Module 8: The Subsystem Realization States
Session 1 of 2

Rev 1.14.0

Objectives

- For these states of the SDP
  1. To identify
     - the role of systems engineers
     - the nature of the problems they face
  2. To introduce
     - the tools, methodologies and techniques available to solve those problems.
  3. To creating functional and physical architectures

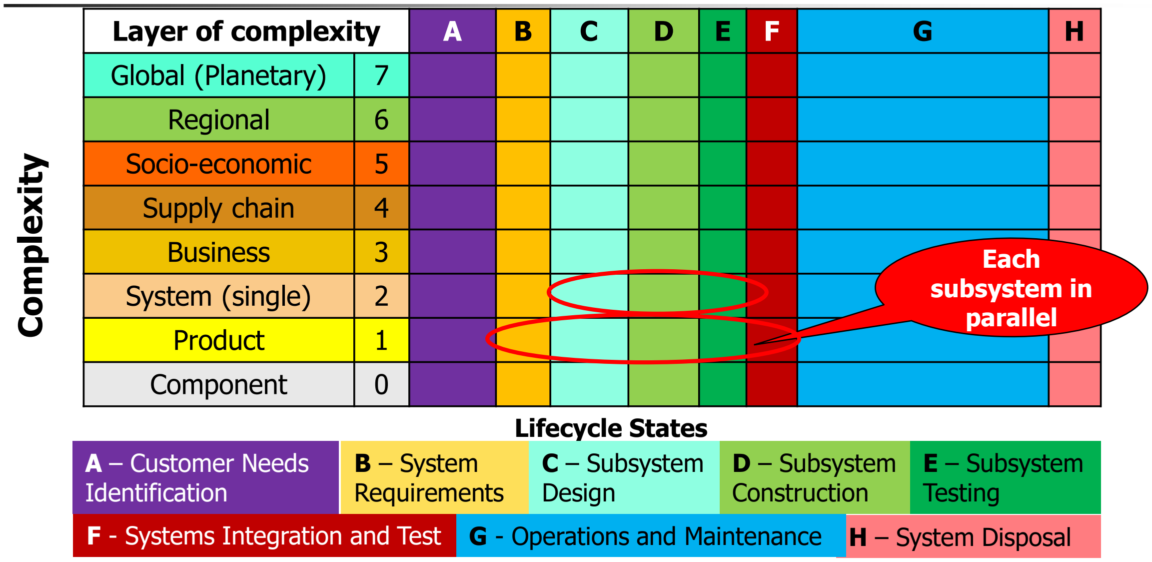
Content

1. The role of systems engineers in the realization (design, construction and subsystem testing) states of the SDP
2. The nature of the problems faced in the states
3. The tools, methodologies and techniques available to solve those problems
4. The critical applied systems engineering tasks which systems and software engineers must perform in these states
5. Factors to consider and monitor in the design for performance, cost, reliability, integration, test, maintainability and safety
6. How the best systems engineers perform problem-solving across subsystem boundaries
7. Interface and change management

Knowledge component

- Lecture
  - Sets the context
- Readings
  - 0802 Systems Engineering Chapter 18, Luz: From Light to Darkness: Lessons learned from the solar system (FUSE Chapter 22 Earlier version)
  - 0804 Systems Engineering Chapter 11: The system and subsystem design states
  - 0805 Systems Engineering Chapter 12: The subsystem construction state
  - 0806 Systems Engineering Chapter 13: The subsystem test states
  - 0807 Systems Engineering Chapter 20, Sections 20.4 and 20.5
  - Alternatives
    - 0809 Applying TQM to Systems Engineering Chapter 16 The Preliminary Design Phase
    - 0810 Applying TQM to Systems Engineering Chapter 17 The Design, Build, Integrate and Test Phases
- Exercises

Big Picture

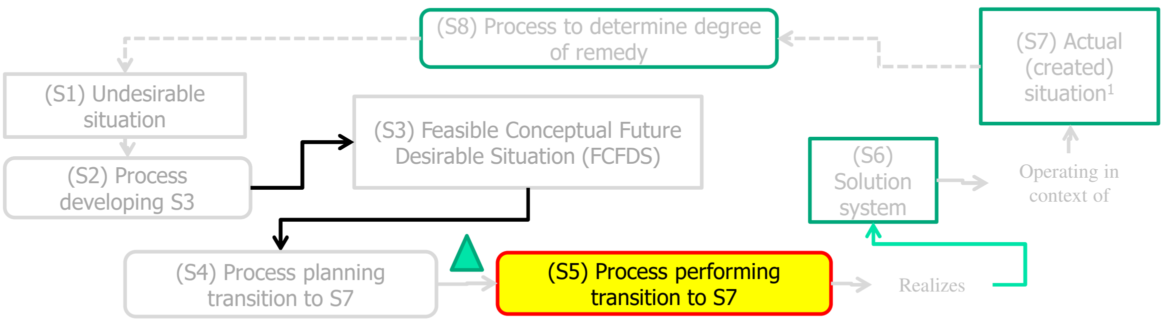


Topics

1. **The Subsystem Realization States**
2. An awareness of the factors involved in functional and physical partitioning of a system
3. Analysis for determination of feasibility
4. Factors to consider and monitor in the design for performance, cost, reliability, integration, test, maintainability and safety
5. Design optimisation
6. Problem solving across subsystem boundaries.
7. Change management
8. Exercises

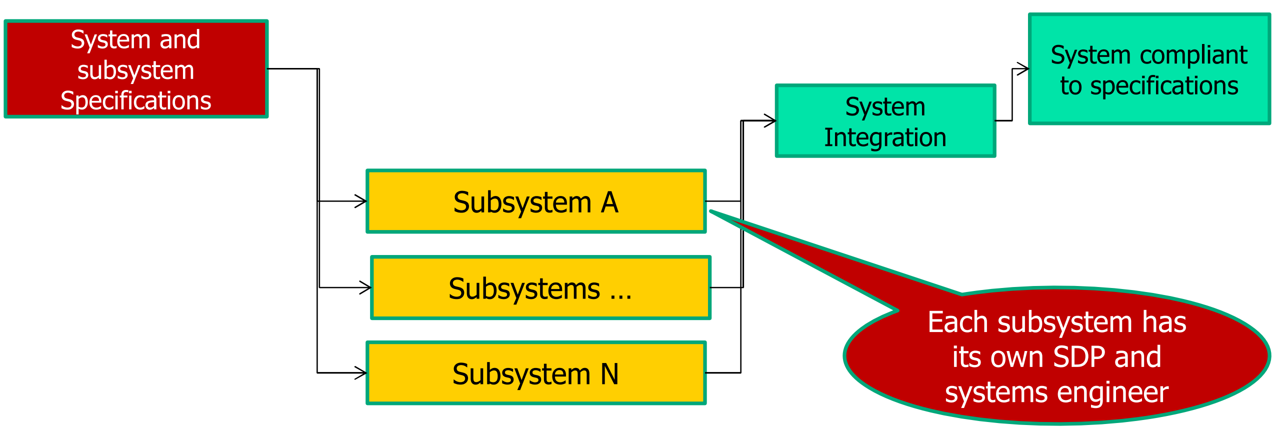
The Nine-System model (Functional view)

The realization states



Creating Outstanding Systems Engineers 8-7

SDP phased activities (HKMF C-E)



Creating Outstanding Systems Engineers 8-8

Subsystem Construction State

1. ***The undesirable situation***
   - The lack of a set of subsystems, that when constructed, in isolation, meet the individual specifications
   - The uncertainty that the system as a whole will meet its specification
2. ***Assumptions***
   - Each subsystem has its own systems engineer
3. ***The FCFDS***
   - A set of subsystems, constructed, in isolation, meeting the individual specifications and the system as a whole meeting its specification
4. ***The problems***
   - To ensure the system as a whole meets, and the set of constructed subsystems meet,
     1. The system and subsystem specifications
     2. Accepted changes to the specifications during the Design and Subsystem Construction States (the evolved undesirable situation)
5. ***The solution at TRR***
   - The FCFDS

The Subsystem Construction State

- Starts at CDR
- Creates the individual parts, subsystems, ~~interactions,~~ etc. *in isolation* but monitored by the systems engineer
- The role of the system systems engineer is to ensure
  - That the parts or subsystems contribute to the whole or system
  - Total system budgets are not exceeded even if subsystem budgets are
  - When completed and integrated, the system will remedy the undesirable situation that will exist when the system is placed into service
- Ends at the (subsystem) Test Readiness Review (TRR)
  - when the subsystems are ready for testing

Subsystem Test State

1. ***The undesirable situation***
   1. The uncertainty that each of the subsystems, in isolation, is compliant to its requirements
2. ***Assumptions***
   1. None
3. ***The FCFDS***
   1. The certainty that the complete set of subsystems have been validated, in isolation, as well as being compliant to their requirements
4. ***The problem***
   1. To ensure the set of subsystem tests validate that each of the subsystems, in isolation, is compliant to its requirements and meets the customer's needs
5. ***The solution*** is the FCFDS
   1. Note that subsystem testing may continue after the IRR should the integration be phased as long as the subsystem-testing for a subsystem is complete before that subsystem is to be integrated

The Subsystem Test State

- Begins with the TRR
  - Subsystems ready to be tested
- Validates the performance of the individual parts, subsystems, etc. *in isolation* against their requirements
- Ends at the Integration Readiness Review (IRR)
  - When the subsystems have passed subsystem testing and are ready to be integrated into a system

Developing a solution

- **OARP** provided problem statement
- **Conversion from problem statement to answer (solution)**
- Conversion from requirement to design
  - **FRAT** the iterative systems engineering methodology
  - **SPARK** documents the solution implementation
    - SEMP and TEMP

System functionality

- Systems functionality is achieved through partitioning (systems design) by a combination of
  - Functions of each subsystem
  - Emergent properties of system
- Functions (functional requirements) – what the system is supposed to do
- Functional Requirements (performance requirements) – how well the function needs to be done
- A requirement is a function quantified with numbers
- Functional Measures of Performance (MOP) – how well the function is being performed

If requirement is >= 6, 5 does not meet it, 7 does

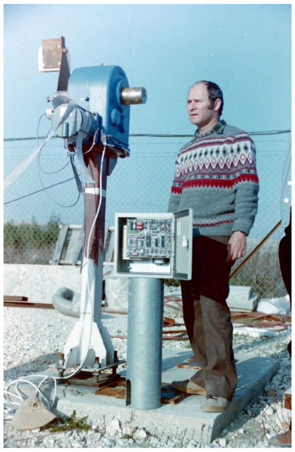
Artist's conception of SEGS-1

Creating Outstanding Systems Engineers 8-15

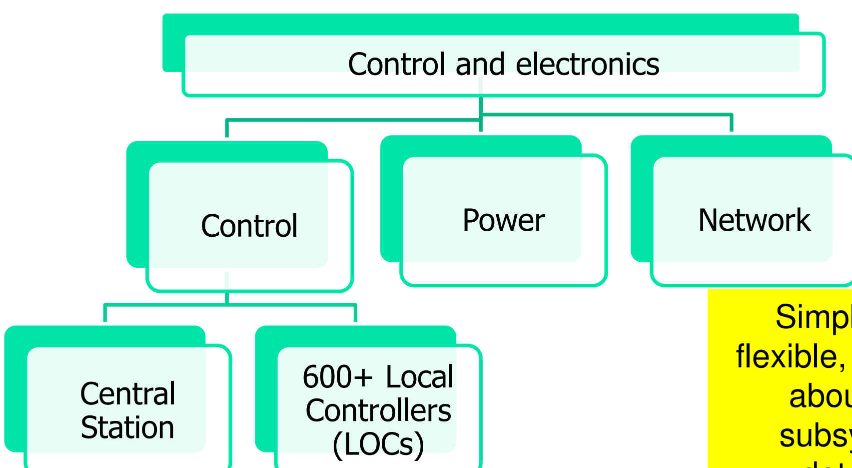
Luz SEGS-1 during installation

Creating Outstanding Systems Engineers 8-16

Size it



System Architecture



Simple and flexible, but what about the subsystem details?

Requirements

- System level
  1. In operation, SEGS-1 shall generate more power than it uses
  2. SEGS-1 shall generate the maximum possible amount of power each day
- Control subsystem level
  1. Mirror pointing accuracy = ±0.2 degrees

Functions at the system level

- Deploying the entire array of mirrors (>600)
  - when $Power_{(generated)} > Power_{(used)}$
- Tracking when and while the sun shines
  - idling for periods of scattered cloud cover
- Stowing the array
  - when $Power_{(generated)} < Power_{(used)}$
- Gathering, displaying and storing status information about the operation of SEGS-1
- Interfacing with the operator

Functionality in LuZ LOC

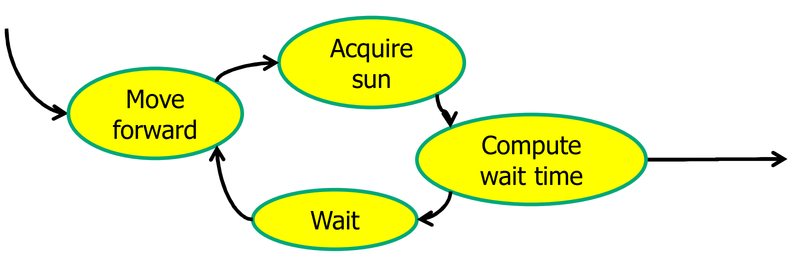
Resting	X (Cmd)				
	Deploying	X (Auto)		X (Cmd)	X (Cmd)
		Tracking	X (Auto)	X (Cmd)	X (Cmd)
	X (Cmd)	X (Auto)	**Idling**	X Auto/Cmd)	X (Cmd)
	X (Cmd)			**Waiting**	X (Cmd)
X (Auto)					**Stowing**

Cmd.
Automatic
C/S or
Operator

Functionality in LuZ LoC

Resting	X (Cmd)				
	Deploying	X (Auto)		X (Cmd)	X (Cmd)
		Tracking	X (Auto)	X (Cmd)	X (Cmd)
	X (Cmd)	X (Auto)	**Idling**	X Auto/Cmd)	X (Cmd)
	X (Cmd)			**Waiting**	X (Cmd)
X (Auto)					**Stowing**

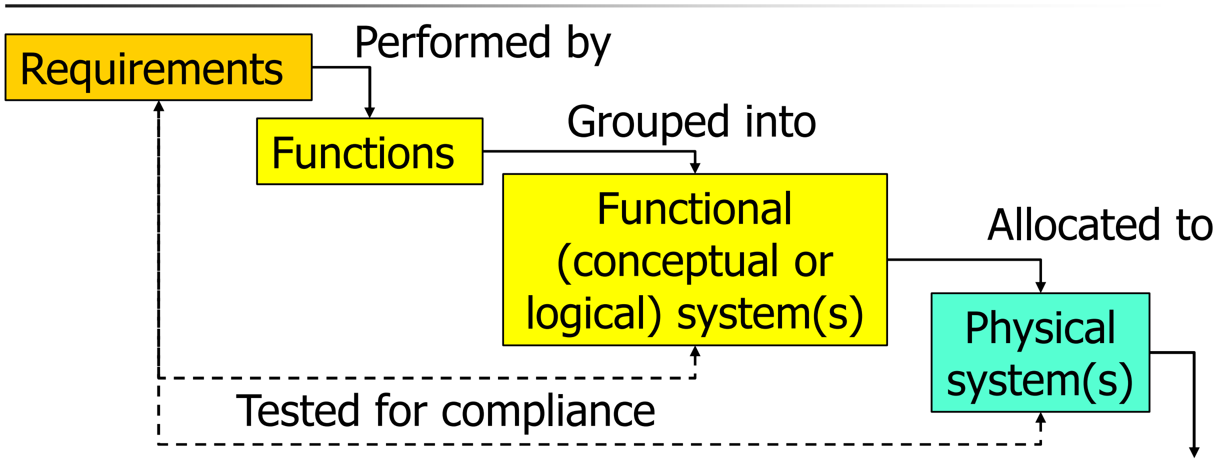
Cmd.
Automatic
C/S or
Operator



Emergent properties (reminder)

- **Known**
  - **Desired** – being the purpose of the system and can only be achieved by the combination of the subsystems or components
  - **Undesired** – known from experience and compensated for in the design
    - E.g. resonance in bridges
  - **Don't care**
- **Unknown**
  - **Undesired** – functionality performed by the system that is undesired, also known as 'side effects'
  - **Serendipitous** – beneficial and desired once discovered, but not part of the original specifications
  - **Don't care**

Functions and physical



Subsystem partitioning

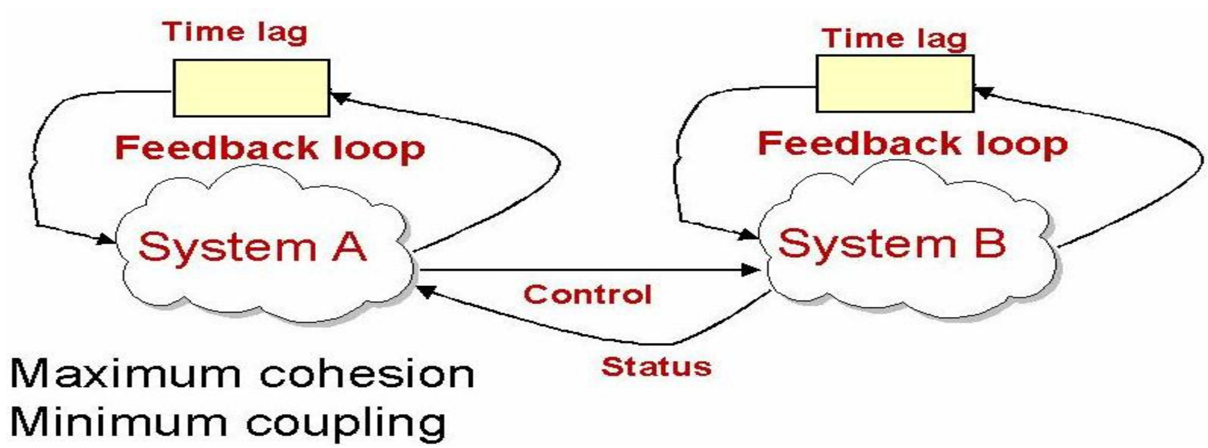
- Design
  - Group functions into physical subsystems (architecture)
  - Try more than one grouping
    - N^2 charts
- Rules for grouping/aggregation/synthesis
  1. Self-regulating or self-sufficient entities (homeostasis)
  2. Maximal internal cohesion of subsystems
  3. Minimal coupling between subsystems
  4. Less than 7 ± 2 subsystems at any level in the hierarchy
  5. Optimize interaction at the interfaces

Functional to physical: N^2 chart representation

A	o		o	o	o		
o	B	o	o	o	o		
	o	C	o			o	o
o	o	o	D			o	o
o	o			E	o	o	
o	o			o	F	o	o
		o	o	o	o	G	o
		o	o		o	o	H

Component

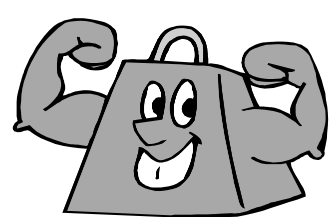
Self regulating systems



Creating Outstanding Systems Engineers 8-27

Design to/for

- Robustness
- Decomposition
  - Functionality and performance
  - Cost
  - Reliability
  - Maintenance
  - Integration and installation
  - Other
- Show how allocations to partitioned bits of system add up to whole
- Traceability matrix



Creating Outstanding Systems Engineers 8-28

Adding up budgets (generic perspective)

- Product dimension (PBS)
  - Functionality $F_s = \Sigma (f_{s1} + f_{s2} + ... + f_{sn}) + f_{sc}$
  - Reliability $R_s = \Sigma (r_{s1}$ and r_{s2} and ... and $r_{sn}) + r_{sc}$
  - Cost $C_s = \Sigma (c_{s1} + c_{s2} + ... + c_{sn}) + c_{sc}$
  - Weight, etc.
- Process dimension (WP/WBS)
  - Work $WP_s = \Sigma (wp_{s1} + wp_{s2} + ... + wp_{sn}) + wp_{sc}$
  - Time $TP_s = \Sigma (tp_{s1} + tp_{s2} + ... + tp_{sn}) + tp_{sc}$
  - Cost $CP_s = \Sigma (cp_{s1} + cp_{s2} + ... + cp_{sn}) + cp_{sc}$
  - Etc.

Example: adjusting mass budget

- System mass specification 10 Kg
- System has 10 subsystems, each specified at 1 Kg
- One day one subsystem reports mass needs to be increased to 2 Kg
- What to do (3 choices)?
  1. Do nothing
     1. if metasystem allows a specification change to 11 Kg
  2. Enquire if other subsystems are using all of their mass budgets
     1. Reduce specifications on other subsystems, increase this one to 2 Kg so system specification remains at 10 Kg
     2. Verify system balance, thermal and other pertinent properties are not affected
  3. Force compliance to 1 Kg
     1. Pay additional costs of forcing compliance

System tradeoffs

- Different designs have different subsystem allocations of
  - Functions, costs, reliability, etc.
- Trade-offs between designs also have to ensure that the combination of all these factors meet the system requirements/specifications
- See examples in Reading 0802

Tools for the design, construction and unit testing states

- Configuration control
- Problem tracking
- Jigs
- Test equipment
- Same set as for the System Requirements State
- N^2 charts
- CRIP Charts
- ETLC Charts

Exercise 8-11

1. Create a preliminary functional architecture for the HEADS
   - Functional perspective, so use functional language (… ing)
2. Prepare <5 minute presentation
   1. This slide and version number of session
   2. Problem formulated per COPS problem formulation template
   3. Summary of functional architecture and justification
   4. Lessons learned
   5. Assumptions
   6. Compliance matrix
3. Save as a PowerPoint file in format Exercise8-11-abcd.pptx
4. Email/post in the Asynchronous group as instructed

Creating Outstanding Systems Engineers 8-33

Exercise 8-12

1. Create a preliminary physical architecture for the HEADS based on the functional architecture you created in Exercise 8-11
2. Prepare <5 minute presentation
   1. Problem formulated per COPS problem formulation template
   2. Summary of physical architecture and justification
   3. Lessons learned
   4. This slide and version number of session
   5. Assumptions
   6. Compliance matrix
3. Save as a PowerPoint file in format Exercise8-12-abcd.pptx
4. Email/Post in the Asynchronous group as instructed

Creating Outstanding Systems Engineers 8-34

Knowledge reading exercise 8-13

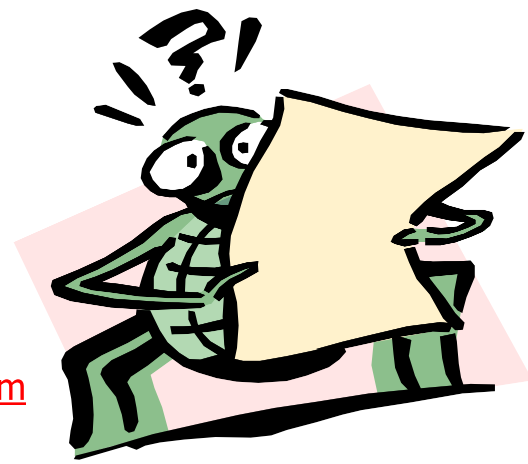
1. Prepare a brief on two main points in reading 0802 (< 5min)
2. Presentation to contain
    1. Formulated problem per COPS problem formulation template
    2. A summary of the content of the reading (<1 minute)
    3. The compliance matrix
    4. This slide and version number of session
    5. The main points (<1 minute)
    6. Brief on one main point (<1 minute per point)
    7. Reflections and comments on reading (<2 minute)
    8. Comparisons of content with other readings and external knowledge
    9. Why you think the reading was assigned to the module
    10. Lessons learned from module and source of learning e.g. readings, exercise, experience, etc. (<2 minutes)
3. Save as a PowerPoint file as Exercise8-13-abcd.pptx
4. Post in the Asynchronous group

Knowledge reading exercise 8-14

1. Prepare a brief on two main points in reading 0804 (< 5min)
2. Presentation to contain
    1. Formulated problem per COPS problem formulation template
    2. A summary of the content of the reading (<1 minute)
    3. The compliance matrix
    4. This slide and version number of session
    5. The main points (<1 minute)
    6. Brief on one main point (<1 minute per point)
    7. Reflections and comments on reading (<2 minute)
    8. Comparisons of content with other readings and external knowledge
    9. Why you think the reading was assigned to the module
    10. Lessons learned from module and source of learning e.g. readings, exercise, experience, etc. (<2 minutes)
3. Save as a PowerPoint file as Exercise8-14-abcd.pptx
4. Post in the Asynchronous group

Any questions ?

1. Best
2. Worst
3. Missing



Email: beyondsystemsthinking@yahoo.com
Subject: <class title> BWM Session #