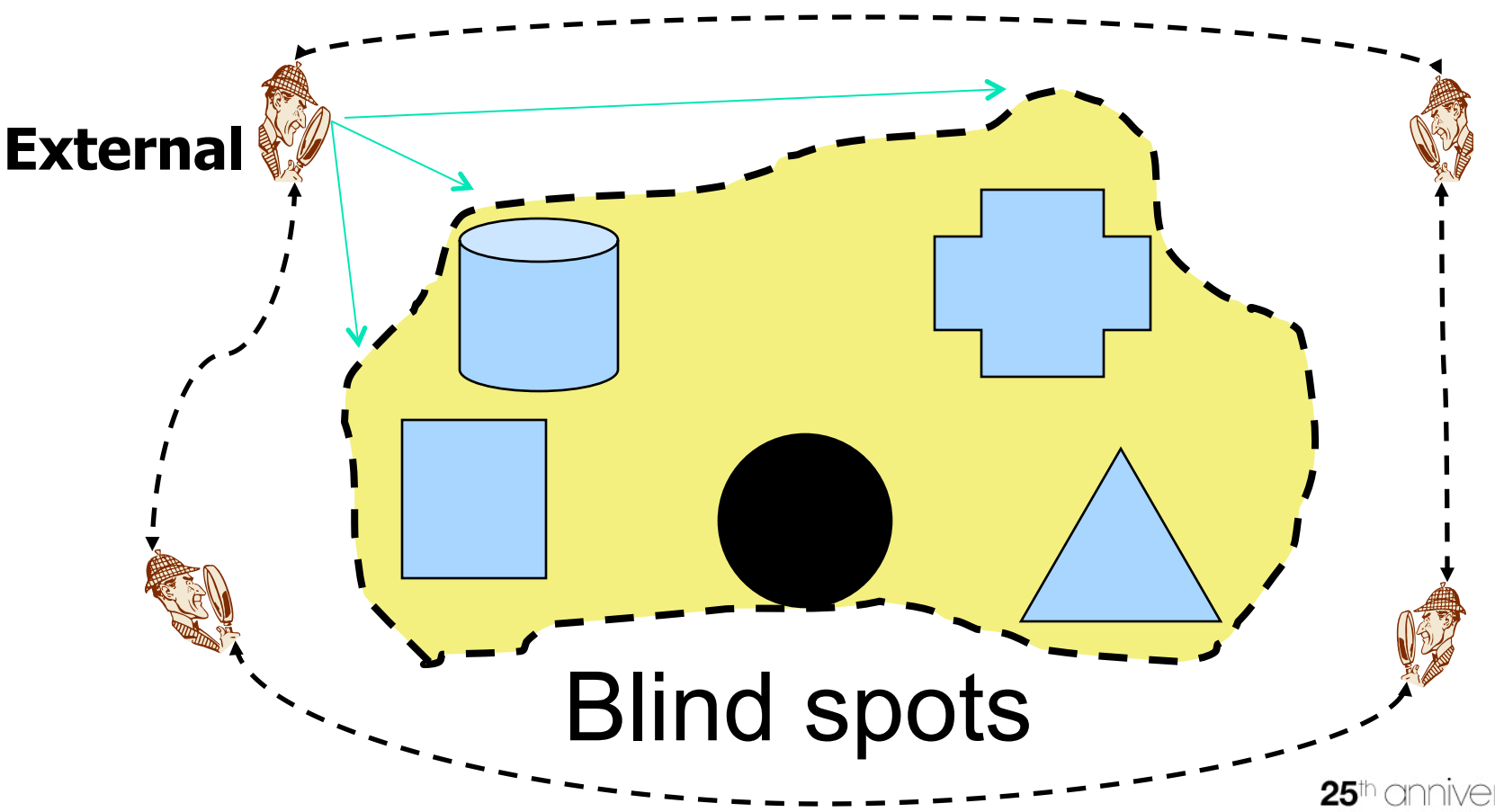
Blind spots

Perspectives of a problem-1a



Blind spots

Perspectives of a problem-2

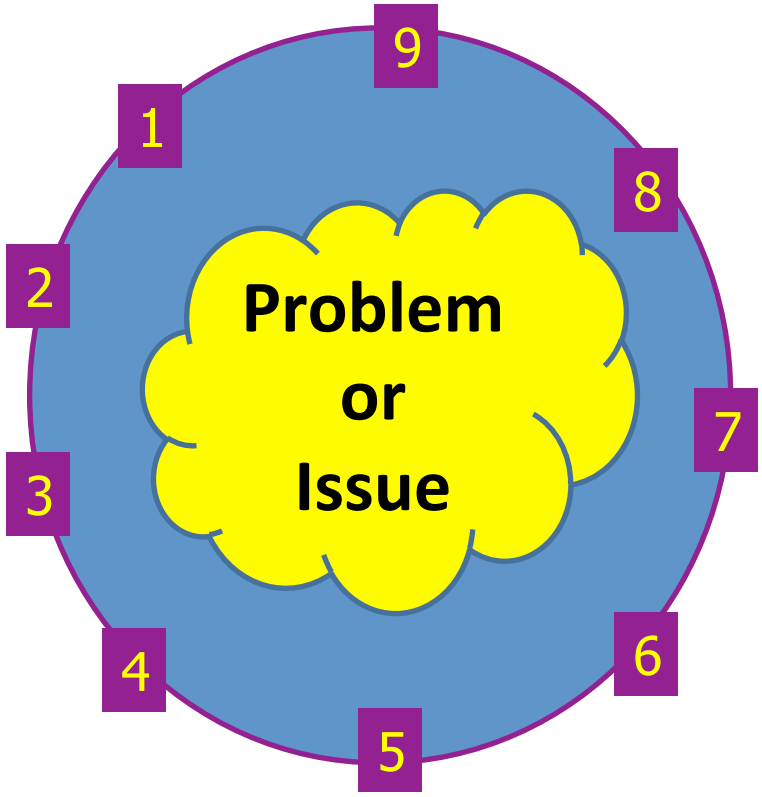


Limits of a single perspective*

* http://signature-strength.com/confidence/changing-perspective/, accessed 18 Nov 2013

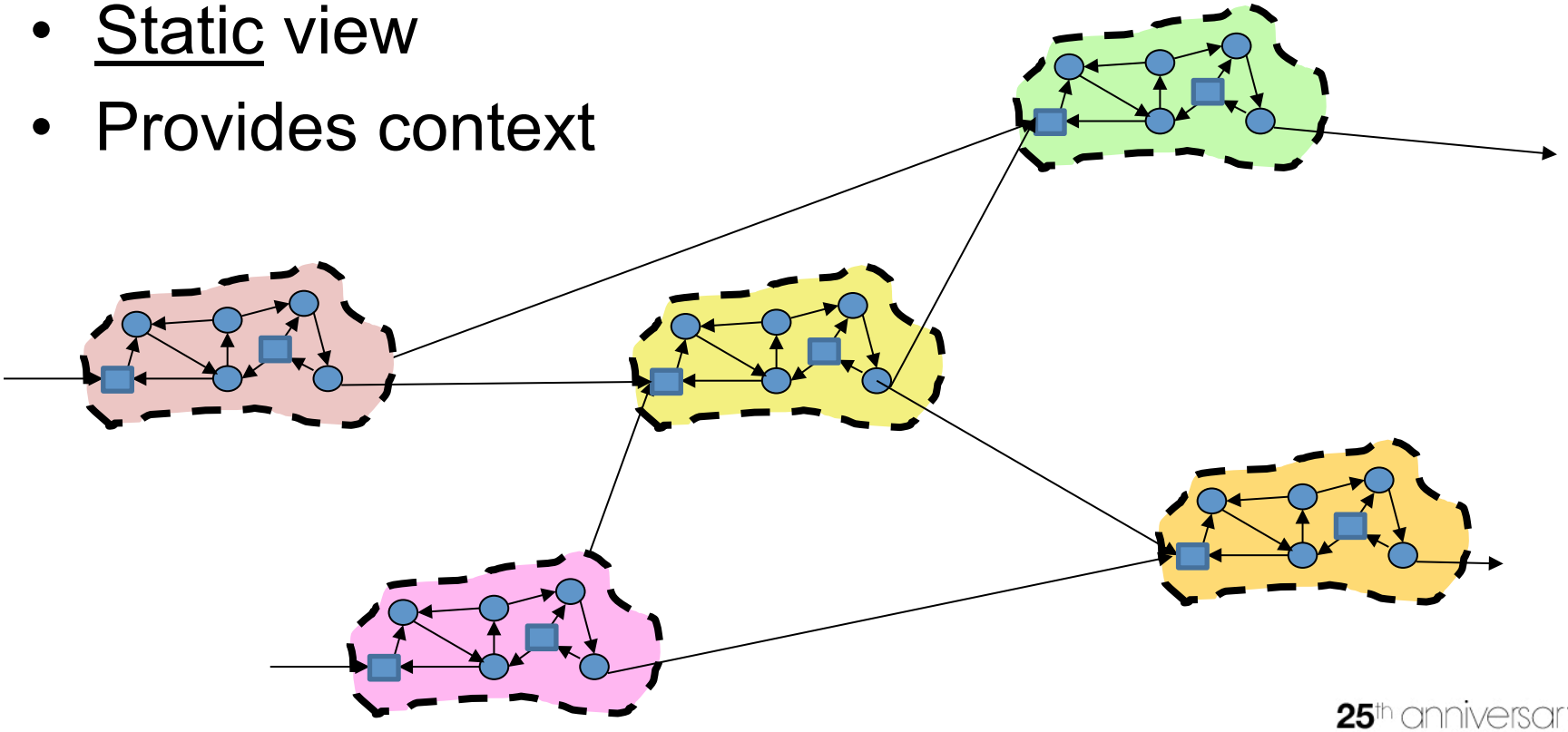
Holistic thinking perspectives*

1. Big picture
2. Operational
3. Functional
4. Structural
5. Generic
6. Continuum
7. Temporal
8. Quantitative
9. Scientific



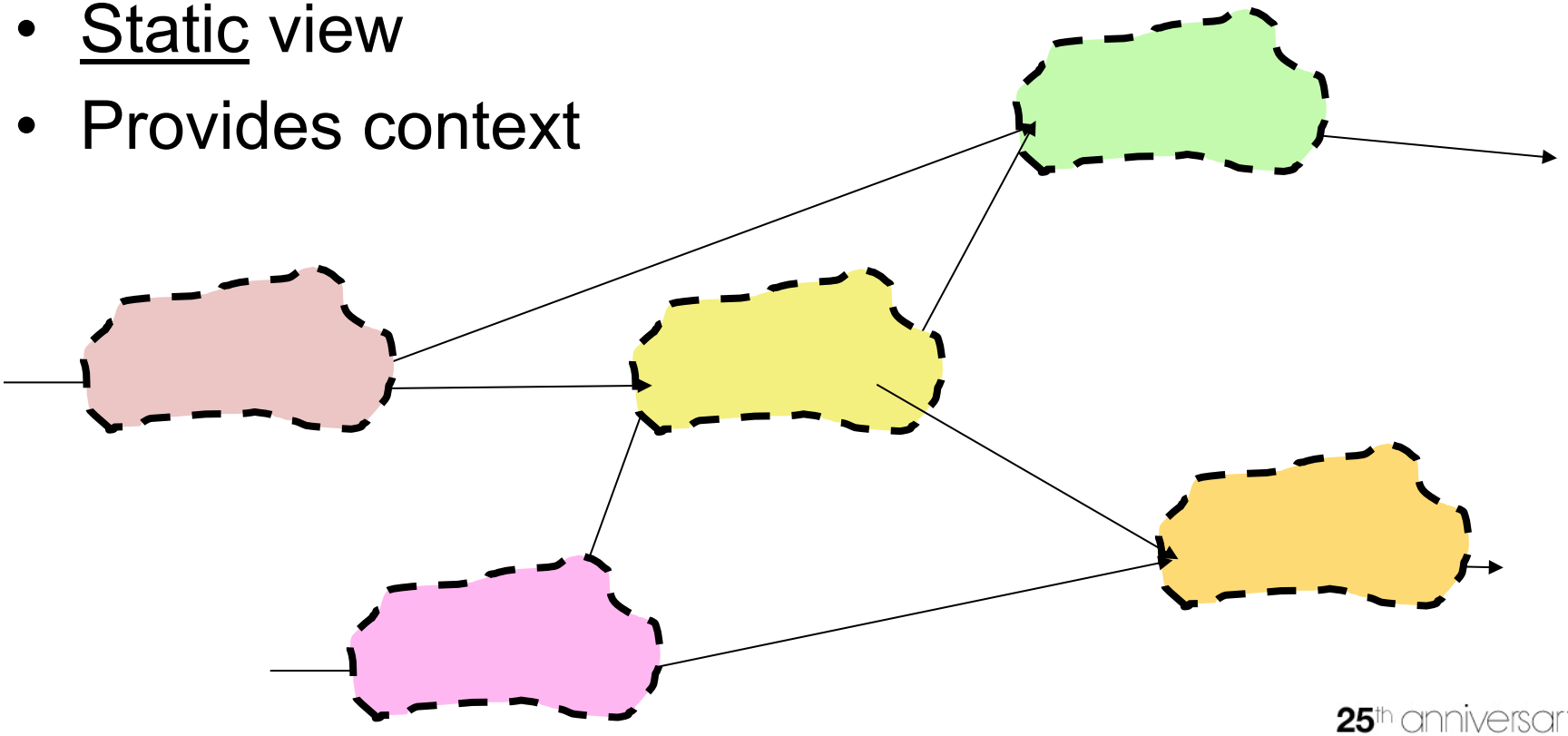
Big picture perspective (Complex)

- **External view**
  - Helicopter (Bird's eye)
- Static view
- Provides context

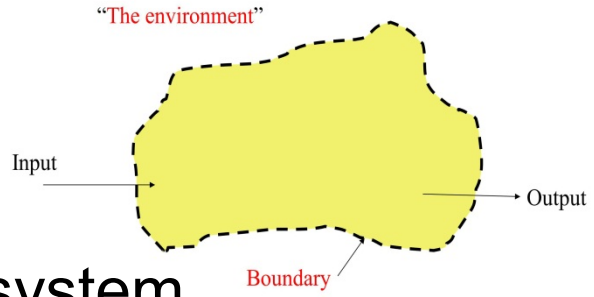


Big picture (abstracted complexity)

- **External view**
  - Helicopter (Bird's eye)
- Static view
- Provides context

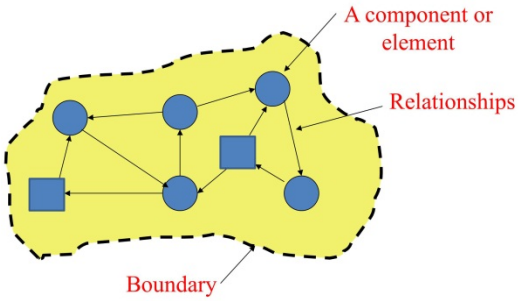


Operational perspective



- **External** or 'black box' view
- Dynamic view
- Missions/operations performed by the system
  - What the system does (in the big picture)
  - How the system is used
  - Scenarios or use cases
  - Causal/feedback loops of external interactions
  - Desired and undesired inputs and outputs
- Everything inside the system boundary is hidden in this view
- Can be a mixture of functional and physical

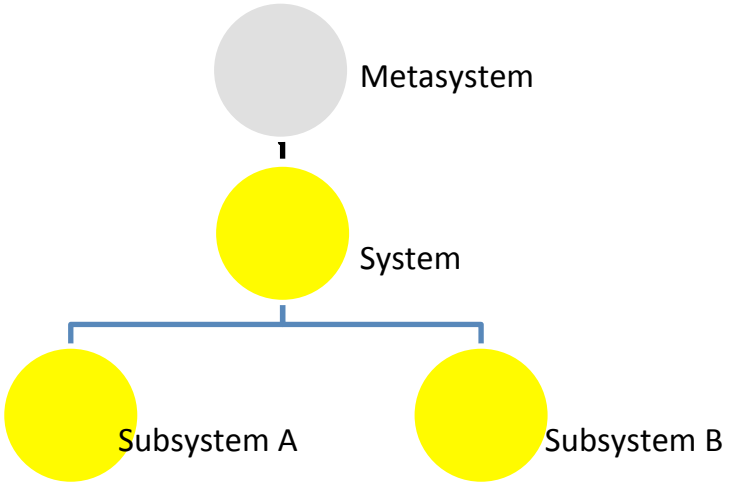
Functional perspective



- **Internal** or 'white box' view
- Dynamic view
- Functions performed by the system
  - How the system works
  - Causal/feedback loops of internal interactions
- Everything outside the system boundary is hidden in this view

Structural perspective

- **Internal** structure of system
- Static view
- Architecture
- Components
  - Technology
  - People
- Internal subsystem boundaries/partitions

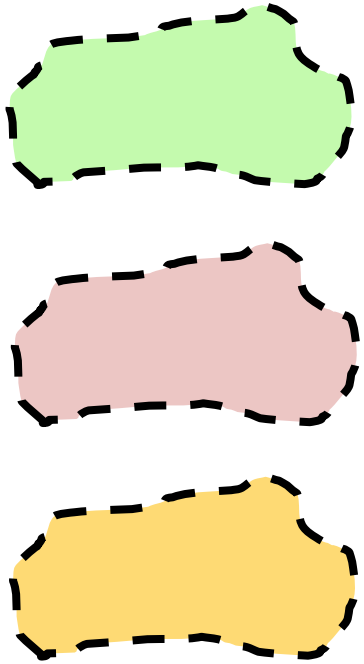


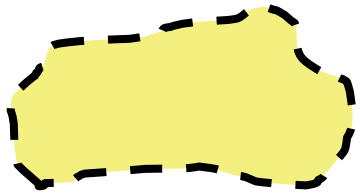
Temporal perspective

- How the system evolves over time
  - Patterns of behavior
    - prevention
  - Availability - Maintenance, Logistics
  - Obsolescence
  - Reflection on past
    - Lessons learned
- Current paradigm is a step in the staircase of history
  - opens mind to new thoughts
- Changes and their effects
  - Innovative and adaptive

Generic perspective

- Similarities to other systems
- System is an instance of a class/ type of system
  - Cargo ship is a surface ship
  - Car is land vehicle
- Patterns of behaviour
- Inheritance

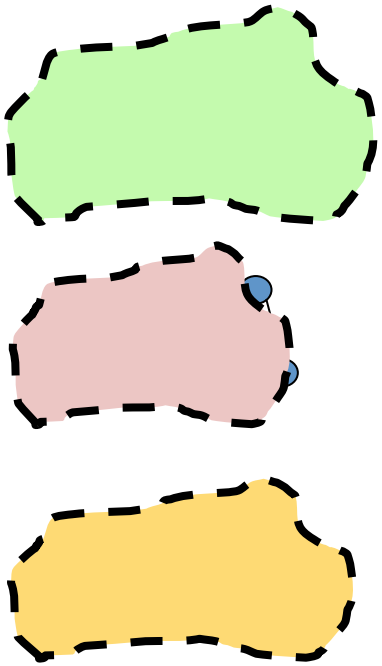


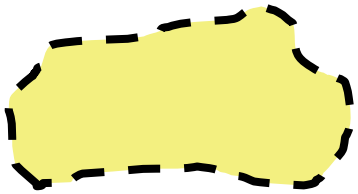


Inherits from …
Behaves like …
Looks like …

Continuum perspective

- Differences to other systems
- Range of solutions
  - Grey rather than black and white
    - Either/or solutions are only two points on a continuum of solutions





Colour is
different
Size
Shape

Quantitative perspective

- Numbers must be useful not necessarily perfect
- Numbers need not be absolute
  - Relative comparisons
- Quantification rather than measurement
- Helps to understand scope of relationships
- Provides values for parameters in models and simulations
- Ordering and ranking
  - Pareto principle

Scientific perspective

- Outcome of the analysis
- Hypothesis/guess
- Understanding of situation
  - unproven
- Statement of problem/issue
  - What needs to change
  - uncertain
- Vision of solution
  - Feasible Conceptual Future Desirable Situation (FCFDS)

Process/rules-2

1. Examine the undesirable situation from several different perspectives
2. **Develop an understanding of the situation**
3. Create the FCFDS containing the SOI
4. Use the principle of hierarchies to abstract out the complexity
5. Abstract out the parts of the situation that are not pertinent to the problem
6. Partition the FCFDS into the SOI and adjacent systems
7. Optimize the interfaces
8. Partition the SOI into subsystems

Develop an understanding of the situation

- The entities involved in the situation should have been identified
- Direct and indirect stakeholders
- The behaviour of the system
  - Can be understood from the information obtained from the relationships in the *Operational* and *Functional* perspectives
    - This information is often used to build a behavioural model
- The undesirable aspects (risks)
  - Tends to show up in the *Structural*, *Operational* and *Functional* perspectives
  - Should have been identified by discussions with the stakeholder and perhaps by analysis.
- The cause or causes of the undesirability
  - Should have been inferred (*Scientific* perspective) from the eight descriptive perspectives

Process/rules-3

1. Examine the undesirable situation from several different perspectives.
2. Develop an understanding of the situation
3. **Create the FCFDS containing the SOI**
4. Use the principle of hierarchies to abstract out the complexity
5. Abstract out the parts of the situation that are not pertinent to the problem
6. Partition the FCFDS into the SOI and adjacent systems
7. Optimize the interfaces
8. Partition the SOI into subsystems

Create the Feasible Conceptual Future Desirable Situation (FCFDS)

- Is a modified existing situation
  - Without the undesirability
  - With suggested improvements added.
- The system (SOI) and its adjacent systems will be subsystems of the FCFDS
- The boundaries of the different subsystems within the FCFDS may be different to the boundaries of the subsystems in the existing situation

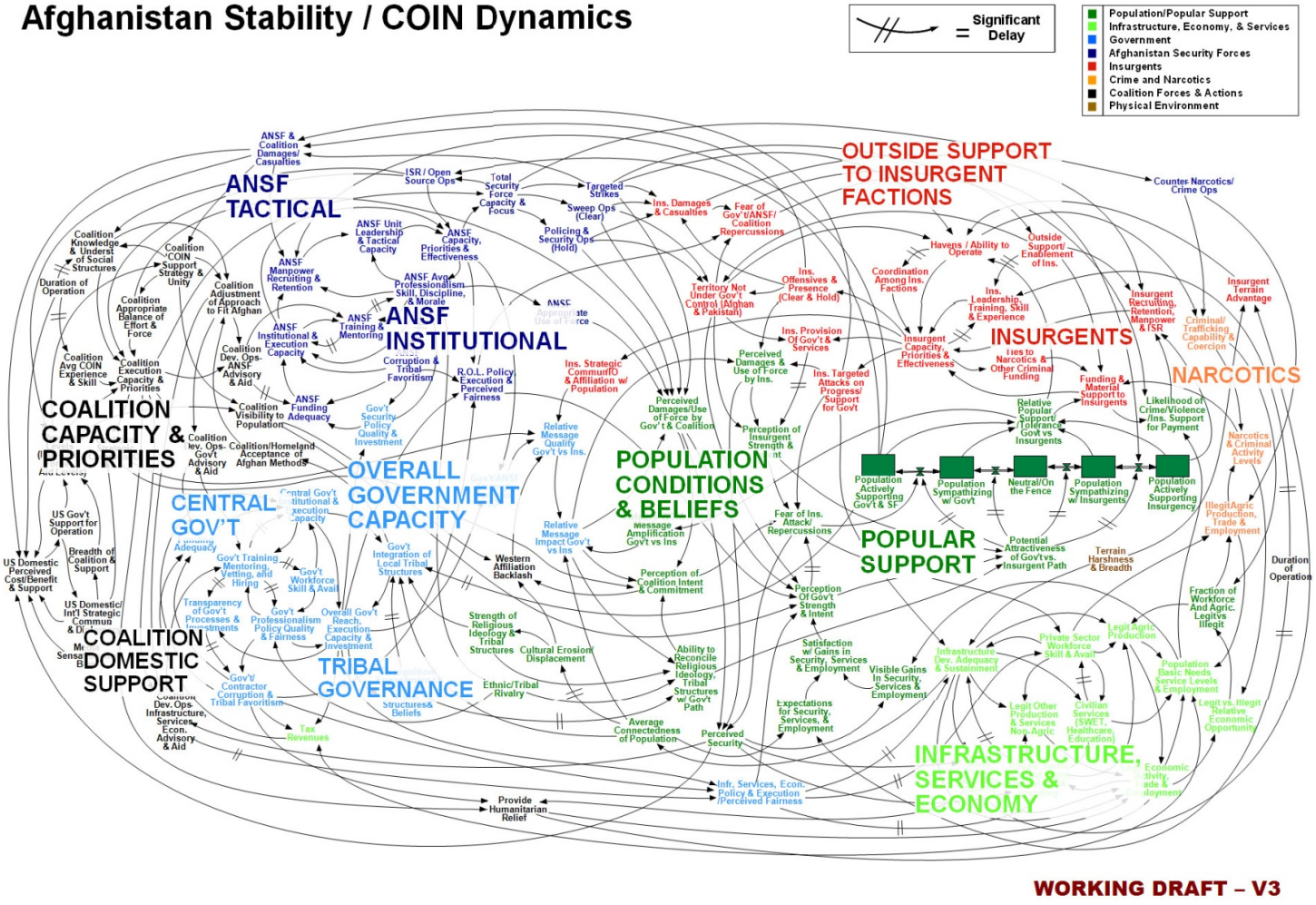
Process/rules-4

1. Examine the undesirable situation from several different perspectives.
2. Develop an understanding of the situation
3. Create the FCFDS containing the SOI
4. **Use the principle of hierarchies to abstract out the complexity**
5. Abstract out the parts of the situation that are not pertinent to the problem
6. Partition the FCFDS into the SOI and adjacent systems
7. Optimize the interfaces
8. Partition the SOI into subsystems

Use the principle of hierarchies

- Keep the systems and subsystems at the same respective level in the hierarchy of systems
- Abstract out or hide the internal components of systems and subsystems
- A situation contains a number of systems
- Each system may contain a number of subsystems
- Each subsystem may be further elaborated into a number of components
  - subsystems of the subsystem
- Risks are associated with level in hierarchy

Do not do this



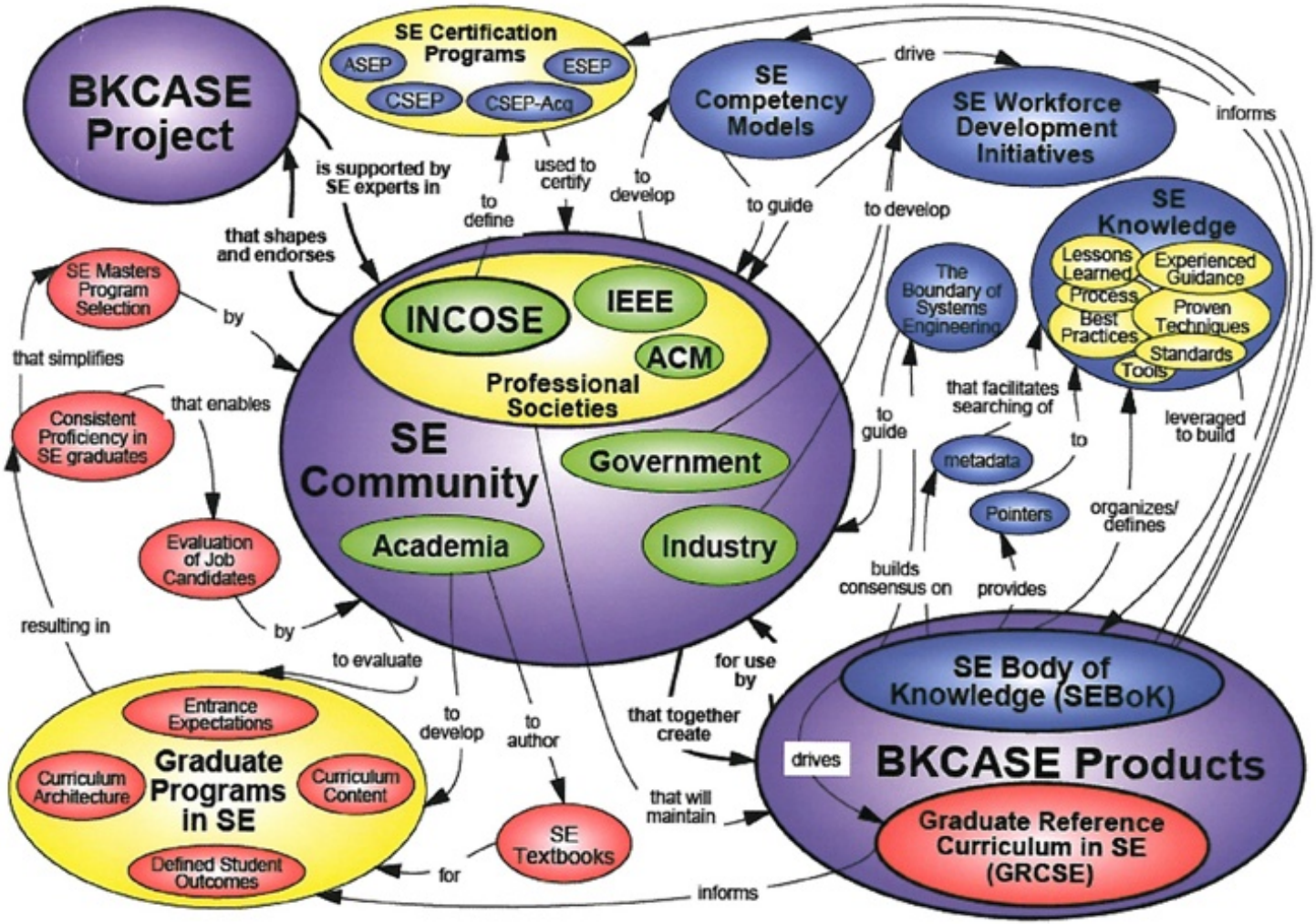
PA Consulting Group
© PA Knowledge Limited 2009

Page 22

"When we understand that slide, we'll have won the war," General McChrystal

* http://msnbcmedia.msn.com/i/MSNBC/Components/Photo/2009/December/091202/091203-engel-big-9a.jpg, accessed 8 April, 2011

Do not do this either



Process/rules-5

1. Examine the undesirable situation from several different perspectives.
2. Develop an understanding of the situation
3. Create the FCFDS containing the SOI
4. Use the principle of hierarchies to abstract out the complexity
5. **Abstract out the parts of the situation that are not pertinent to the problem**
6. Partition the FCFDS into the SOI and adjacent systems
7. Optimize the interfaces
8. Partition the SOI into subsystems

Abstract out the non-pertinent parts of the situation

- For the purpose of dealing with the problem
- Keep each abstracted view simple to facilitate its purpose,
- There is no single system view that represents the entire area of interest
- There are a number of views of the SOI
  - Each of them dealing with some aspect of the area of interest

Examples

- Docking a resupply vehicle to the International Space Station (ISS)
- A rock
- A camera

Docking in Space-1

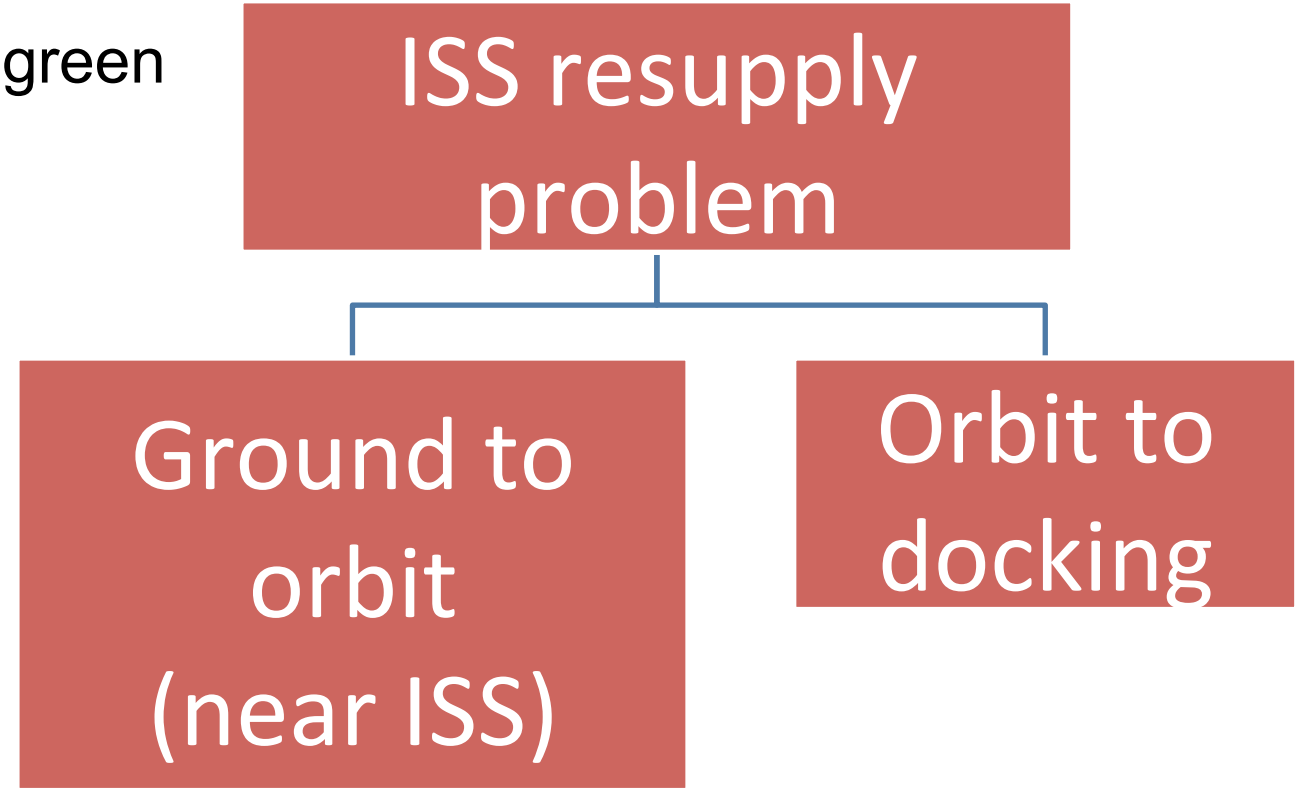
Docking in Space-2

- We construct a “closed” system
- We abstract out everything other than information pertinent to
  - Relative positions of the spacecraft
  - Relative velocity
  - Relative alignment in X, Y and Z orientation

ISS Problem Breakdown Structure

***Generic* HTTP:** Golf

1. Hole in one
2. Hole in two via green



A rock

- Simple chemical system
- Views depend on problem
  - ***Sight:*** one looks at its colours
  - ***Taste:*** taste might give us some information about the chemicals in the rock
  - ***Weight/mass:*** might tell us something about its composition
  - ***Touch:*** the surface texture might be of interest
  - ***Chemical analysis:*** the components might be of interest
  - ***Radiation:*** could tell us something

A camera-1

- The device that takes the photograph
  - *Structural* and *Functional* perspectives
  - System boundary
    - the camera
- The act of taking the photograph
  - *Operational* perspective
  - System boundary
    - includes the camera and photographer
- Transporting the camera
  - *Operational* perspective
  - System boundary
    - includes the camera and transportation elements including the carrying case

A camera-2: your choice

- Create unnecessary complexity by
  - Developing one representation that includes all the elements for photographing and transportation
  - Requiring the elements under consideration for a specific situation to be
    - Abstracted out of the representation
    - Ignored in the representation
- Use three separate simpler views
  - Abstracted out of the real world for understanding the various aspects of the use of a camera

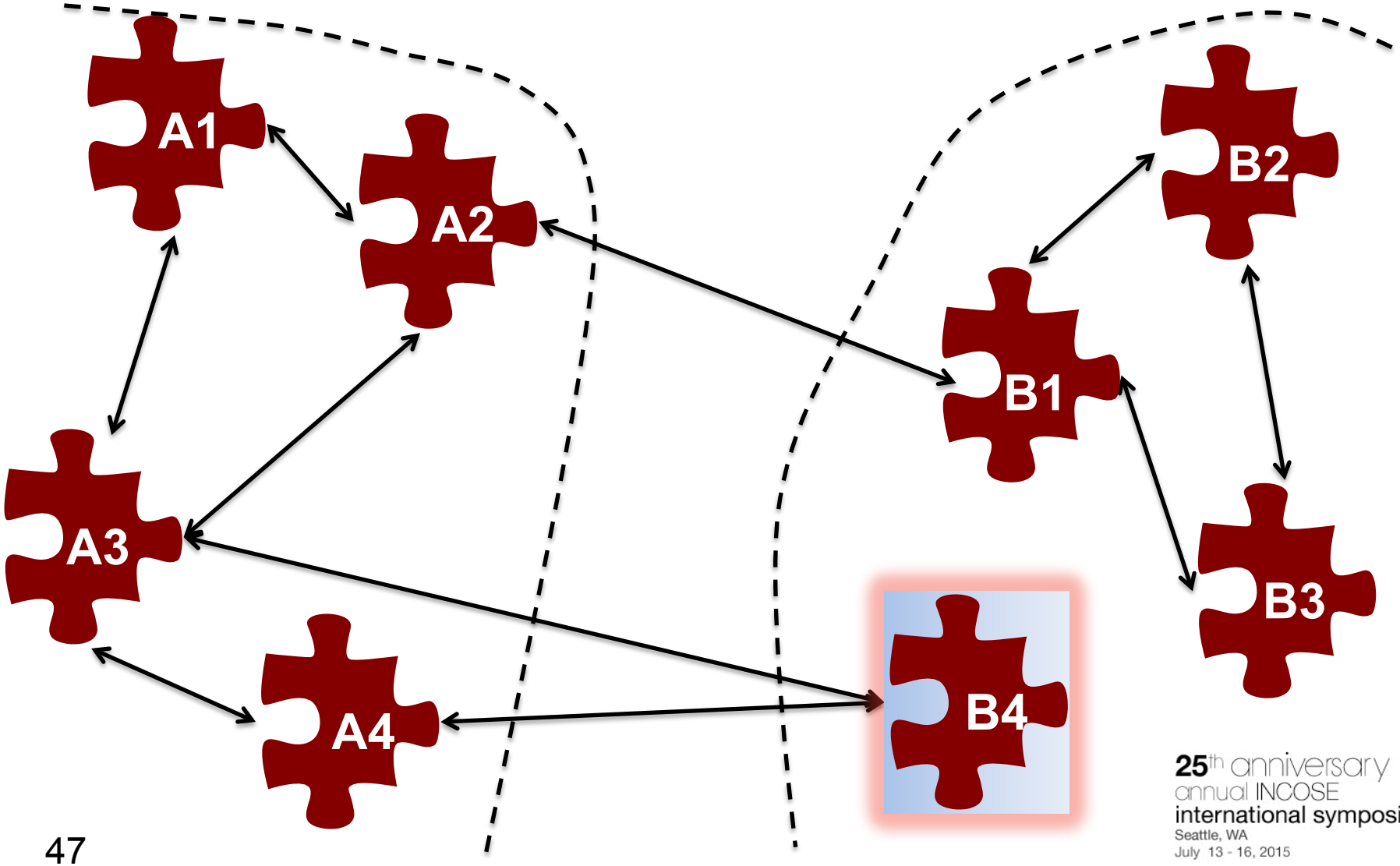
Process/rules-6

1. Examine the undesirable situation from several different perspectives.
2. Develop an understanding of the situation
3. Create the FCFDS containing the SOI
4. Use the principle of hierarchies to abstract out the complexity
5. Abstract out the parts of the situation that are not pertinent to the problem
6. **Partition the FCFDS into the SOI and adjacent systems**
7. Optimize the interfaces
8. Partition the SOI into subsystems

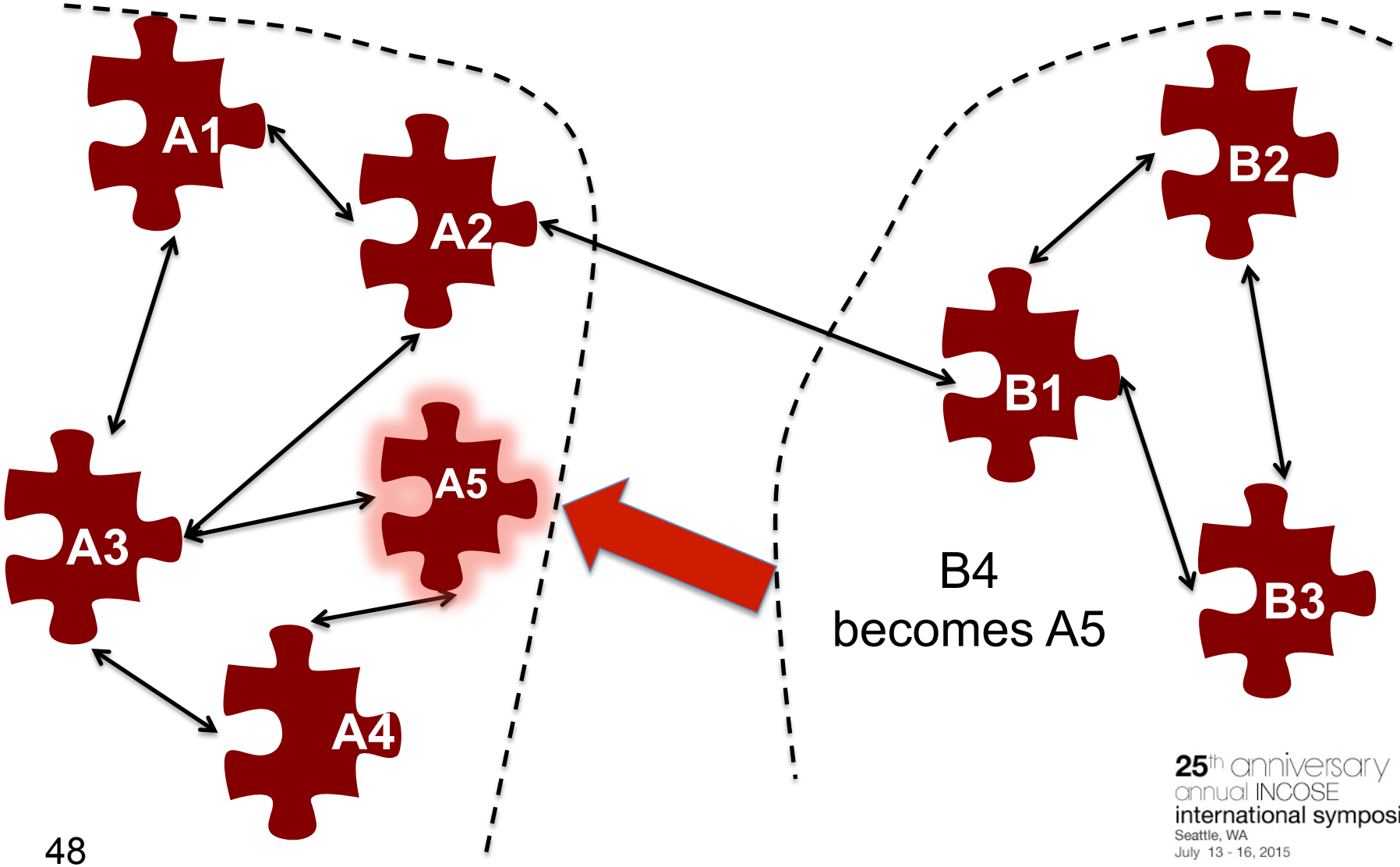
Partition the FCFDS into the SOI and adjacent systems

- It is the act of drawing the system boundary that creates the system (Beer, 1994; Churchman, 1979) page 91).
- When the undesirable situation already contains a SOI, then the existing SOI tends to be the starting point for creating a new SOI
  - Feel free to examine alternatives
- Rules for performing aggregation of entities into SOI
  1. Keep number of subsystems at any level to less than 7±2
  2. Configure each subsystem for the maximum degree of homeostasis
  3. Maximize the cohesion of the individual subsystems, minimize the coupling between subsystems

Coupling and cohesion (poor?)



Coupling and cohesion (better)



Process/rules-7

1. Examine the undesirable situation from several different perspectives.
2. Develop an understanding of the situation
3. Create the FCFDS containing the SOI
4. Use the principle of hierarchies to abstract out the complexity
5. Abstract out the parts of the situation that are not pertinent to the problem
6. Partition the FCFDS into the SOI and adjacent systems
7. **Optimize the interfaces**
8. Partition the SOI into subsystems

Optimization paradox*

- *The principle of suboptimization states that*
  - *optimization of each subsystem independently will not lead in general to a system optimum*
  - *improvement of a particular subsystem actually may worsen the overall system*
- *Since every system is merely a subsystem of some larger system, this principle presents a difficult if not insoluble problem*
  - *one that is always present in any major systems design*

* Machol and Miles Jnr, 1973: page 39

Optimize the interfaces

- Iterative step with aggregation
- Minimize interaction between subsystems at interfaces
  - Ideally a single interface between entities
    - Coupling and cohesion
  - Hierarchies
    - 7±2 subsystems at any level in hierarchy
      - Miller's Rule

Optimizing interfaces: examples

- Optimizing your sex life
- Weapons systems (tanks)
- Logistics systems
- The Apollo Program
- Resupplying the MIR space station
- The human cardiovascular system
- A distance-learning classroom
- The Library
- Forming the INCOSE Australia chapter

Weapons systems (tanks)

- ***The undesirable situation in WW I***
  - is the inability to break through the enemy front line trenches swept by machine gun fire
  - according to lessons learned from experience
    - precluded the traditional infantry or cavalry charge from performing the function
    - so that infantry and cavalry could then be used in their traditional manner to route the enemy after a breakthrough
- ***The FCFDS***
  - a break through into the enemy front line trenches by the application of yet-to-be-developed technology
- ***The problem***
  - provide a solution to create the FCFDS
- ***The solution***
  - was unknown at the time the problem was formulated

Apollo program-1

- Optimized to transfer men and ALSEPs between the earth and the moon
  - in the most efficient manner within the constraints of the then available technology.
- From the *Structural* perspective
  - The system contains three top-level physical subsystems
    1. the earth
    2. the lunar
    3. the interface system between the earth and lunar subsystems

Apollo program-2

- ***The earth subsystem***
  - The NASA manned spacecraft centers and headquarters
- ***The lunar subsystem***
  - Empty before the first landing
  - Contained an increasing number of Apollo Lunar Surface Experiments Packages (ALSEP), the set of scientific instruments deployed by the astronauts at each of the landing sites
  - Two astronauts while on they were on the lunar surface
- ***The interface subsystem***
  - The spacecraft
  - The astronauts (three while in transit, one when in lunar orbit)
  - The NASA Communications Network (NASCOM) communications subsystem

Forming the INCOSE Australia chapter

- The undesirable situation
  - The Memorandum of Understanding (MOU) between the INCOSE and the Systems Engineering Society of Australia (SESA) expired in 2004
  - The majority of members of SESA attending its annual general meeting voted that SESA not become a chapter of INCOSE and remain an independent organization
  - There was also a desire and support for a Chapter of INCOSE in Australia
  - Feelings were running high on the issue
- The FCFDS
  - A single professional organisation for systems engineers in Australia
    - What the overwhelming majority of Australian systems engineers wanted
- The problem
  - Create the FCFDS

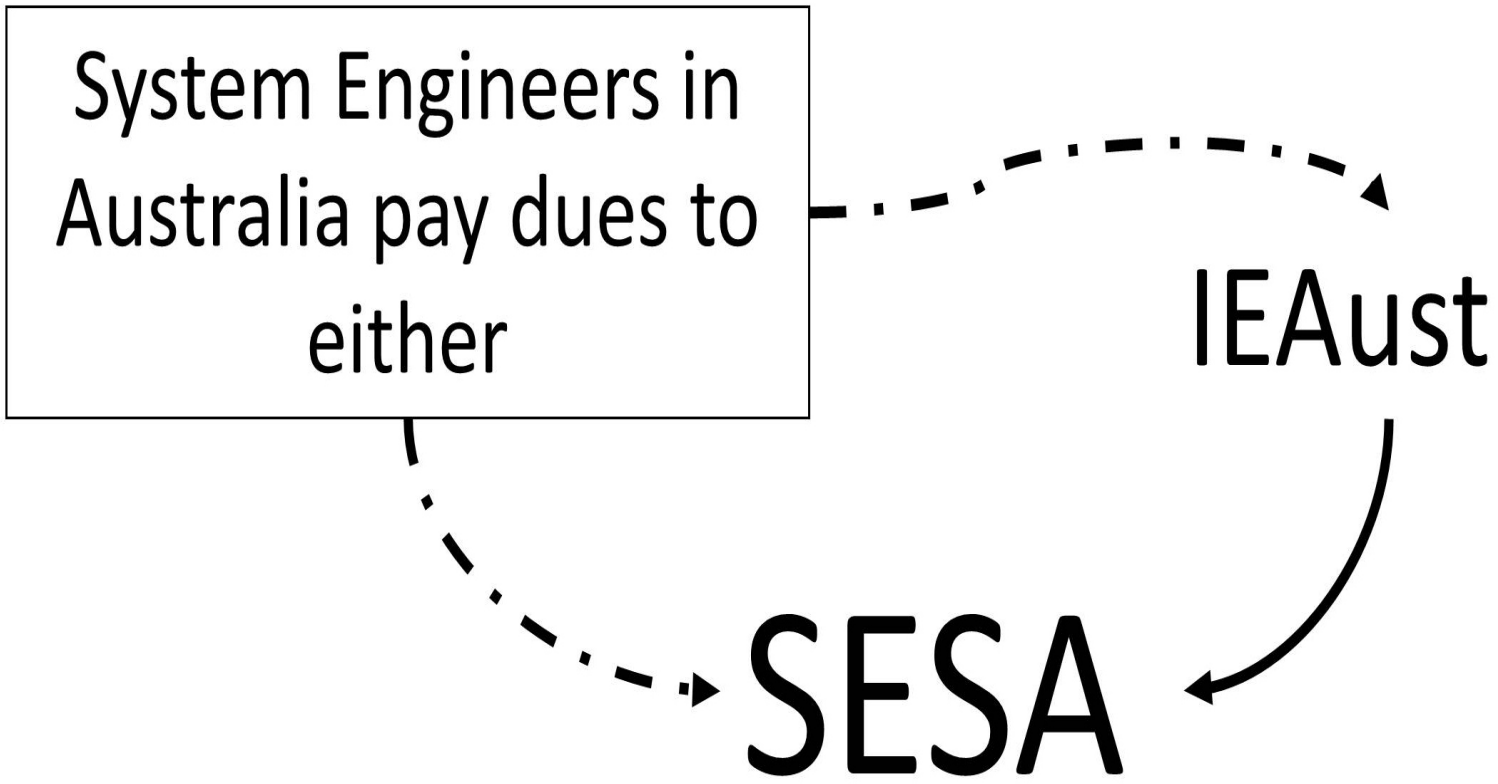
The innovative solution

- Constitute the chapter of INCOSE in Australia, INCOSE-Australia as *a special interest group within SESA*.
  - Avoided a “civil war” within the systems engineering profession in Australia
  - Meant that nobody could join INCOSE-Australia without being a member of both INCOSE and SESA
  - Allowed those SESA members who desired INCOSE services and products to obtain them without having to join two professional societies
  - Allowed those systems engineers that did not desire the INCOSE products and services to be part of SESA

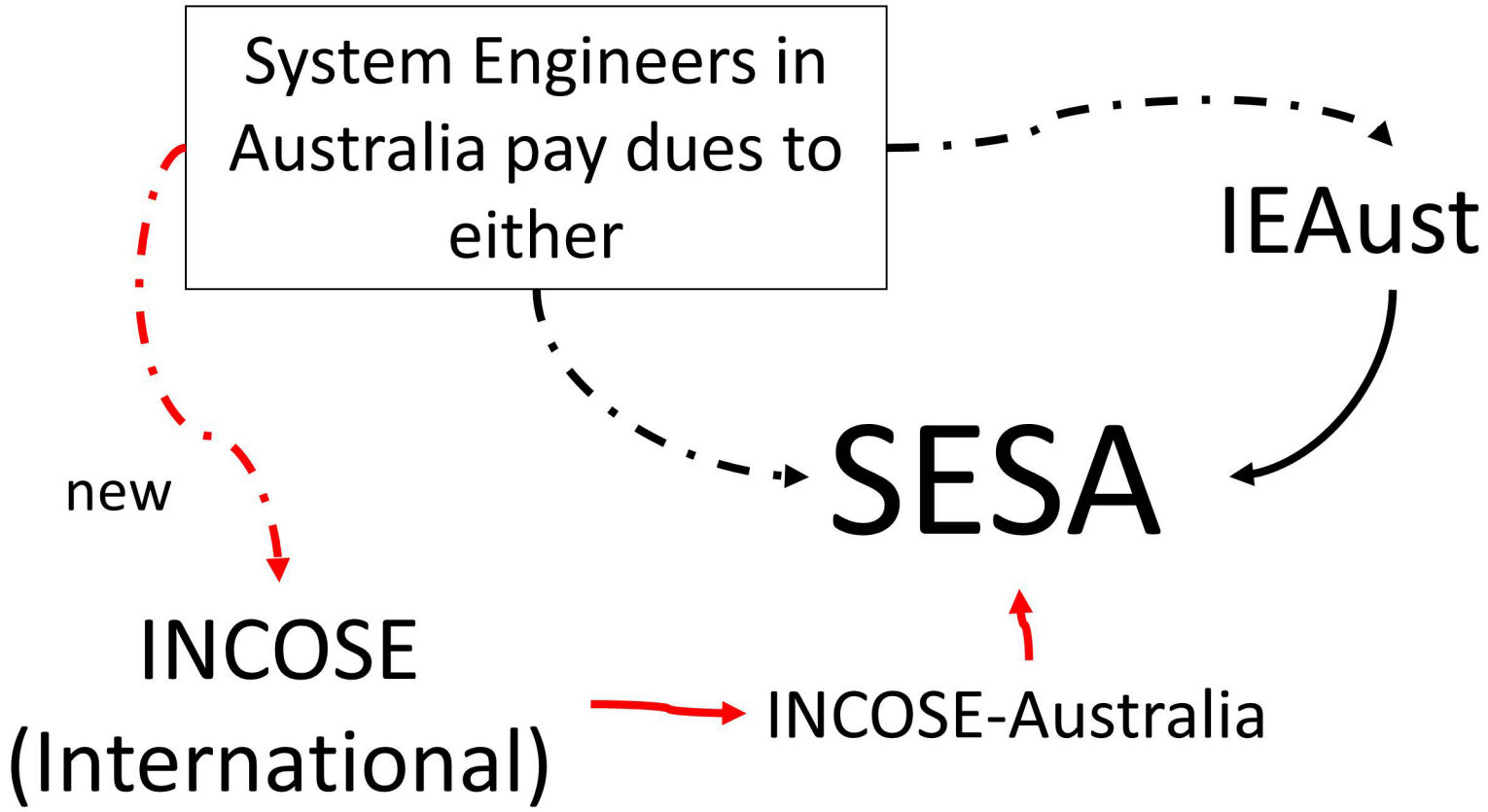
Three functional subsystems

- The innovative solution was made possible by considering SESA as containing the following non-traditional three functional subsystems
  1. INCOSE Australia which constituted the members of SESA who were also members of INCOSE
  2. The remaining non-INCOSE membership of SESA
  3. The SESA Headquarters which received the dues payment from INCOSE

SESA dues payment process



Modified SESA dues payment process



Process/rules-8

1. Examine the undesirable situation from several different perspectives.
2. Develop an understanding of the situation
3. Create the FCFDS containing the SOI
4. Use the principle of hierarchies to abstract out the complexity
5. Abstract out the parts of the situation that are not pertinent to the problem
6. Partition the FCFDS into the SOI and adjacent systems
7. Optimize the interfaces
8. **Partition the SOI into subsystems**

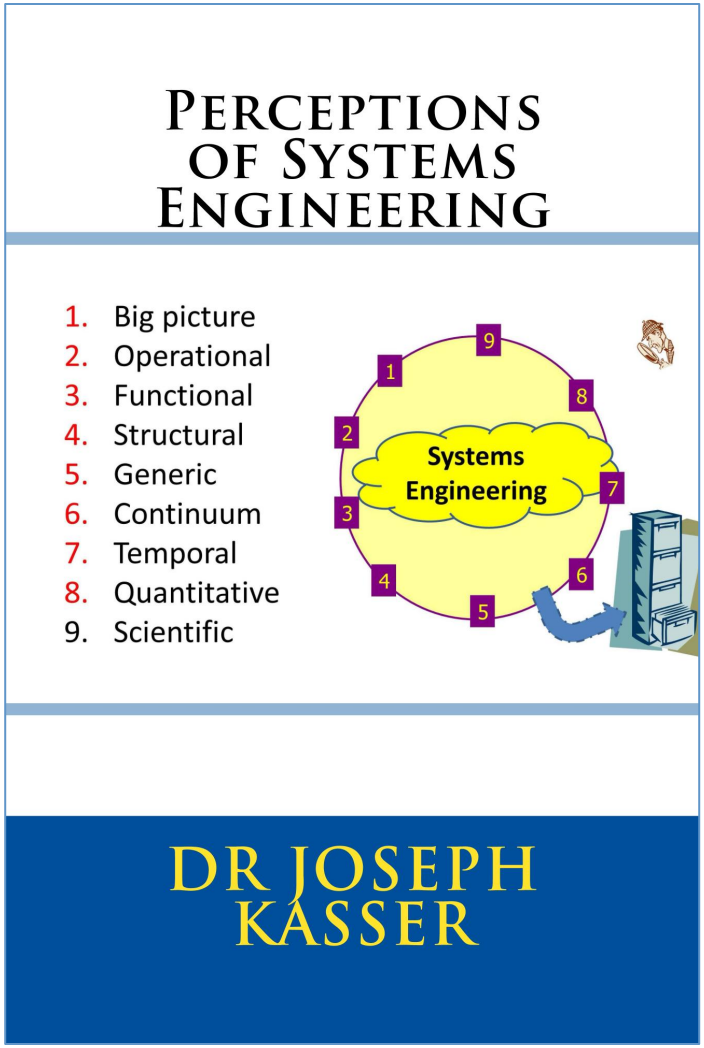
Partition the SOI into subsystems

- Using the same previous seven steps
- The Metasystem was partitioned into the SOI and its adjacent systems by the Metasystem system engineer
- The SOI is partitioned into subsystems by the SOI systems engineer
- One systems engineer's subsystem is another systems engineer's system in the hierarchy of systems

Summary

- Mind the gap
- The problem
- Rules for creating systems
- Summary
- Questions and comments

Questions and comments?



- This presentation is based on an updated version of the paper in the proceedings
  - See *Perceptions of Systems Engineering* Chapter 18
  - Printable Desk pdf version can be found on "INCOSE - International Council on Systems Engineering" Facebook web site