Massachusetts Institute of Technology

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Fundamentals of Systems Engineering

Prof. Olivier L. de Weck

Session 4

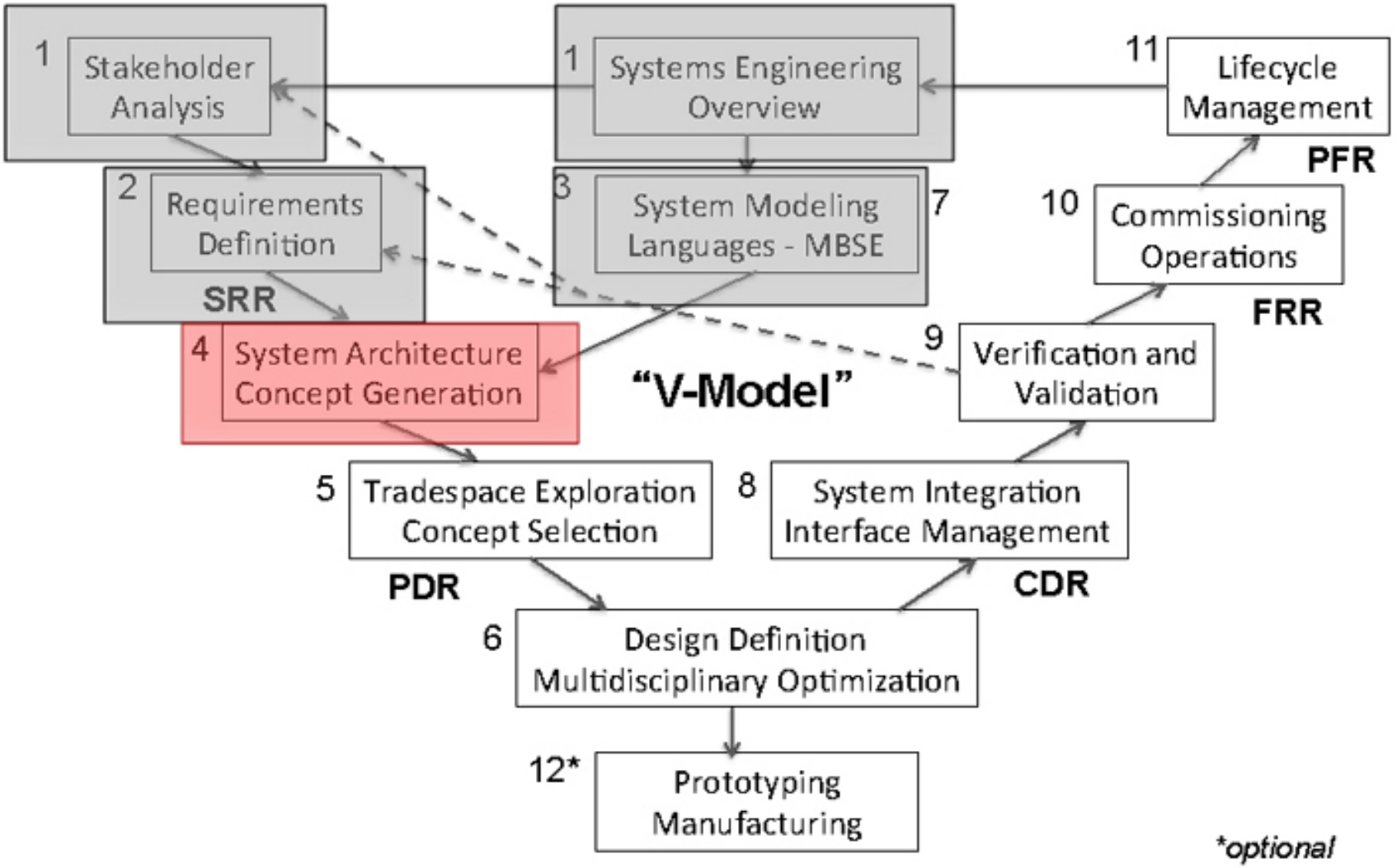
System Architecture
Concept Generation

Reminder: A2 is due today !

Assignment	*Topic*	*Weight*
A1 (group)	Team Formation, Definitions, Stakeholders, Concept of Operations (CONOPS)	12.5%
A2 (group)	Requirements Definition and Analysis Margins Allocation	12.5%
A3 (group)	System Architecture, Concept Generation	12.5%
A4 (group)	Tradespace Exploration, Concept Selection	12.5%
A5 (group)	Preliminary Design Review (PDR) Package and Presentation	20%
Quiz (individual)	Written online quiz	10%
Oral Exam (individual)	20' Oral Exam with Instructor 2-page reflective memorandum	10%

The "V-Model" of Systems Engineering

16.842/ENG-421 Fundamentals of Systems Engineering



Numbers indicate the session # in this class

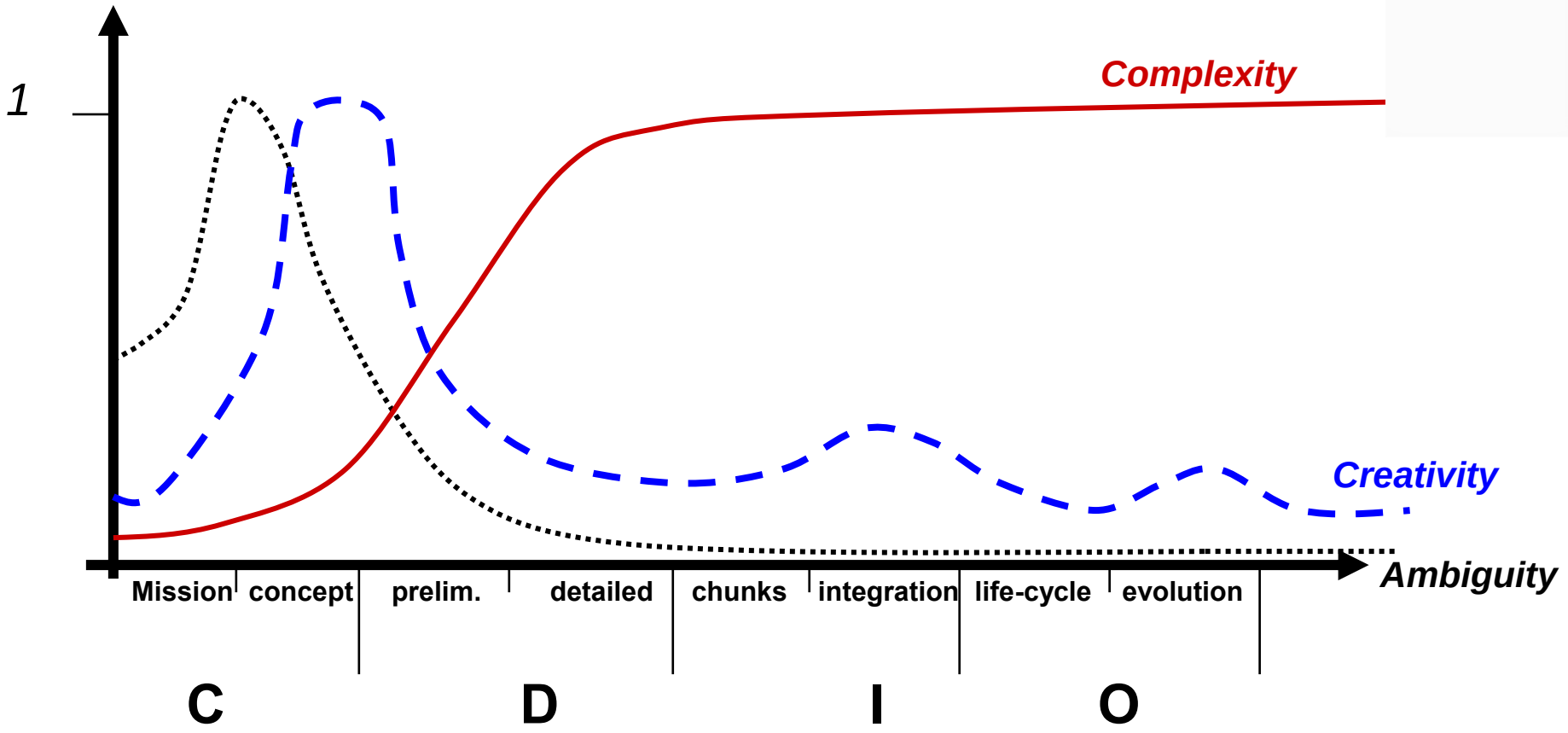
Overview

- System Architecture
  - Definition, Themes, Exercise
- NASA Approach
  - Logical Decomposition
- Methods/Tools for Concept Generation
  - Creativity Techniques
  - Brainstorming
  - Morphological Matrix / Architecture Enumeration

Questions to be asked

- Why are we doing the project? → Stakeholder Analysis
- What must we achieve → Requirements Definition
- How could we do it? → today !
  - Oftentimes there are many different ways

Themes: Ambiguity-Creativity-Complexity

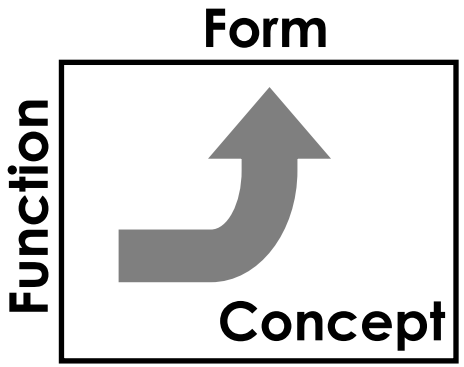


Early on ambiguity is high → reduce ambiguity → requirements
Next concept are needed → focus creativity → concepts
Then complexity starts increasing → manage complexity → designs

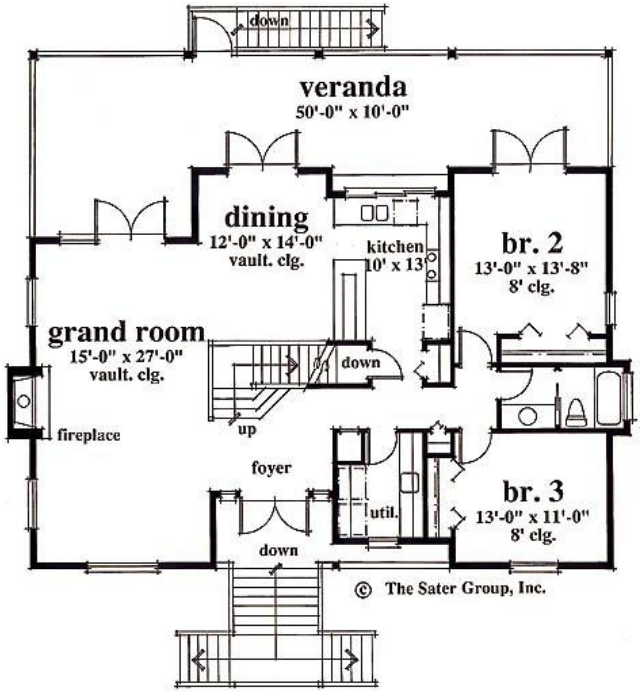
A Definition

- Architecture
  - The embodiment of **concept**, and the allocation of physical/informational **function** (process) to elements of **form** (objects) and definition of structural interfaces among the objects

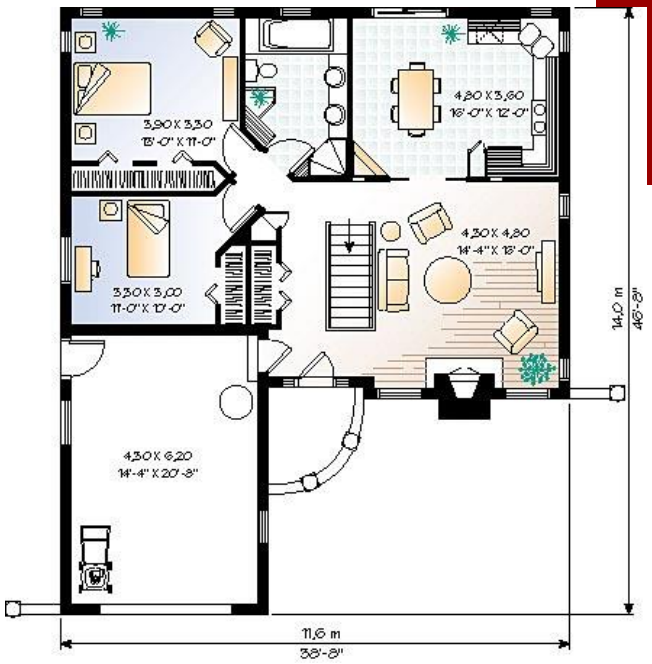
- Consists of:
  - Function
  - Related by Concept
  - To Form



Architecture – Civil



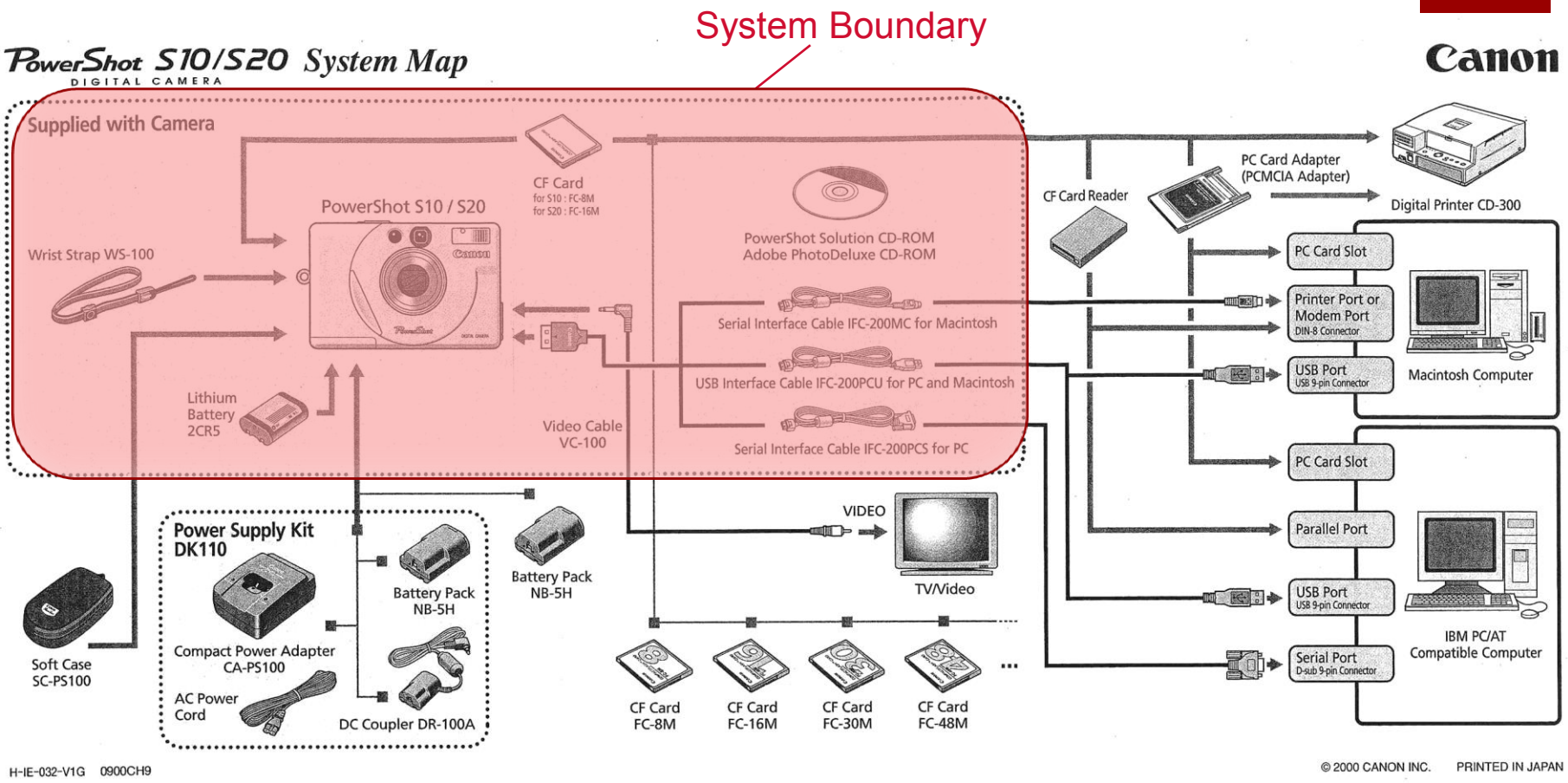
Beach



Contemporary

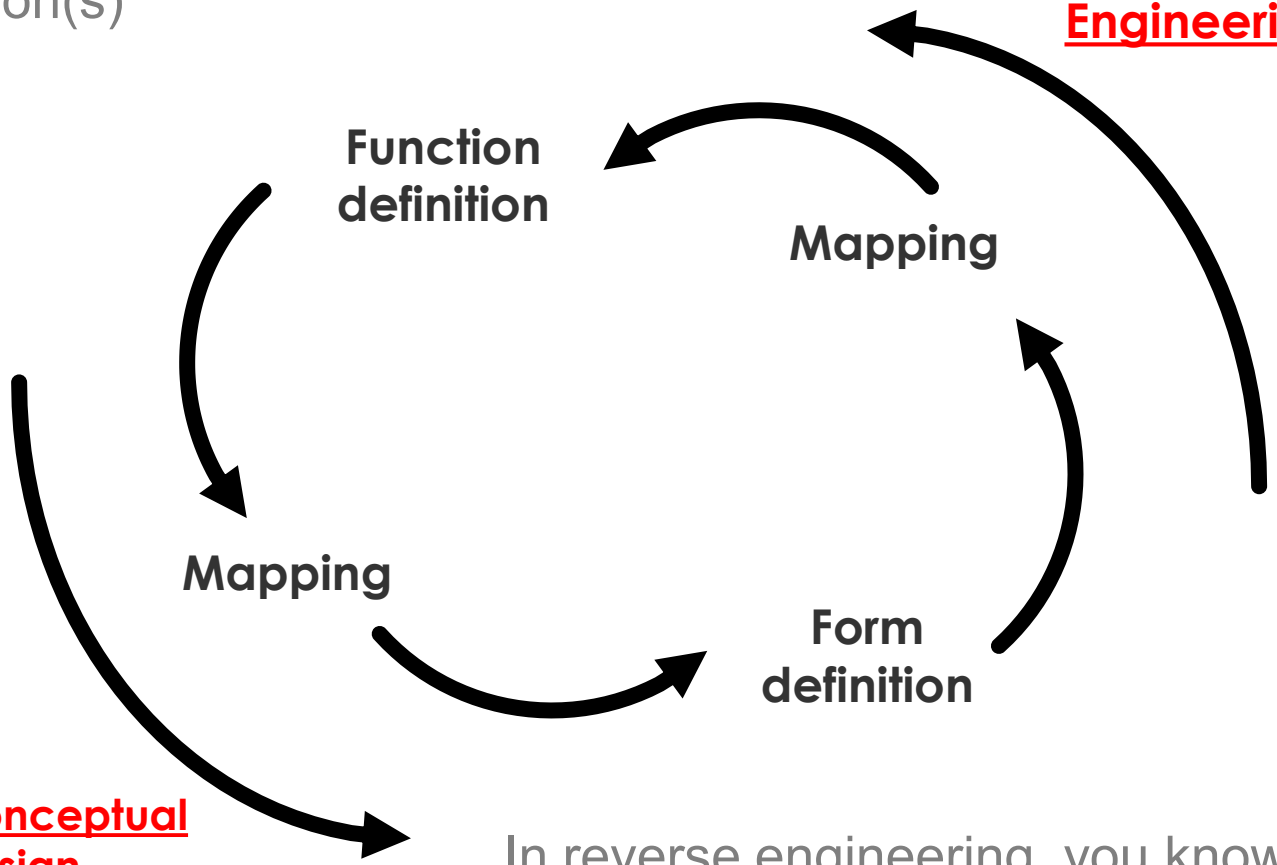
Source - www.coolhouseplans.com

Architecture - Informational



Architecting Sequence

In design, you know the functions you want and try to create the form to deliver the function(s)



In reverse engineering, you know the form, and are trying to infer the function (and presumably eventually the goals)

Sequences in Design and Reverse Engineering

- In Architecting and Design
  - Define externally delivered function, create concept, break down functions ("functional decomposition"), define subsystems

- In Reverse Engineering
  - Define subsystems, infer function, infer concept, infer externally delivered function (or lack thereof)
  - Last two steps are difficult due to the emergence of function

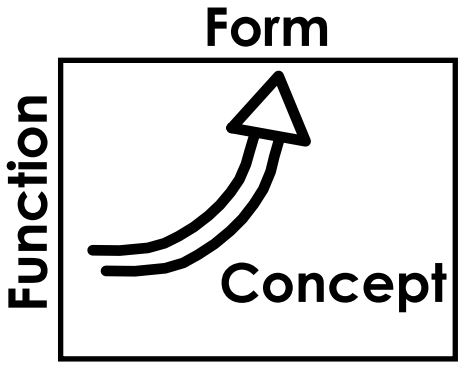
Famous case of reverse engineering: Acutan Zero
http://en.wikipedia.org/wiki/Akutan_Zero

Mitsubishi Zero Fighter (WWII)

Concept

- Is created by the architect
- Must allow for execution of all functions
- Establishes the solution vocabulary
- Implicitly represents a technology

Concept – Informal Definition

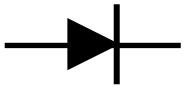


- A product or system vision, idea, notion or mental image which:
  - Maps Form to Function
  - Embodies "Working Principles"
- Is in the solution-specific vocabulary - it is the solution
- Is an abstraction of form

Is not a product/system attribute, but a mapping

Expressing Concept

- New Concepts are expressed by a few words or a short phrase (e.g. **refrigerator** = insulated box containing mechanically chilled air)

- Established concepts can often be expressed by a word or two (sometimes the common name of the form associated with the concept) or an icon 

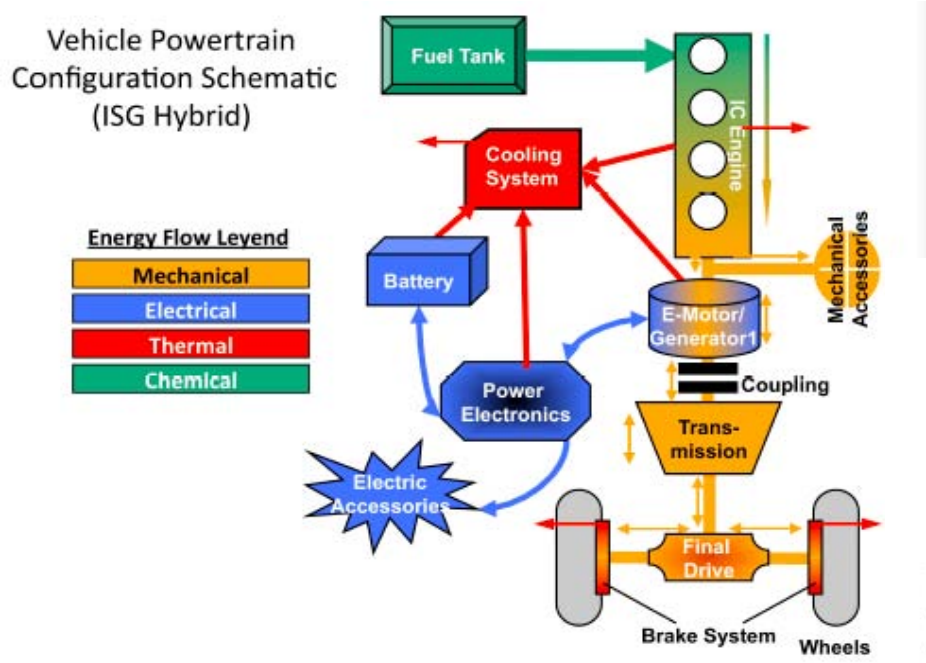
- Once concept is specified, the nature and list of parts is more or less established
- Concepts, like form, are expressed as nouns, but concepts tend to be more abstract, while form is actually implemented

- **What is the key concept of some of the products we see in this room?**

Concept - Formal Definition

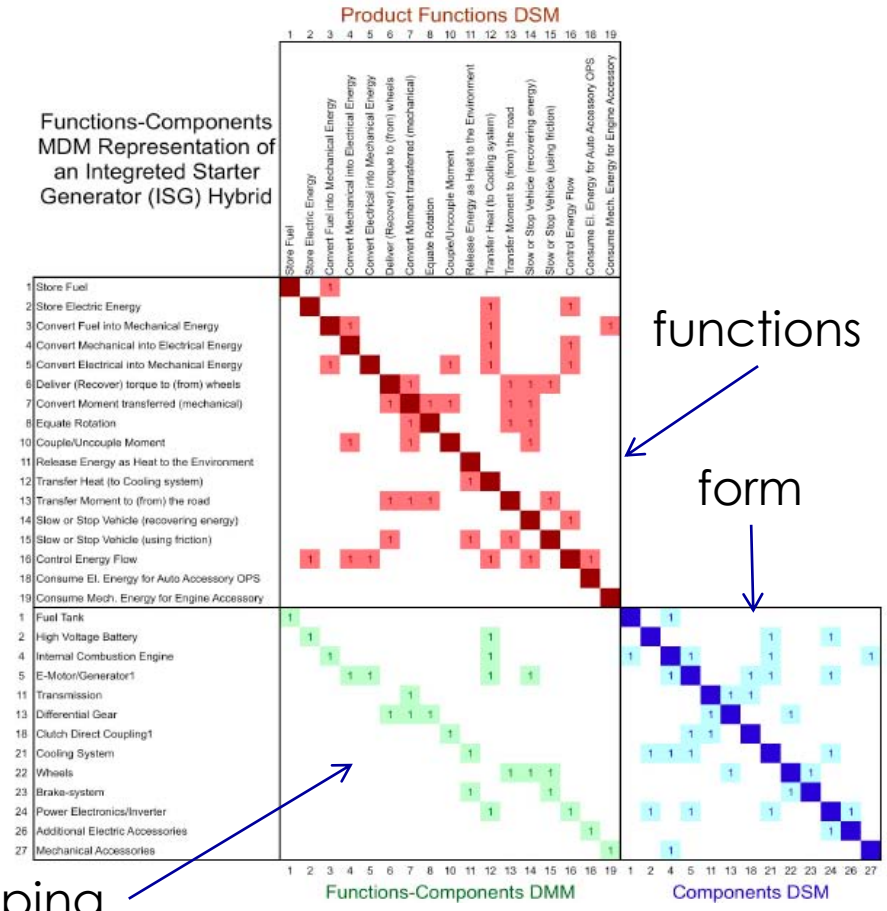
- The specialization of function and mapping to its physical embodiment of form
- The specification of the list of the design variables, which when specified will define the design
- Products based on the same concept are "continuously connected"
  - **Typically products in a family are based on same concept**

Concept: Gasoline-Electric Hybrid Vehicle



Schematic showing Form

Multi-domain-matrix (MDM) showing form and function



functions

form

form-function-mapping

Credit: Dr. Carlos Gorbea, TU Munich

Partner Exercise (2 min)

- Pick one of these three objects and describe the concept, using the language of system architecture

A

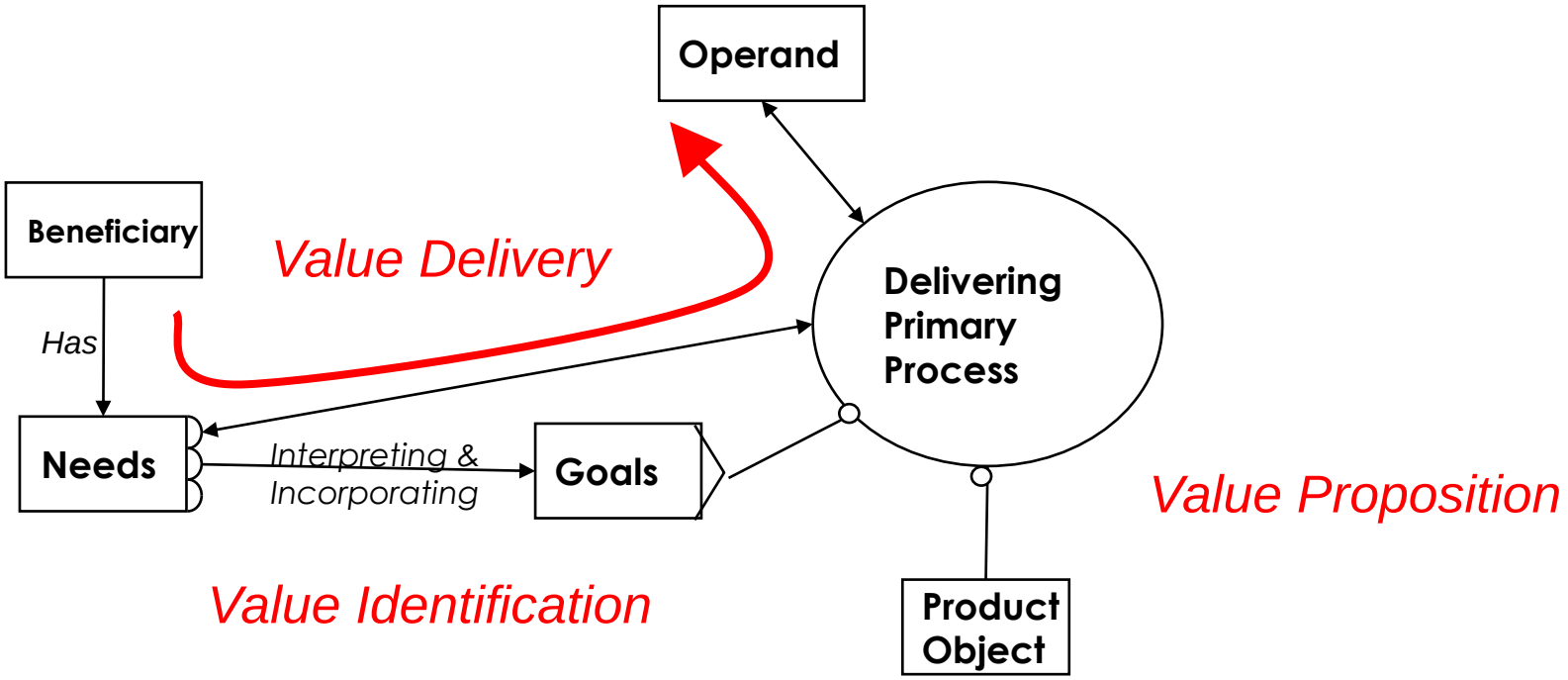
B

C

Refrigerator Case Study

Value - A Formal Definition

Value is delivered when the primary external process(es) acts on the operand in such a way that the needs of the beneficiary are satisfied.

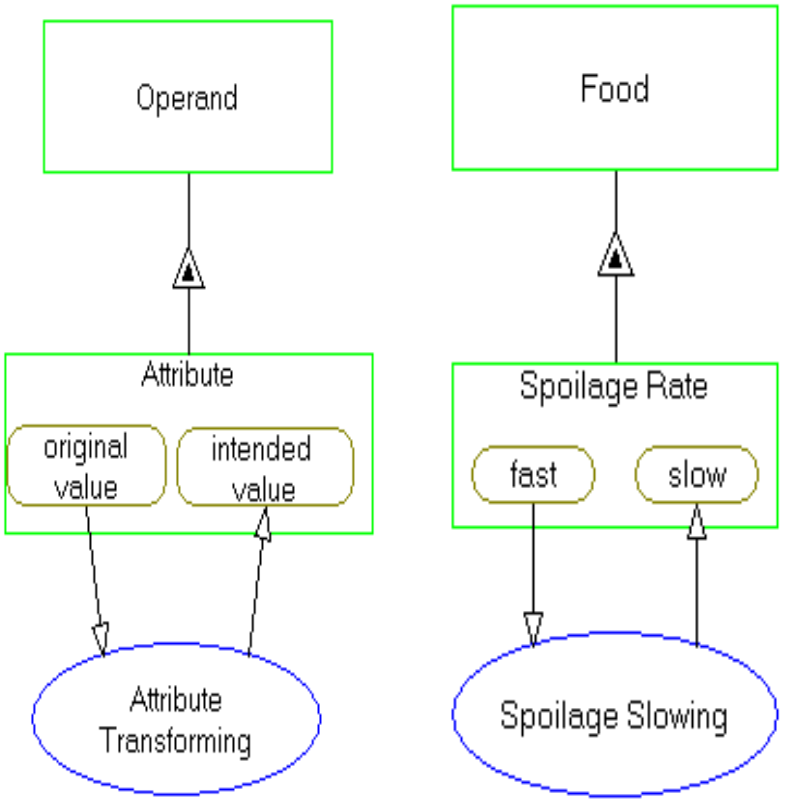


Using Object-Process-Methodology (**OPM**) Nomenclature [Dori, 2002]

Reduce Ambiguity: Goal Identification

- Start by examining the operand associated with value
- Next identify the attribute of the operand whose change is associated with value
- Next define the transformation of the attribute associated with value, in solution neutral form

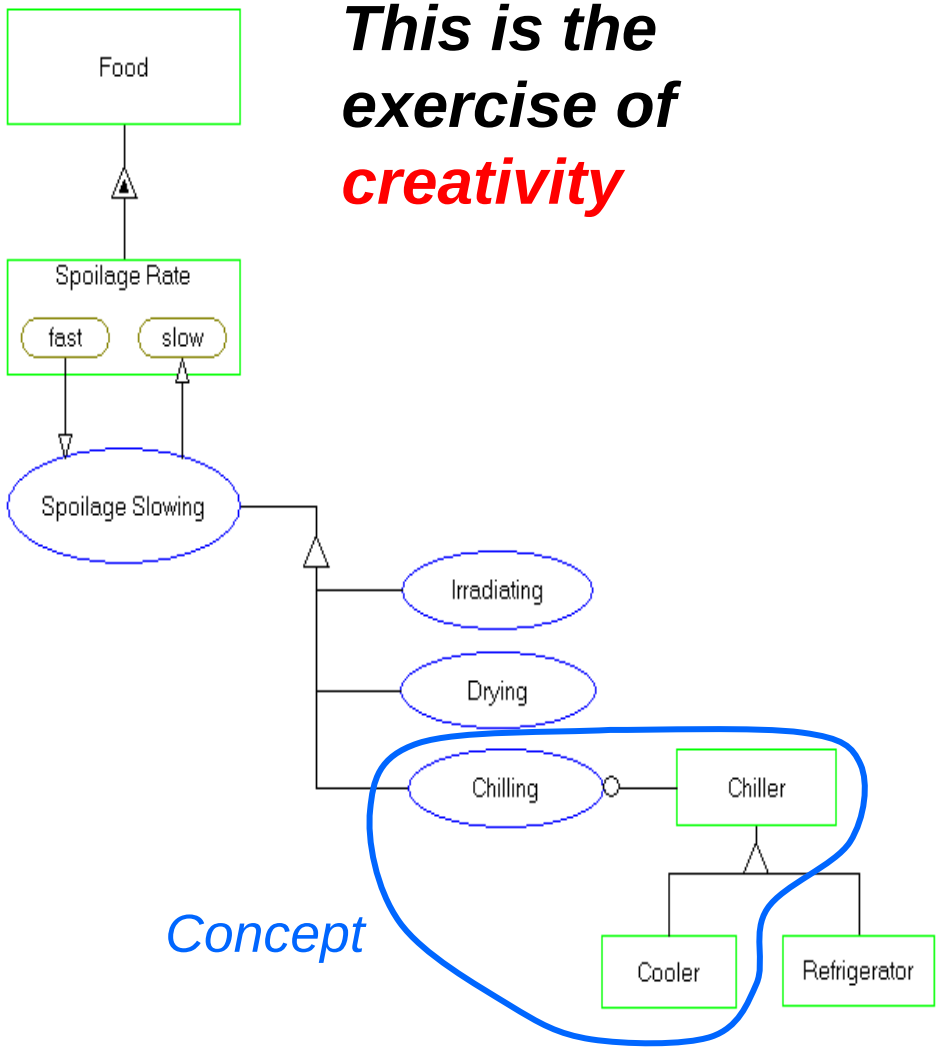
Note: For "Production Systems" the value could be found not in an operand whose attributes are affected but in a resultee that is created



This will reduce ambiguity and lead you to a value focused, solution neutral statement of intent on process

Focus Creativity : Concept

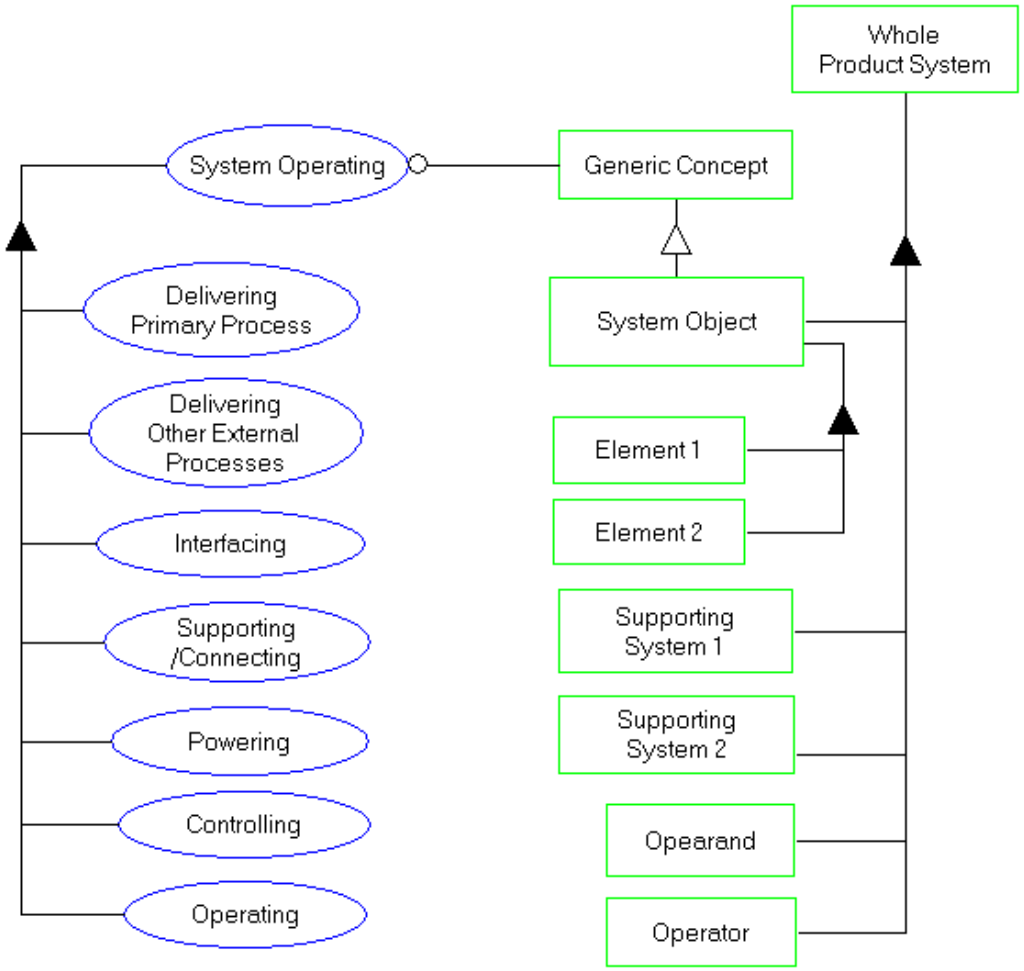
- Concept: a system vision, which embodies working principles, a mapping from function to form
- Choose from among the system operating processes that specialize to the desired solution neutral, value related process
- Specialize the related generic concept to the product form



This is the exercise of creativity

Managing Complexity: Decomposition of Function and Form

- Identify form of the whole product system
- Zoom the processes of function
- Decompose the form of the product object
- Establish the object process links



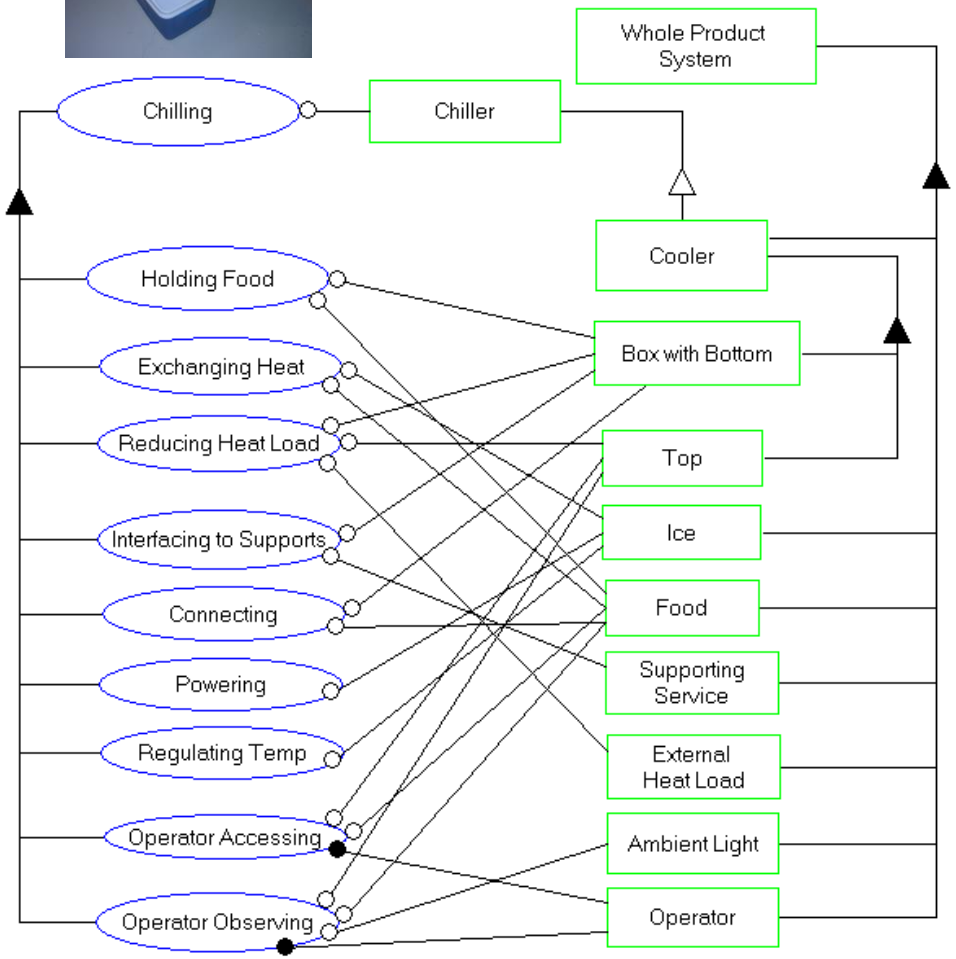
Form and Function -Cooler

The whole product includes the ice, food, supporting surface, heat load, light and operator

Chilling zooms to the stated processes (using process precedence framework)

Cooler decomposes to box and top

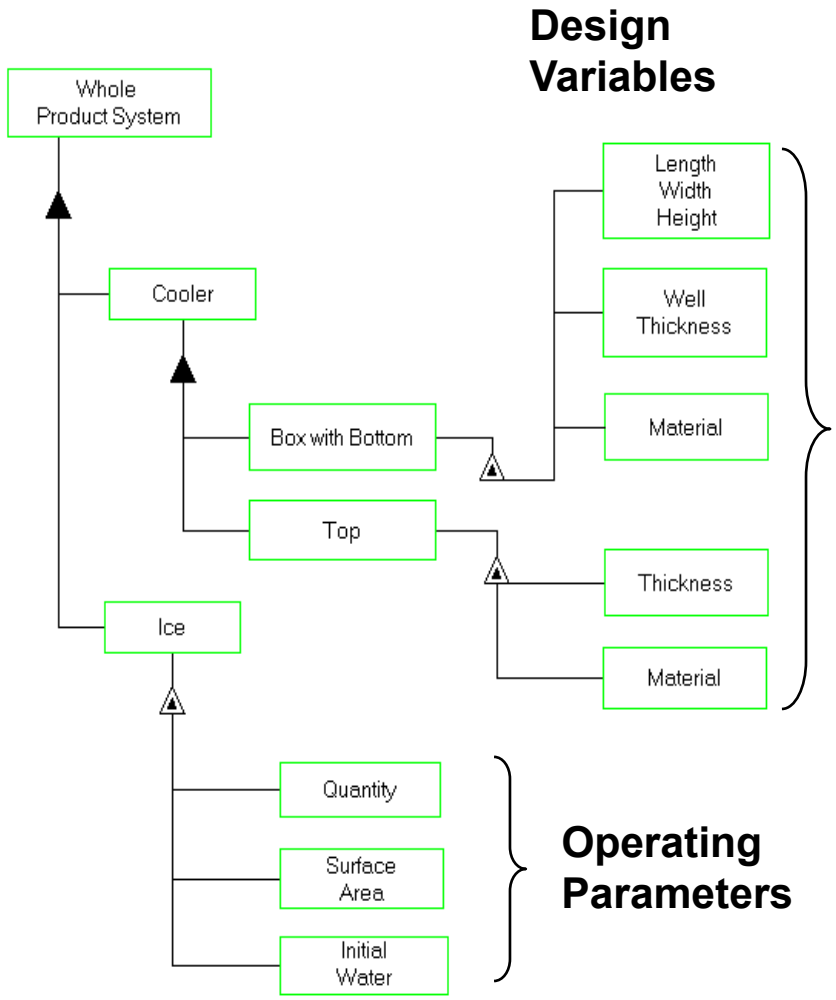
Map objects to processes to determine object-process architecture



Establishing the complexity of the object-process architecture

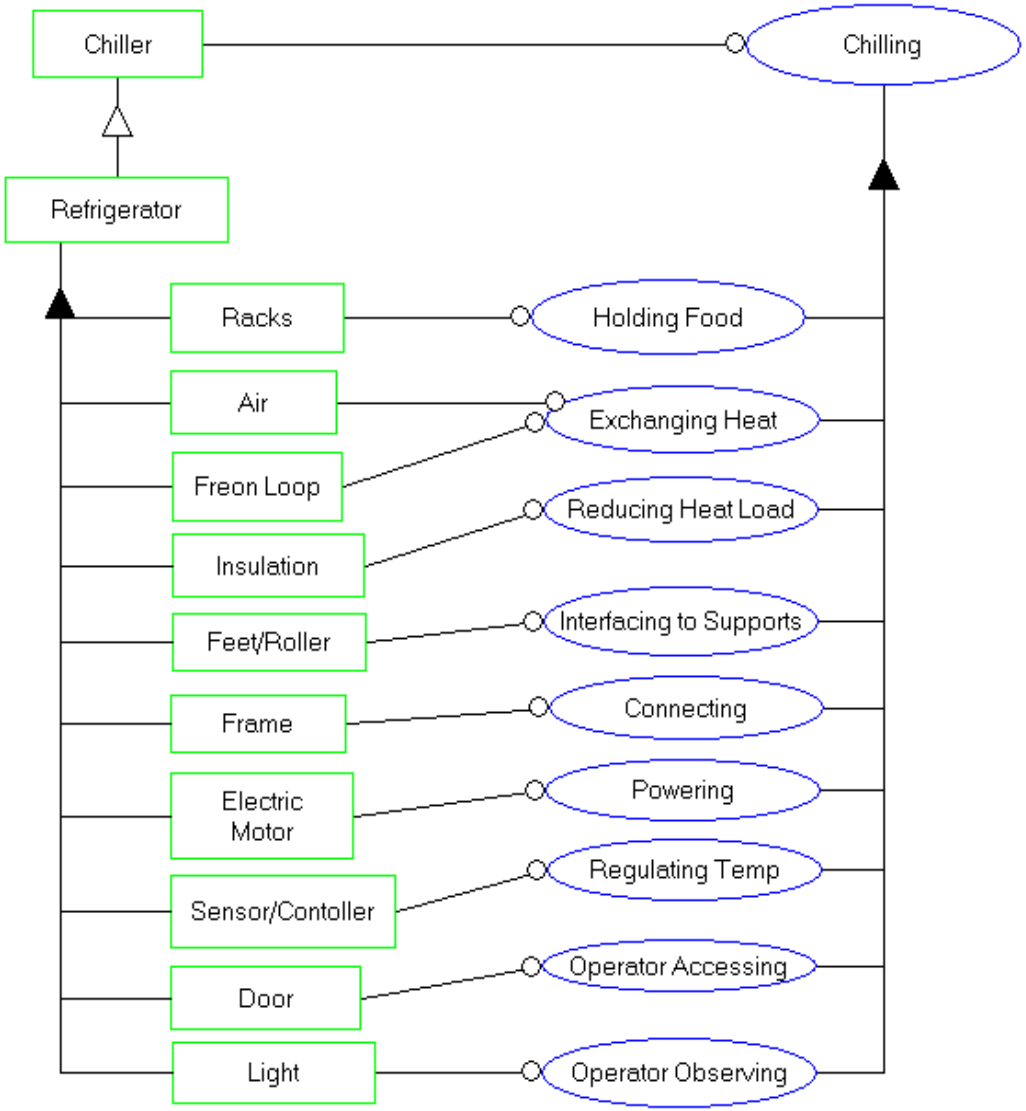
Design vs. Architecture

- Architecture selects the concept, decomposition and mapping of form to function
- Architecture establishes the vector of design variables and operating parameters
- Design selects of the values of the vector of parameters
- This is what optimization is good for
- Some work in "architecture" is just an exhaustive search over the design of one architecture

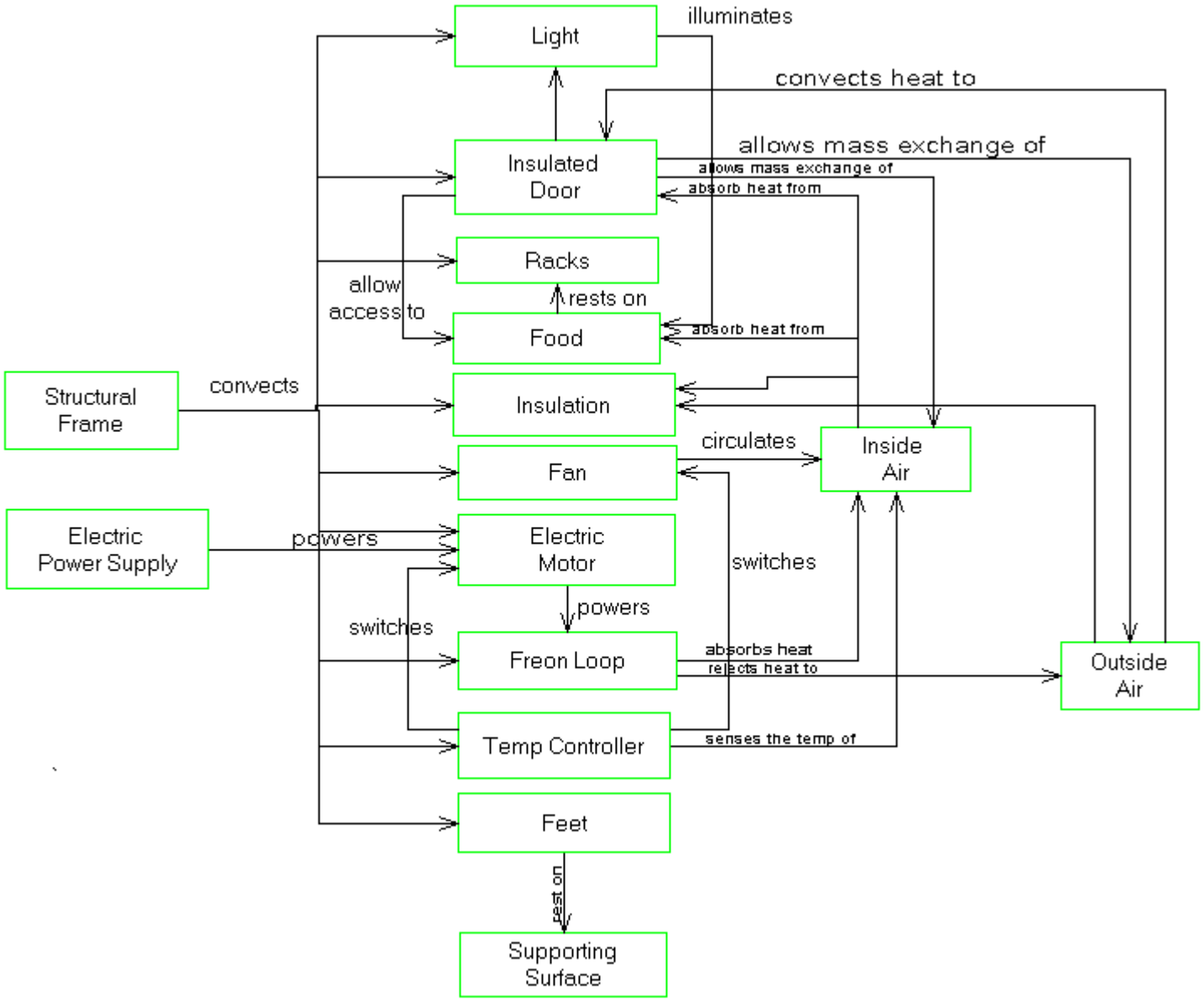


Form and Function - Refrigerator

- More one to one correspondence of objects and processes
- Note the whole product elements suppressed:
  - Food
  - Support structure
  - Heat load
  - Operator
- Simple Object-Process Architecture



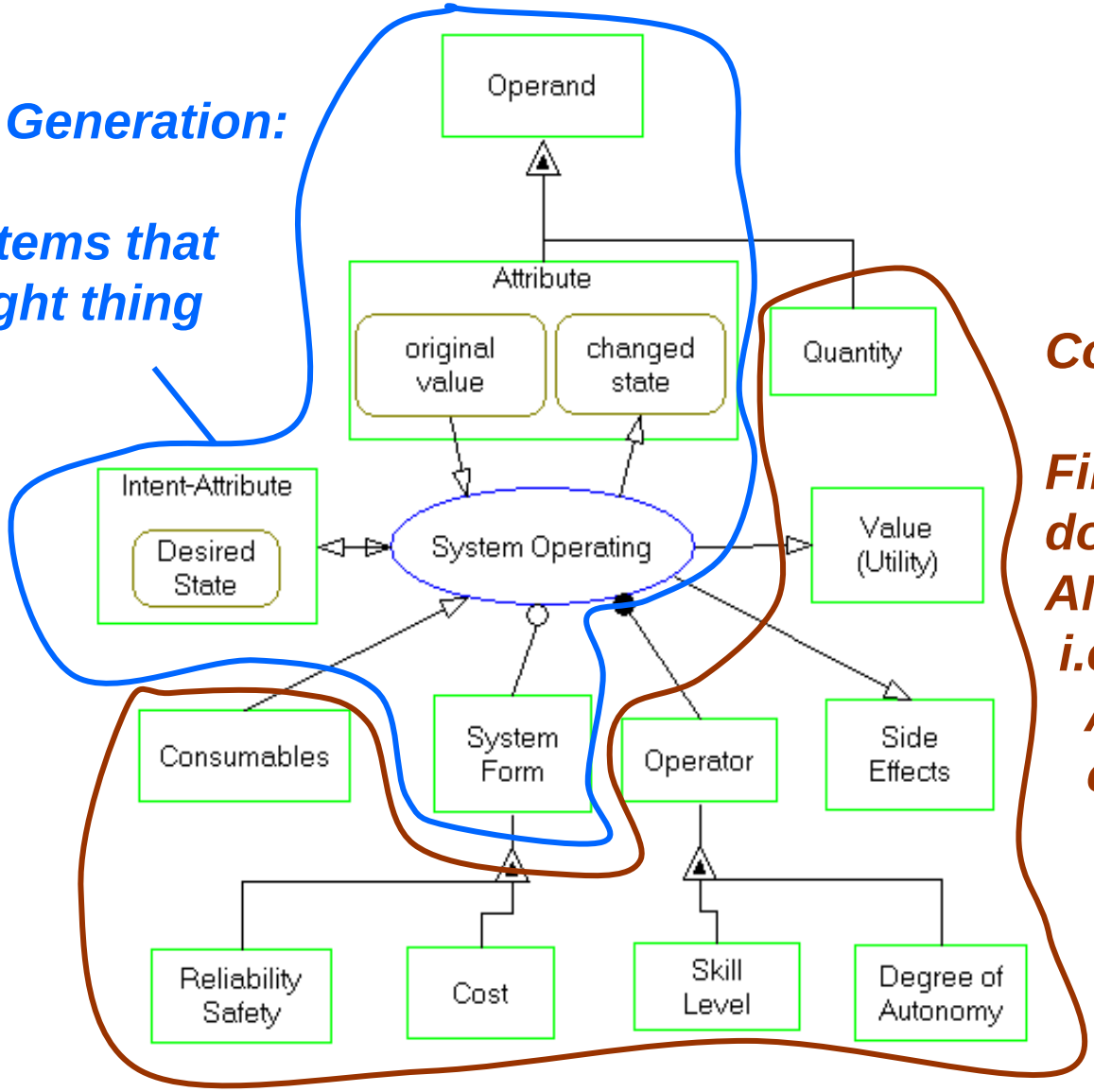
Structure of Form - Refrigerator



Concept Generation versus Selection

Concept Generation:

Find systems that do the right thing



Concept Selection:

Find systems that do the right thing AND do it well, i.e. deliver value, AND comply with current and future regulations and standards

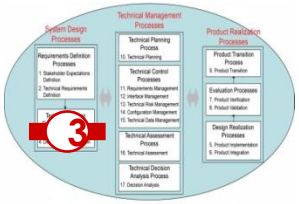
Systems Architecture - Summary

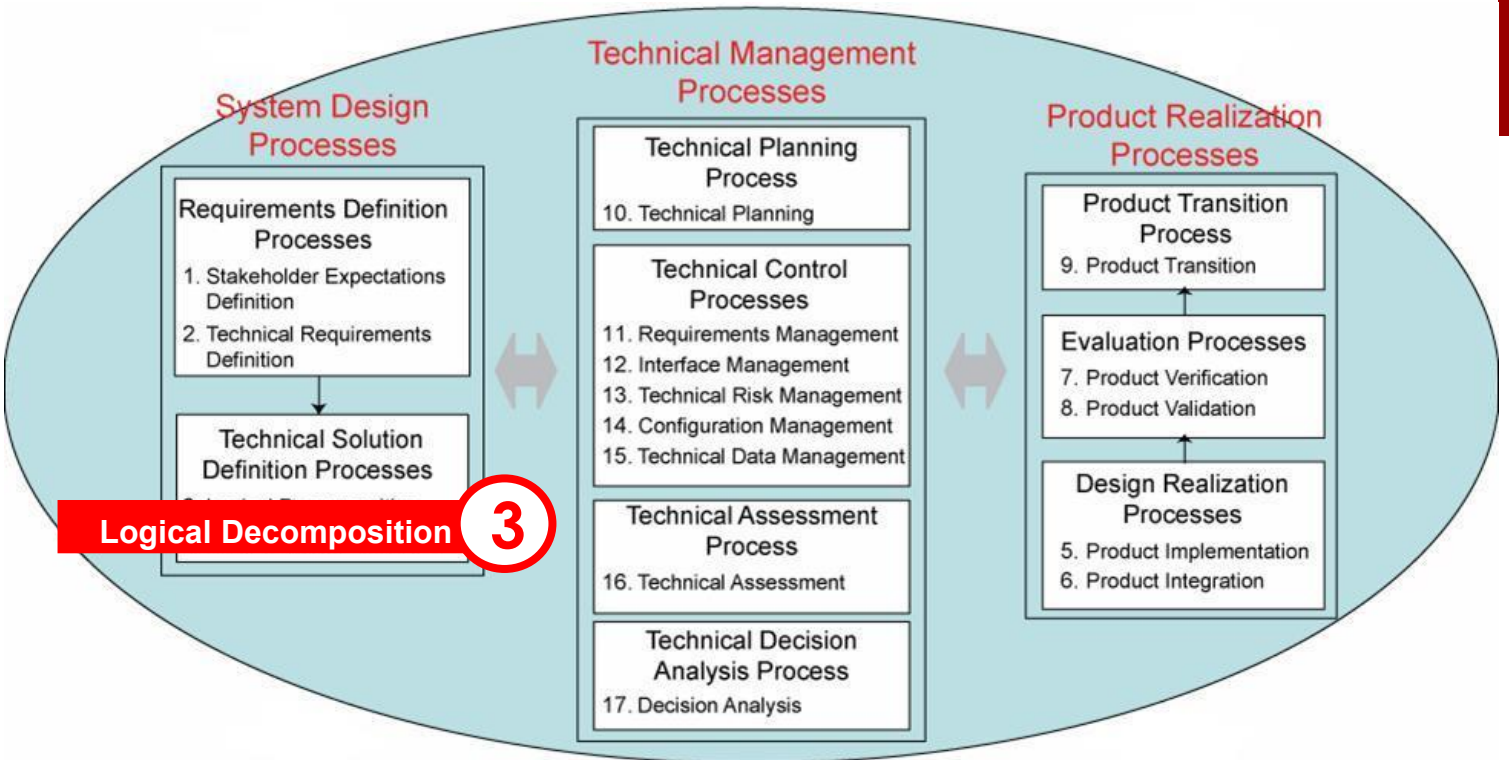
- Architecture requires consideration of form and function, related through concept
- Starting with the operand, its transformation identifies concepts which deliver value and meet requirements
- Concepts elaborate into architectures which have form-function and structural complexity
- “Goodness” of an architecture is a multiobjective value-delivering quality that includes performance, resource utilization, cost, operability and capacity among others

Overview

- System Architecture
  - Definition, Themes, Exercise
- NASA Approach
  - Logical Decomposition
- Methods/Tools for Concept Generation
  - Creativity Techniques
  - Brainstorming
  - Morphological Matrix

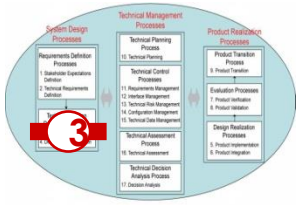
NASA SE Handbook:
Logical Decomposition Process





- **Requirement 17 (Section 3.2.3.1) "The Center Directors or designees shall establish and maintain a process, to include activities, requirements, guidelines, and documentation, for logical decomposition of the validated technical requirements of the applicable WBS."**

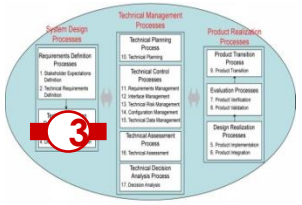
Logical Decomposition Purpose

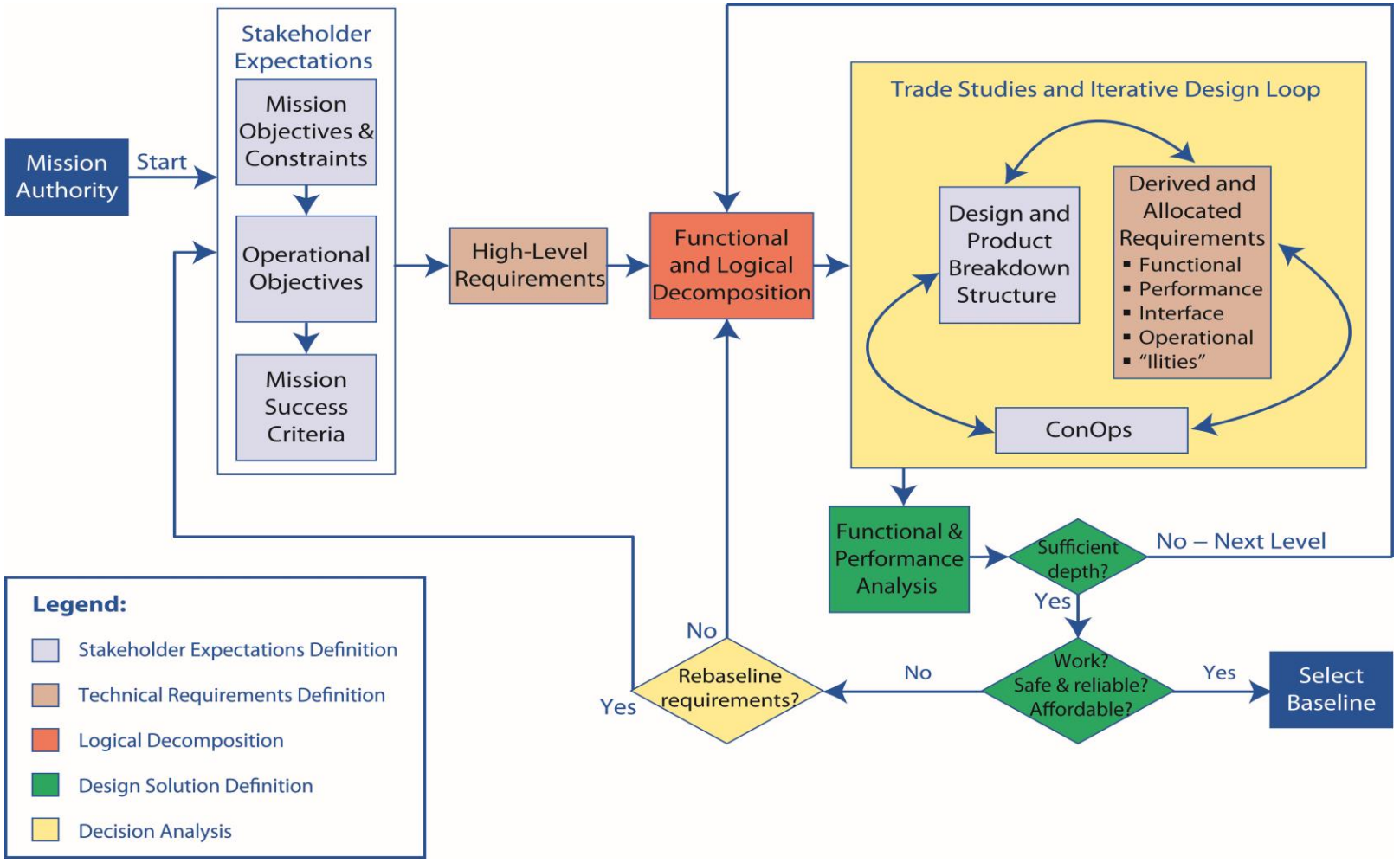


- The Logical Decomposition Process is used to:
  - Improve understanding of the defined technical requirements and the relationships among the requirements (e.g. functional, behavioral, and temporal)
  - Transform the defined set of technical requirements into a set of logical decomposition models and their associated set of derived technical requirements for input into the Design Solution Definition Process

ARCHITECT THE SYSTEM

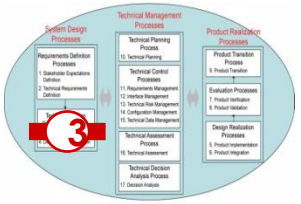
Interrelationships Among the System Design Processes



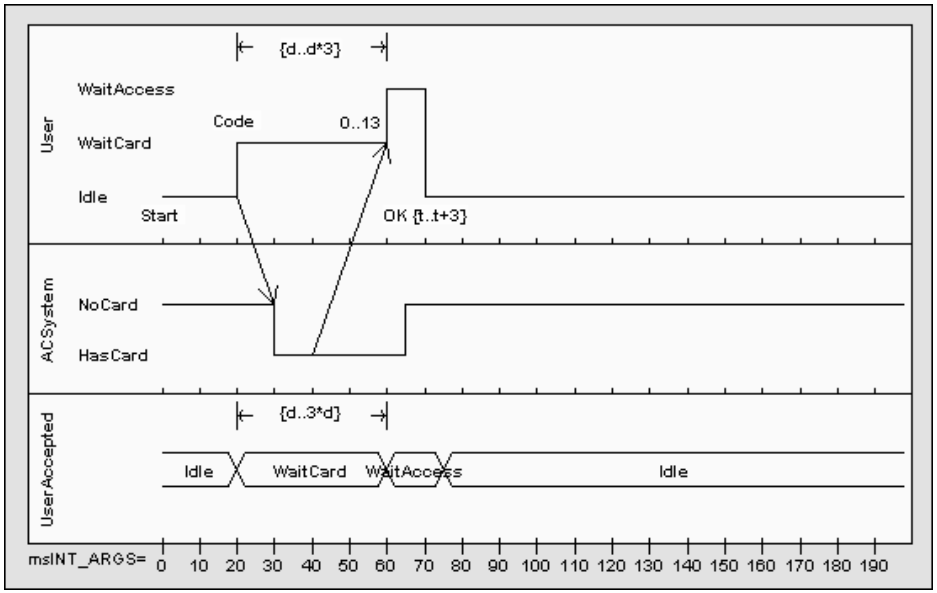


SP-2007-6105, Figure 4.01

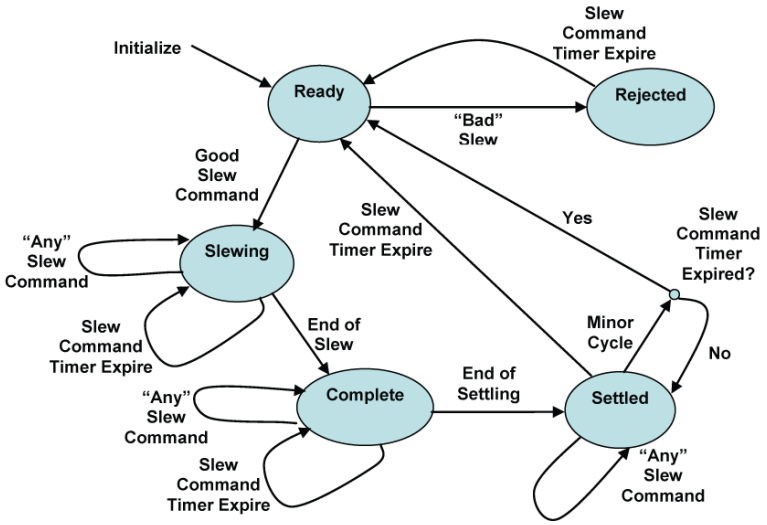
Example of Decomposition Models



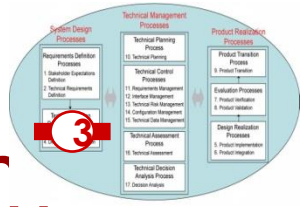
Timing Diagram



State Diagrams



Logical Decomposition Best Practice Process Flow Diagram



Input

From Technical Requirements Definition and Configuration Management Processes

Baselined Technical Requirements

From Technical Requirements Definition and Technical Data Management Processes

Measures of Performance

Activities

Define One or More Logical Decomposition Models

Allocate Technical Requirements to Logical Decomposition Models to Form a Set of Derived Technical Requirements

Resolve Derived Technical Requirement Conflicts

Validate the Resulting Set of Derived Technical Requirements

Establish the Derived Technical Requirements Baseline

Output

To Design Solution and Requirements and Interface Management Processes

Derived Technical Requirements

To Design Solution and Configuration Management Processes

Logical Decomposition Models

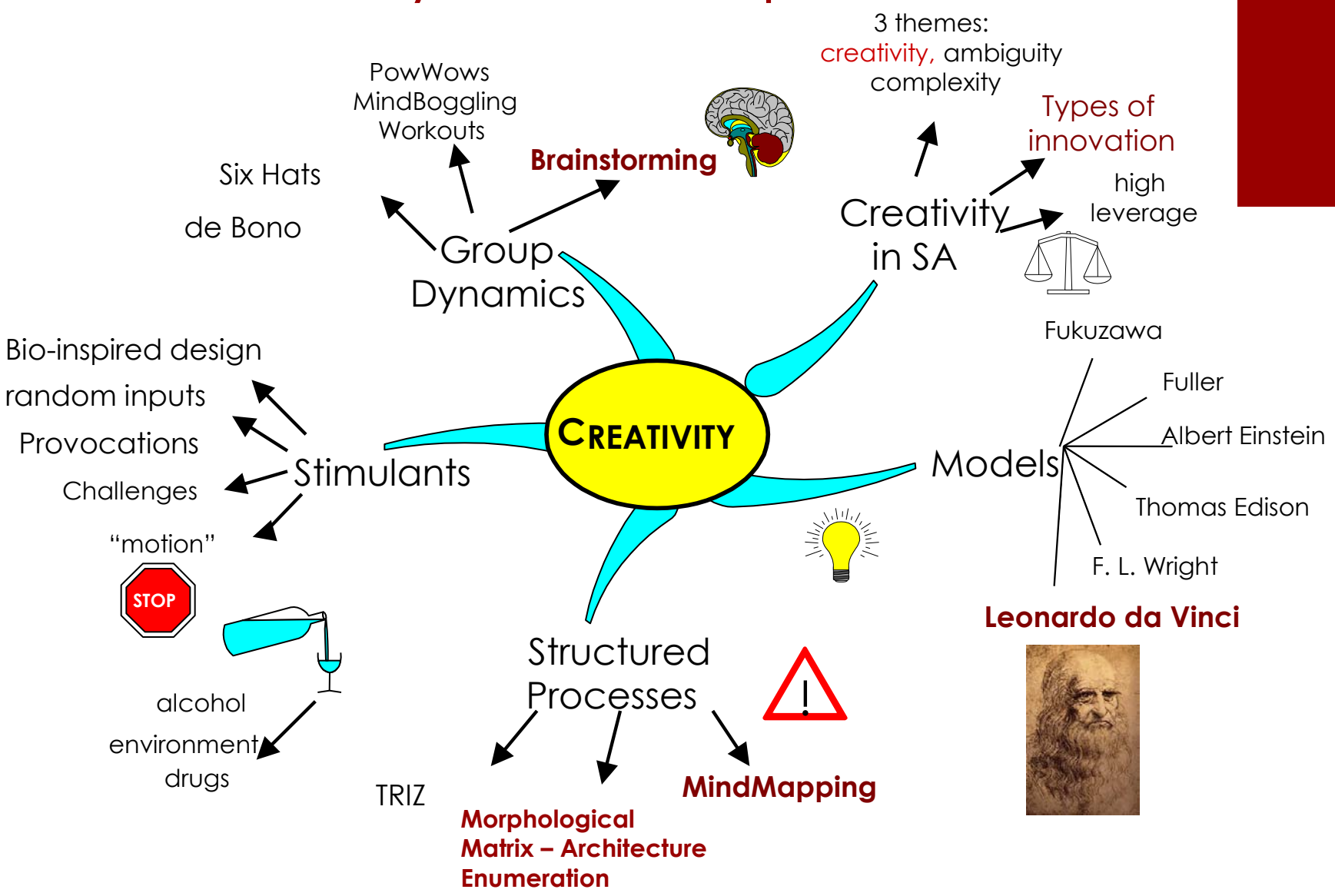
To Technical Data Management Processes

Logical Decomposition Work Products

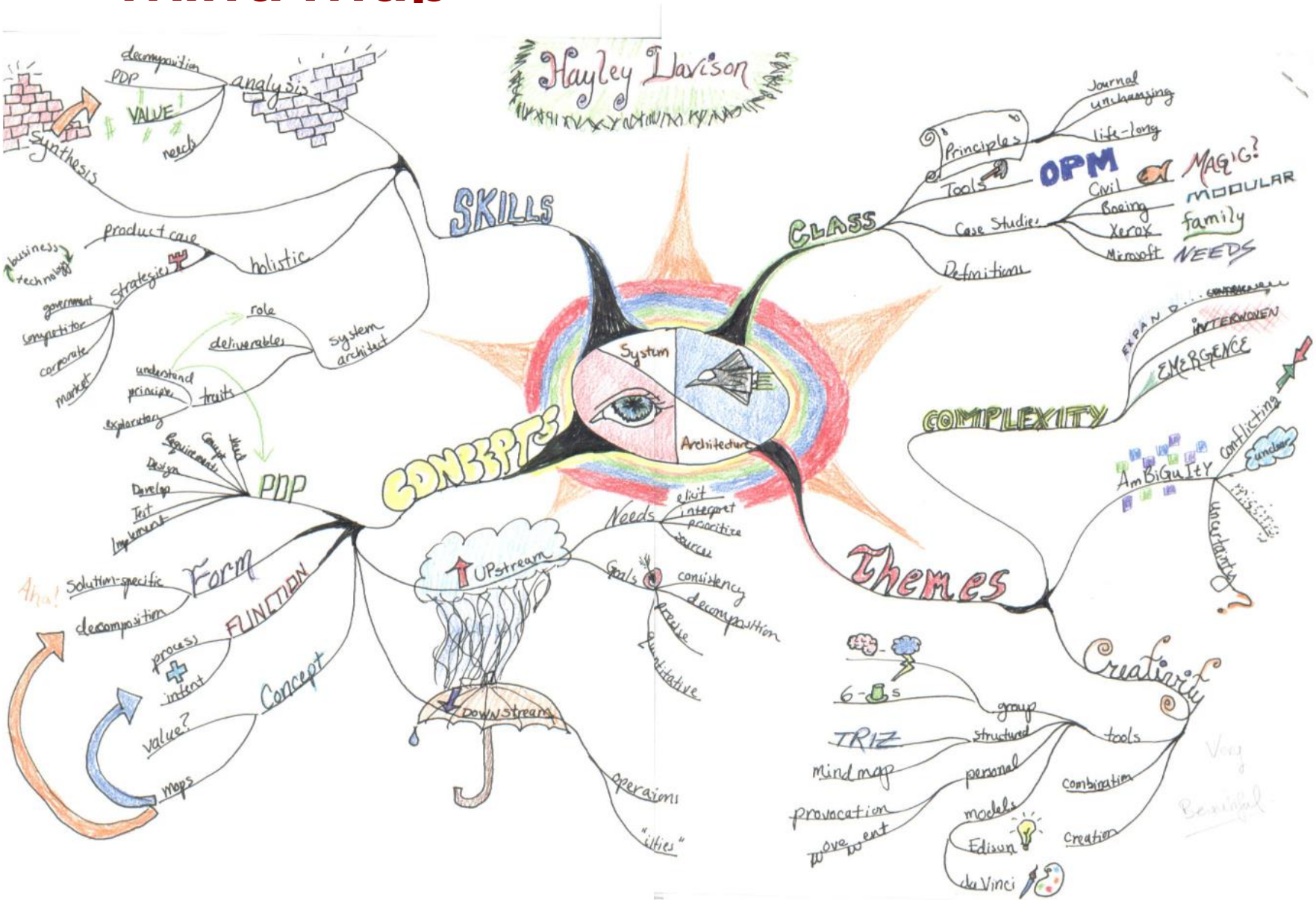
Overview

- System Architecture
  - Definition, Themes, Exercise
- NASA Approach
  - Logical Decomposition
- Methods/Tools for Concept Generation
  - Creativity Techniques
  - Brainstorming
  - Morphological Matrix / Architecture Enumeration

Creativity Mind Map



Mind Map



Courtesy of Hayley Davison Reynolds. Used with permission.

Brainstorming

- Creativity technique designed to find new ideas as a TEAM
- Lower or remove creativity barriers
- Increase idea production via mutual stimulation
- Ideal group size is 5-10 people with diverse backgrounds
- Use of intuition and association
- Based on a solution-neutral question such as:
  "What can be done to …. ?", "How could we improve … ?"

Attributed to **A.F. Osborn**
Osborn, A.F., "Applied Imagination. Principles and procedures of creative thinking", Rev. Ed., New York, 1957

- Why is brainstorming useful ?
- How to organize and host a brainstorming session ?
- Killer sentences
- What to do with the results ?

Organizing a Brainstorming Session

(1) Send out an invitation a few days ahead of time. Announce brainstorming topic. The brainstorming is the ONLY agenda item.

(2) Ideally 5-10 (7+/- 2) participants. Facilitator welcomes participants and briefly describes problem or topic.

(3) Participants take turns expressing thoughts, suggestions, ideas without the constraints of a particular order. It is allowed to extend or combine ideas of others leading to mutual stimulation.

(4) Facilitator takes notes, visible for everyone (e.g. flipchart) without associating names.

(5) Principle of delayed judgement. No one is allowed to criticize or particularly praise other ideas. Avoid killer phrases.

(6) Produce a large amount and diversity of ideas. Session ends after 30-60 minutes or so. Post-Processing happens separately.

Creativity Killer Sentences

This will never work !
We don't even need to talk about this !
There's no time for this !
Why change it … it's working just fine !
Everyone does it this way !
Who the hell is going to pay for all this ?
You can't view it this way !
What is YOUR expertise in this field ?
Have you had any previous experience with this ?
You can't possibly be serious ?
You are completely missing the point !
I have already studied this problem for years !
Do you know where I went to school ?
Don't worry… I know I'm right.
How long have you been with this company ?
I know what it means to….
Now that we agree on this, let's move on …

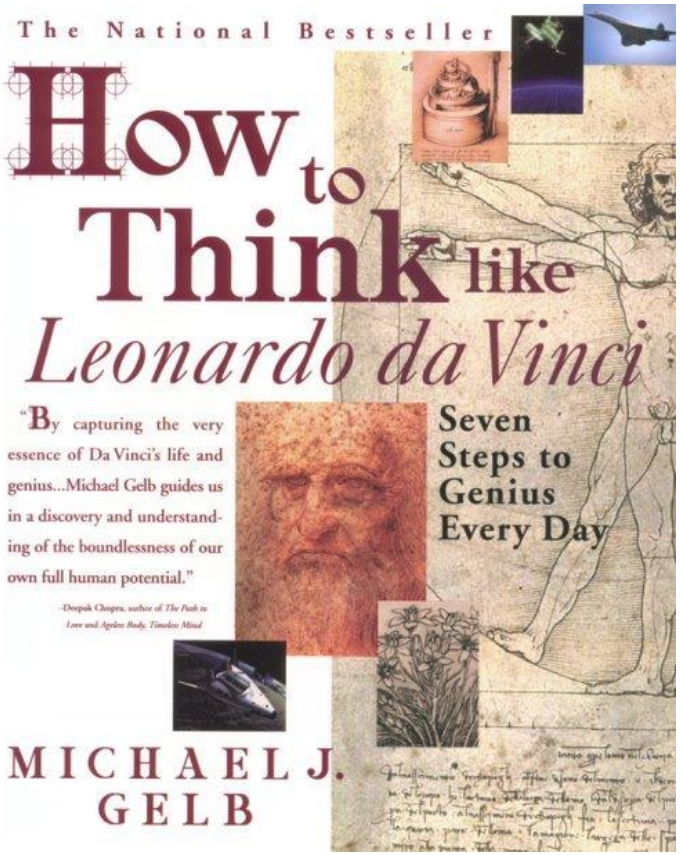
Leonardo da Vinci

"Models" approach to creativity:

Identify exceptionally creative individuals and ask:

"what principles did they follow ?"
" where did they find inspiration ? "

How to Think Like Leonardo Da Vinci: Seven Steps to Genius by Michael J. Gelb



Seven Da Vincian Principles

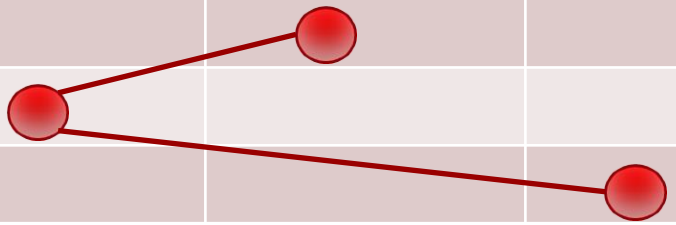
- Curiosita - a lifelong quest for learning
- Dimostrazione - testing knowledge through experience
- Sensazione - continual refinement of the senses
- Sfumato - mastering ambiguity, paradox, and uncertainty
- Arte/Scienza - "whole brain" thinking
- Corporalita the balance of body and mind
- Connessione - the appreciation of patterns, relationships, connections, systems

Morphological Matrix

- The simplest structured way to generate different concepts / architectures
- List the *m* key decisions or factors as rows
- In each row determine the number of possible alternatives n_i
- Enumerate all possible combinations *N*
- Example

$$N = \prod_{i=1}^{m} n_i$$

Alternatives	1	2	3
Factor A			
Factor B			
Factor C			



The concept shown above is: A2-B1-C3
In total the above matrix can generate N=27 architectures

- **For many factors, may generate many infeasible architectures**
- How to prevent that?

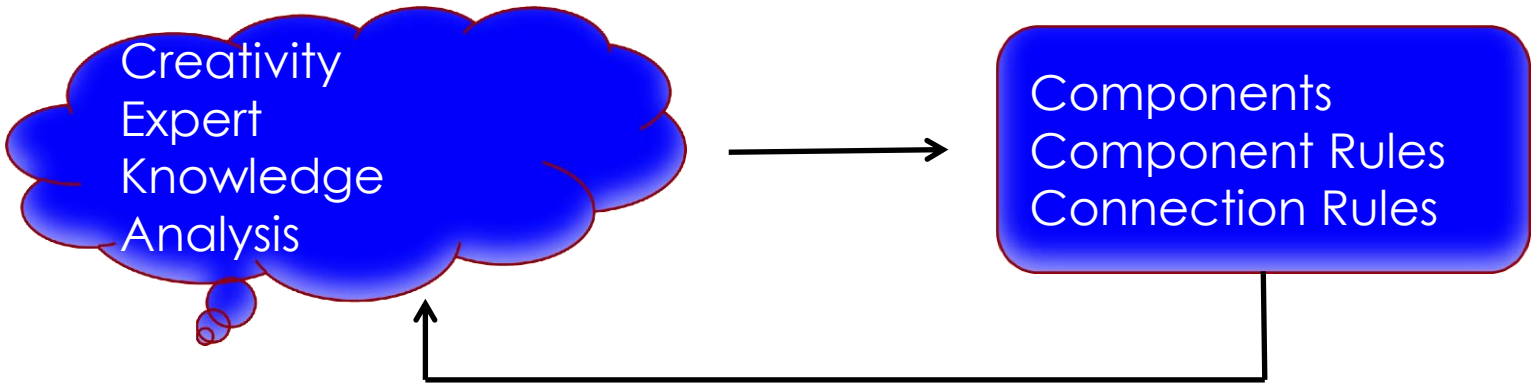
Architecture Enumeration: Overview

- Systems can be described as sets of components or subsystems connected together to form architectures.

- Designers use **creativity**, **previous experience** and **analysis** to compose components/subsystems into architectures which meet functional requirements.

Rule Based Architecture Enumeration

- Express **creativity**, **previous experience and analysis** via rules and components
- Rules constrain which components can exist/coexist
- Rules constrain which connections can exist/coexist



Architecture Enumeration: Abstraction Layers

Abstraction Layers: Formal Division of design into different layers of fidelity and subsystem aggregation

1) Intellectually manageable number of components and connections

2) Minimize the computational resources required to run each level

Abstraction Layer 0
Tail

Abstraction Layer 1
Tail Architectures

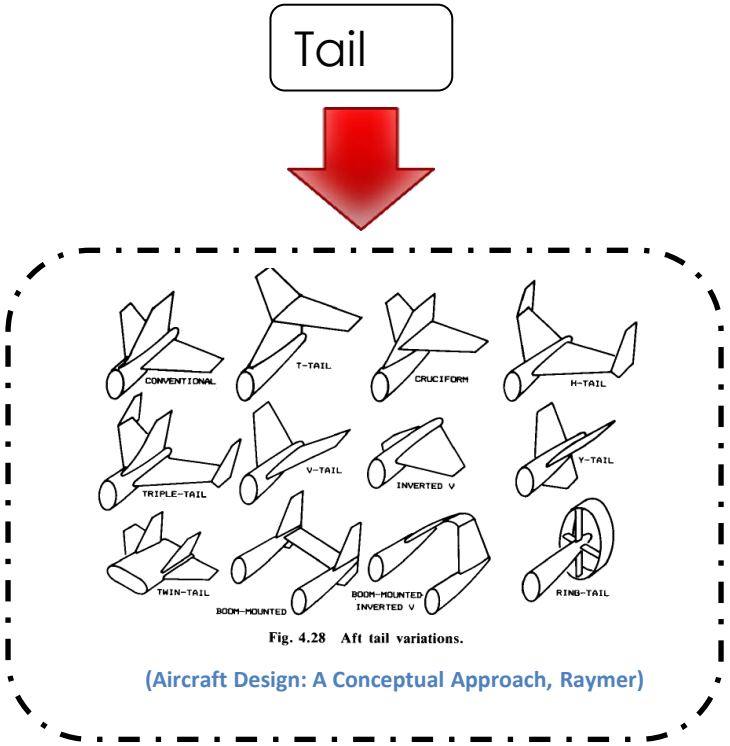
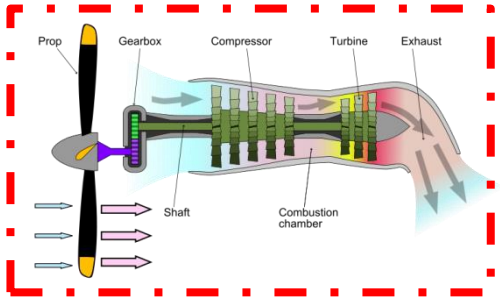


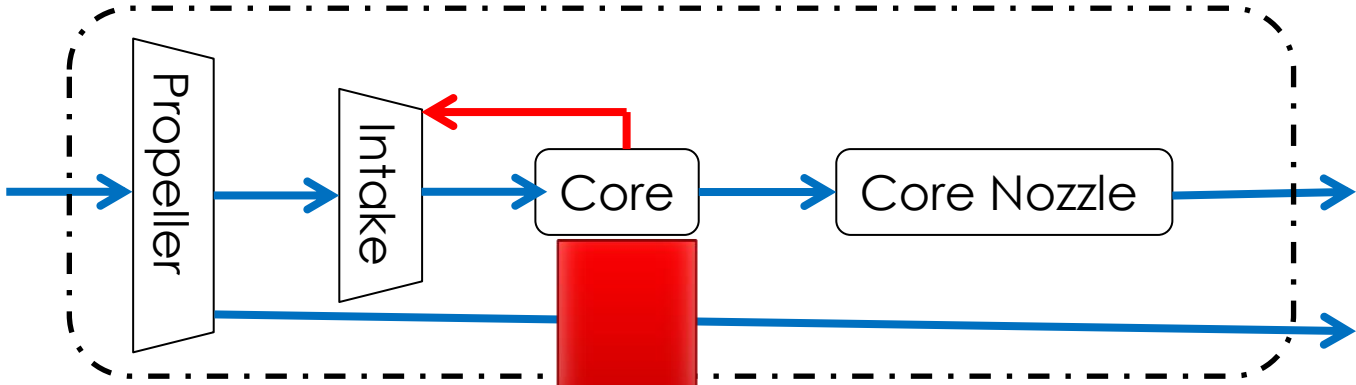
Fig. 4.28 Aft tail variations.

(Aircraft Design: A Conceptual Approach, Raymer)

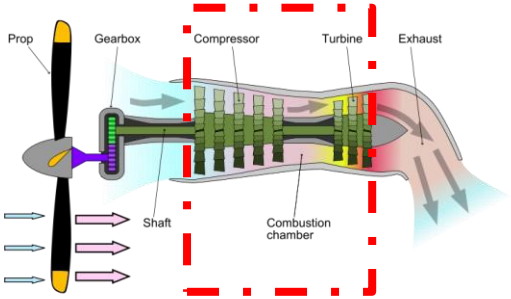
Abstraction Layer Example: Turboprop Core

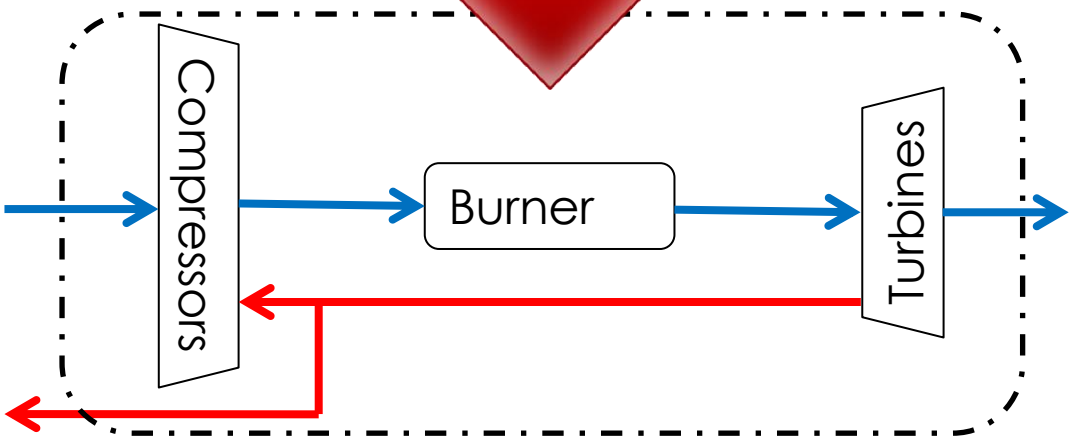
Abstraction Layer 0
Turboprop





Abstraction Layer 1
Engine Core





Architecture Enumeration Tool
240 Potential Designs

Excel Macro based tool from Dr. L. Zeidner at UTRC

Components

Device#	Abbrev	Device Description	Block Name	Parameters min	Parameters max	Variables (unused)	Variables (unused)	Variables (unused)
1	Single_Hull		Hull	0	1			
2	Double_Hull		Hull	0	1			
3	Tripe_Hull		Hull	0	1			
4	Conventional_Chasse		Chasse	0	1			
5	Integrated_Chasse		Chasse	0	1			
6	Conventional_Skirt		Skirt	0	1			
7	Total_Lift_Hydrofoil		Hydrofoil	0	1			
8	Partial_Lift_Hydrofoil		Hydrofoil	0	1			
9	Vertical_Lift_Fan		Vertical_Propulsor	0	1			
10	Single_Air_Prop		Horizontal_Propulsor	0	1			
11	Double_Air_Prop		Horizontal_Propulsor	0	1			
12	Singe_Water_Prop		Horizontal_Propulsor	0	1			
13	Double_Water_Prop		Horizontal_Propulsor	0	1			
14	Single_Waterjet		Horizontal_Propulsor	0	1			
15	Double_Waterjet		Horizontal_Propulsor	0	1			

Rules

Y	DefineRule		Comparison	#BlockInstances	Hull	<=	Constant	1
Y	DefineRule		Comparison	#BlockInstances	Hydrofoil	<=	Constant	1
Y	DefineRule		Comparison	#BlockInstances	Skirt	<=	Constant	1
Y	DefineRule		Comparison	#BlockInstances	Chasse	=	Constant	1
Y	DefineRule		Comparison	#BlockInstances	Horizontal_Propulsor	=	Constant	1
Y	DefineRule		Comparison	#BlockInstances	Vertical_Propulsor	<=	Constant	1
Y	DefinePremise	Skirt_Premise	Comparison	#BlockInstances	Skirt	=	Constant	1
Y	DefinePremise	No_Hull	Comparison	#BlockInstances	Hull	=	Constant	0
Y	DefinePremise	Hull_Premise	Comparison	#BlockInstances	Hull	=	Constant	1
Y	DefinePremise	No_Skirt	Comparison	#BlockInstances	Skirt	=	Constant	0
Y	DefinePremise	No_Total_Lift_Hydrofoil_Premise	Comparison	#DeviceInstances	Total_Lift_Hydrofoil	=	Constant	0
Y	DefinePremise	Vertical_Propulsor_Premise	Comparison	#BlockInstances	Vertical_Propulsor	=	Constant	1
Y	DefineRule		IF	Skirt_Premise	THEN	Vertical_Propulsor_Premise		1
Y	DefineRule		IF	No_Hull	THEN	Skirt_Premise		1
Y	DefineRule		IF	Skirt_Premise	THEN	No_Hull		1
Y	DefineRule		IF	No_Skirt	THEN	Hull_Premise		1
Y	DefineRule		IF	Hull_Premise	THEN	No_Skirt		1
Y	DefineRule		IF	Skirt_Premise	THEN	o_Total_Lift_Hydrofoil_Premise		1

Lecture Summary

- **System Architecture** is the most abstract, but also the most influential activity in Systems Engineering
  - Concept = Mapping Function to Form
  - Down ~2-levels of abstraction, not all the details
- NASA SE Approach focuses on "**Logical Decomposition**"
- Concept Generation is a **creative activity**
  - Group Dynamics: e.g. Brainstorming
  - Models: e.g. Leonardo da Vinci
  - Structured Processes: MindMaps, Morphological Matrix, Architecture Enumeration
- In **Assignment A3** you will generate concepts for 2016 Cansat using at least two different creativity methods of your choice.

16.842 Fundamentals of Systems Engineering
Fall 2015