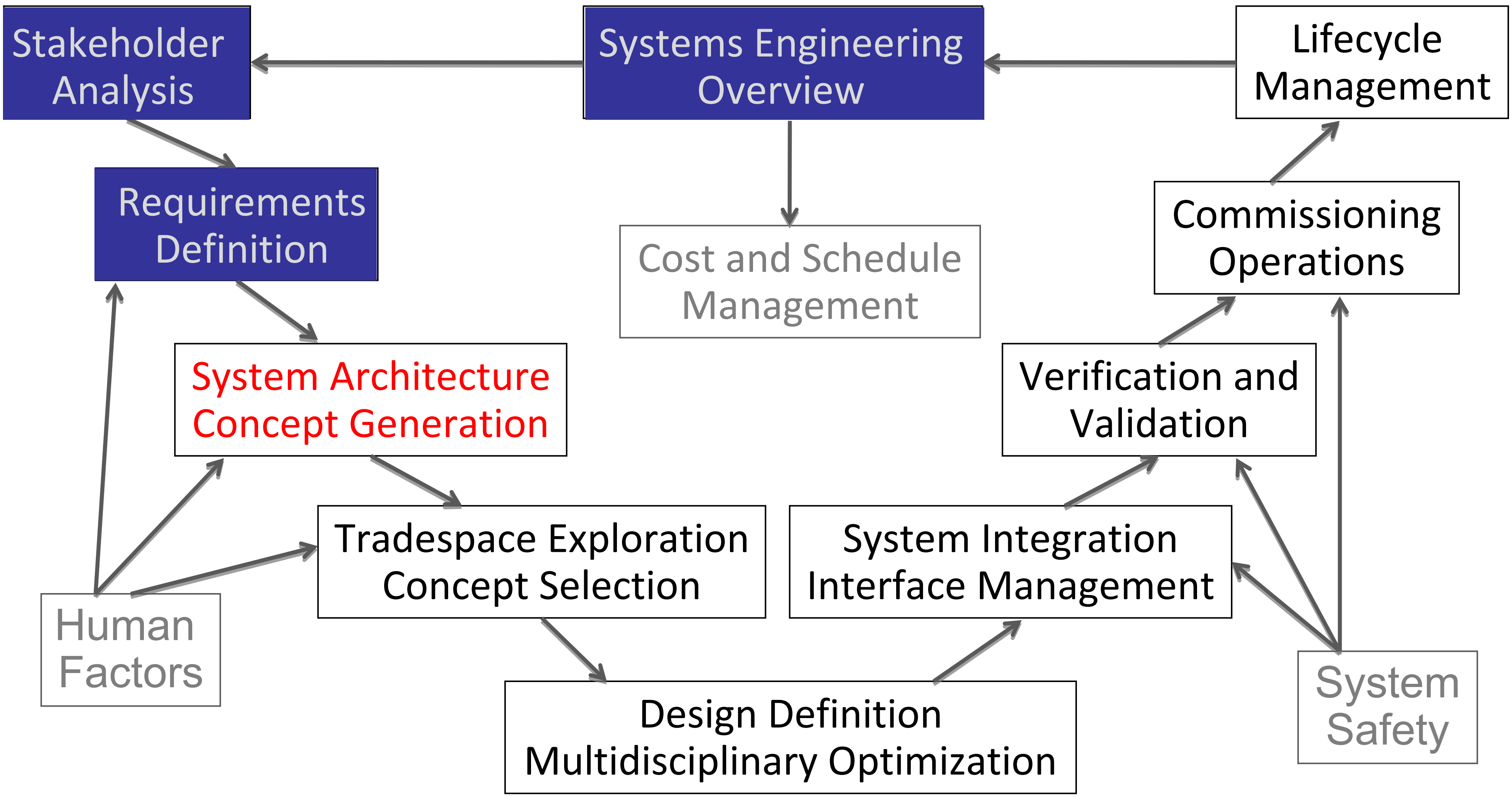
System Architecture
Concept Generation

October 2, 2009

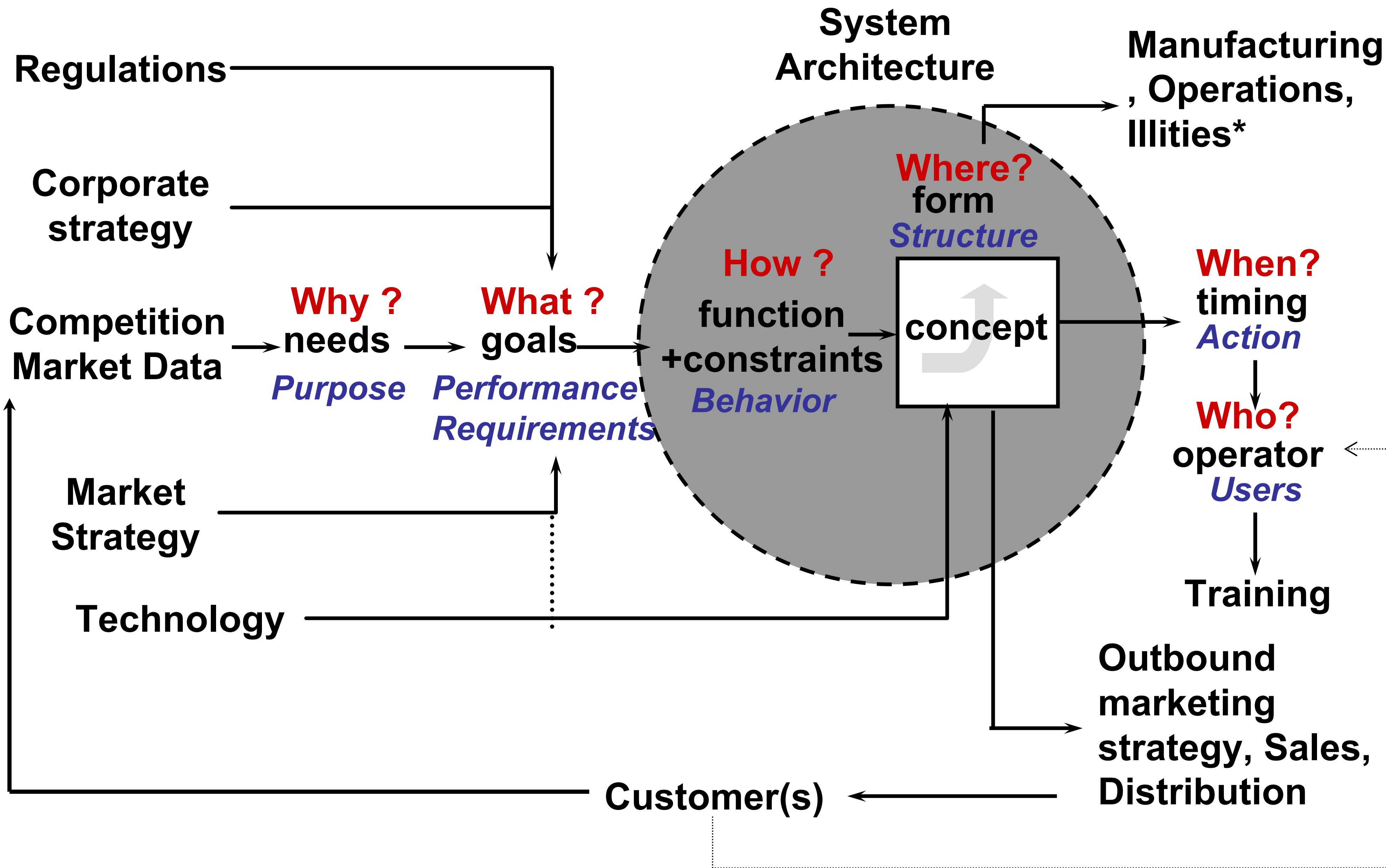
Prof. Olivier de Weck

Note: System Architecture is a very rich topic that can take up an entire semester by itself. ESD.34 is a recommended course (E. Crawley) and a number of slides in this lecture are adapted from it.

V-Model – Oct 2, 2009

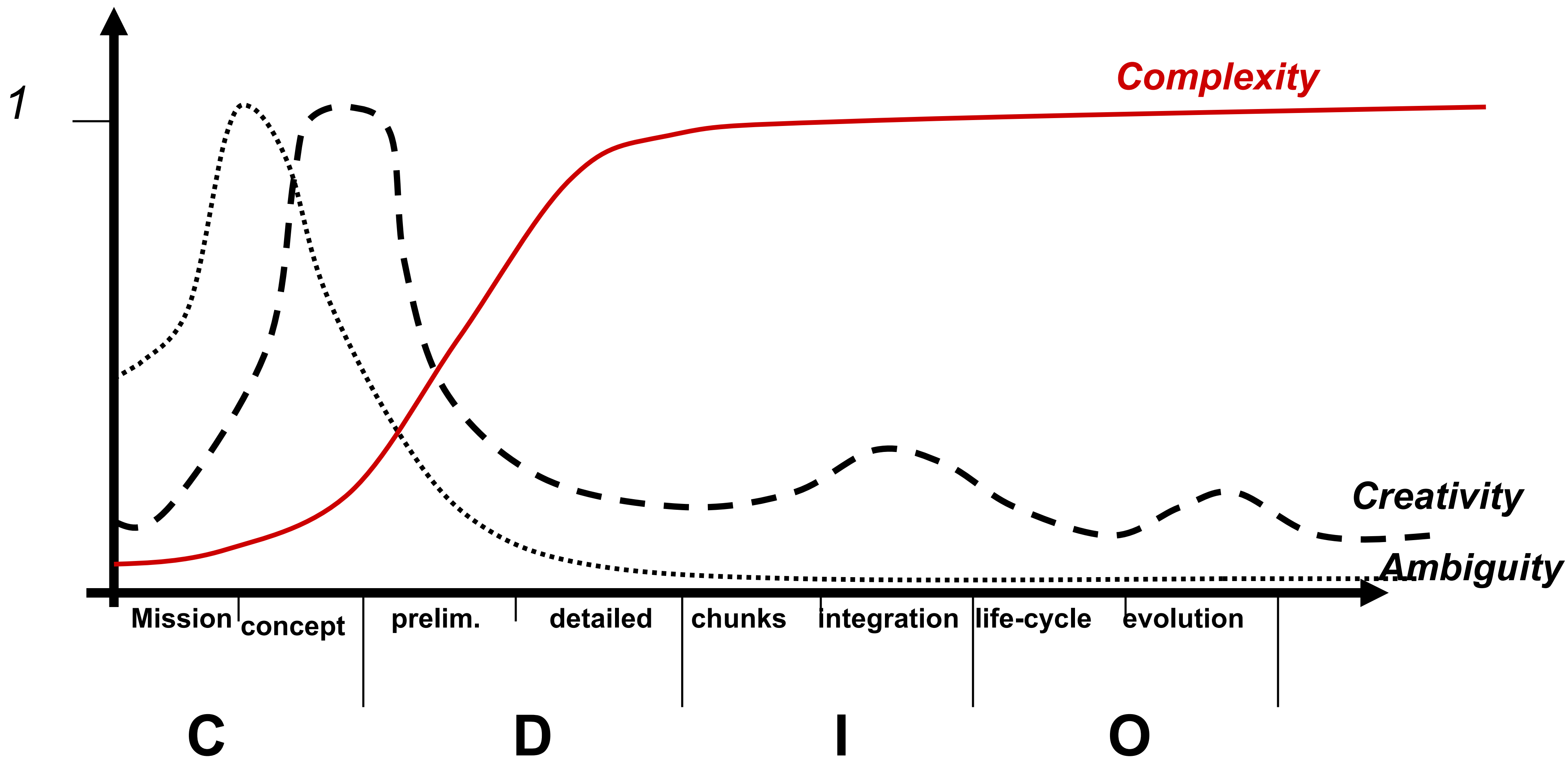


System/Product Architecture Framework (ESD.34)



*Reliability, Servicability, Environmental Impact, Upgradeability, Flexibility,etc…

Themes: Ambiguity-Creativity-Complexity

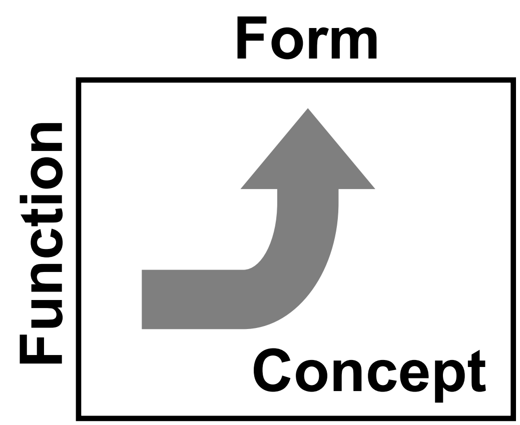


Early on ambiguity is high -> reduce ambiguity
Next concept are needed -> focus creativity
Then complexity starts increasing -> manage complexity

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

These images of a beach and contemporary style houses and corresponding floor plans have been removed due to copyright restrictions. See http://www.coolhouseplans.com for examples.

Beach

Contemporary

Architecture - Informational

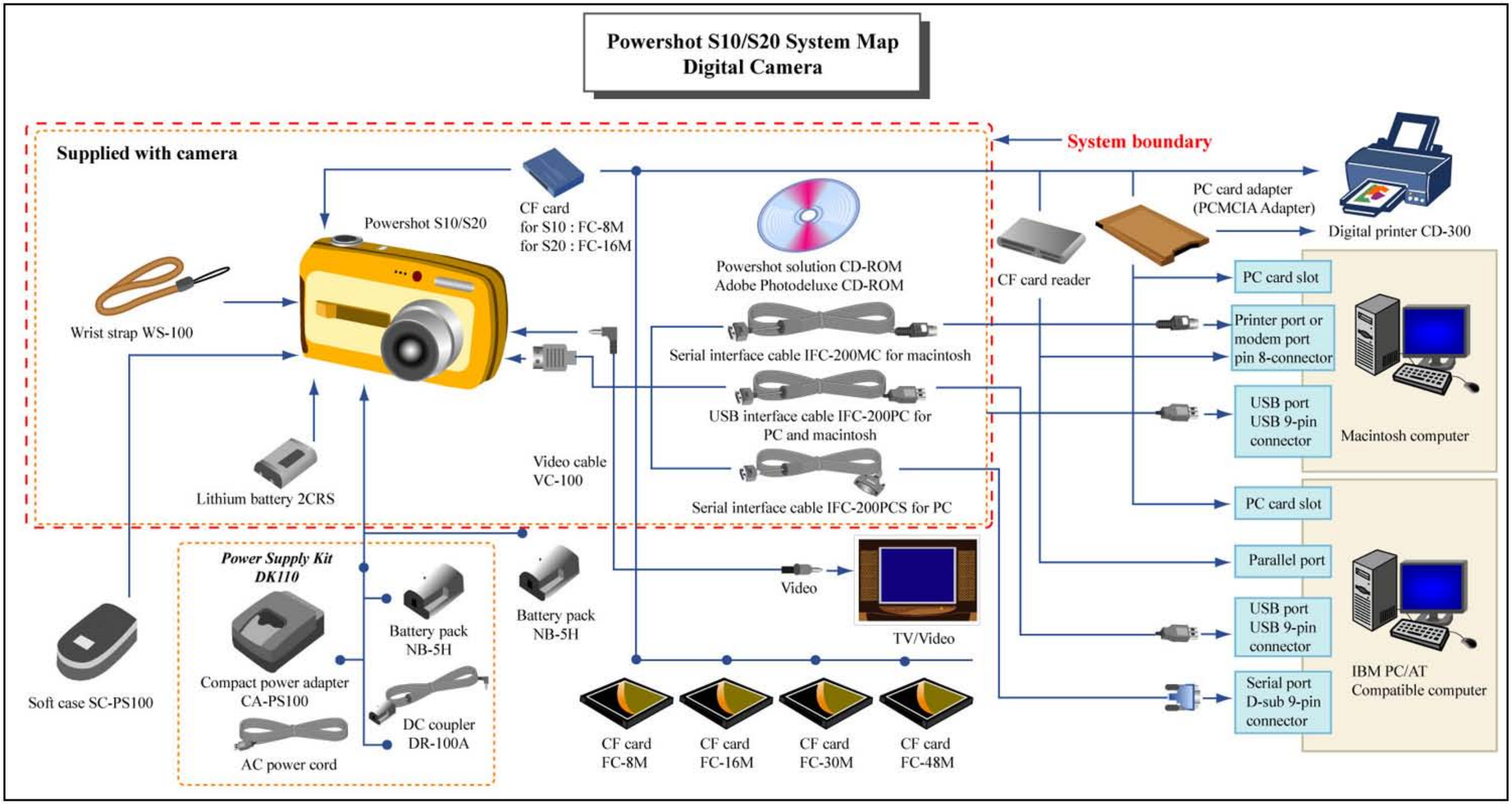
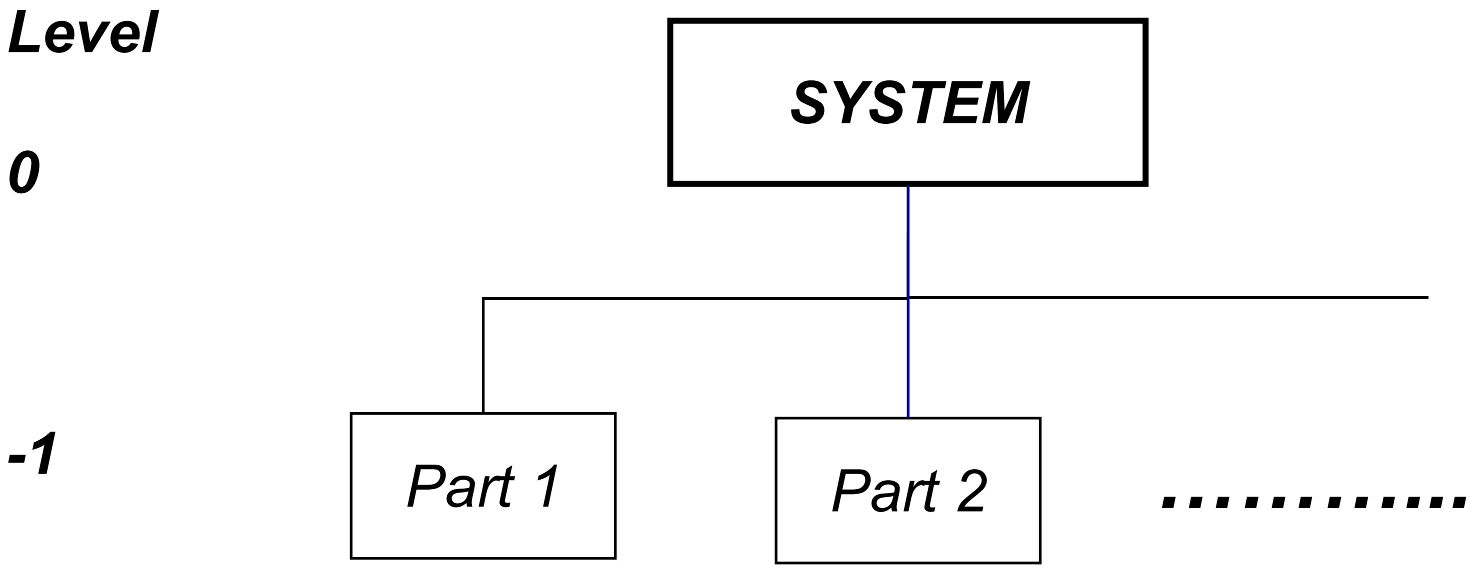


Image by MIT OpenCourseWare.

Form - Defined

- The sum of the elements (objects)
- The structure or arrangement of the physical/logical embodiment
- The shape or configuration
- (often but not always) What can be seen
- What is implemented (formed, manufactured, assembled, written, sculpted or drawn)
- What **it is**.

Form of a Simple System



- Generally 5-9 parts (7+/- 2)
- At level -1 we encounter real or **atomic** parts
  - A part cannot be taken **a-part** without loosing its functionality or integrity
  - Definition of what is a part is not always unambiguous
- Tree structure is symbolic, and may or may not represent the actual connectivity of the parts (the structure) - all elements on a level can interface, but don't necessarily all do
- Examples ?

Complex Form: FLIR System for Helicopter

These images have been removed due to copyright restrictions.

L-3: Adds/Removes Hardware & Details

L0: Top Kit Collector

L-1: Avionics Sub Kit

L-2: Transition

L-1: Elec Harness Sub Kit

L-1: Airframe Sub Kit

L-2: Turret Avionics

L-2: Cockpit Avionics

L-2: Cockpit, LBL Beam

L-2: Nose Floor

L-2: Cockpit, RBL Beam

L-2: Turret Support

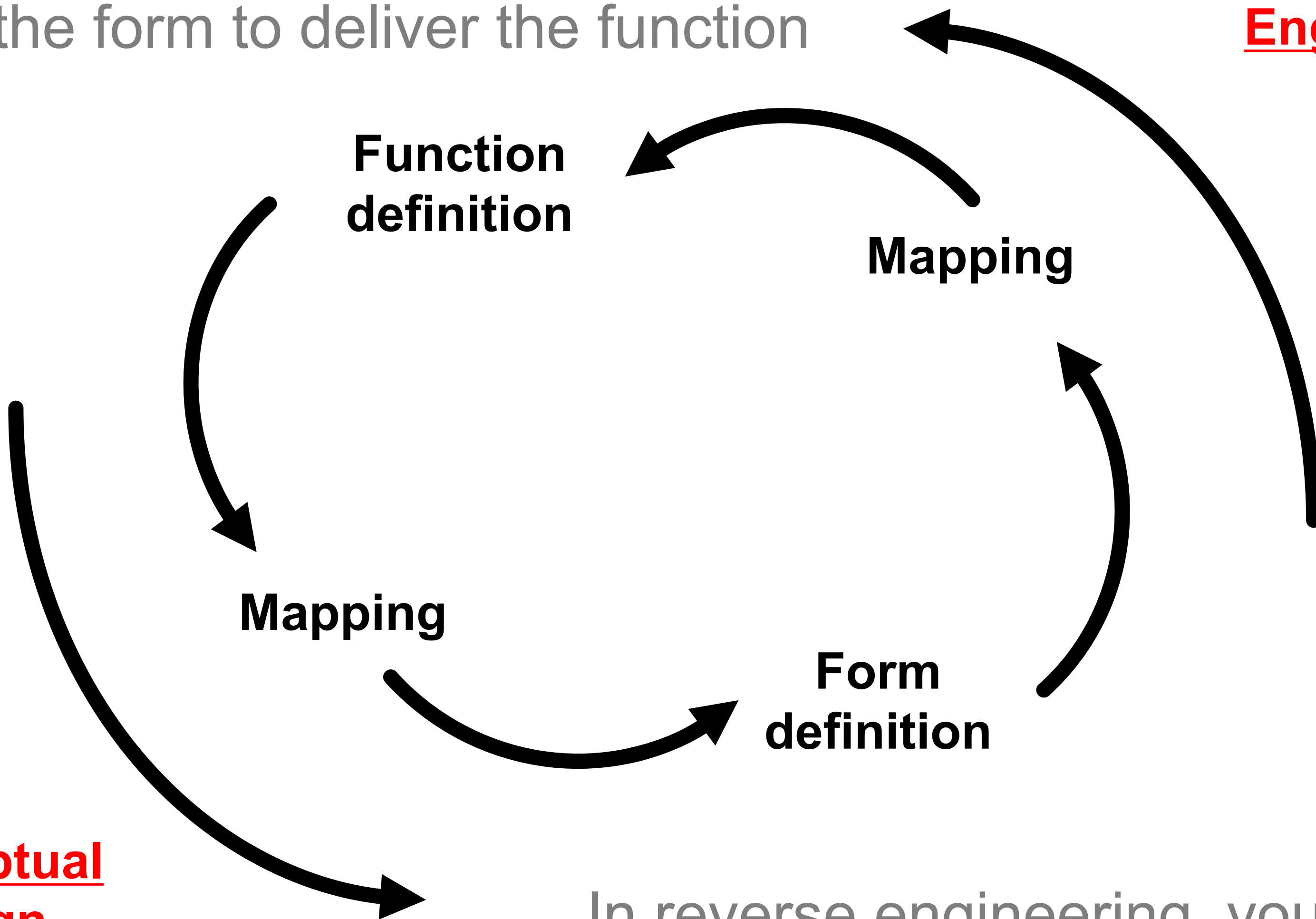
L-2: Cabin

Function

- The activities, operations and transformations that cause, create or contribute to performance (i.e. meeting goals)
- The actions for which a thing exists or is employed
- What the product/system **does.**
- Is what the system eventually does, the activities and transformations which emerges as sub-function aggregate
- Can be decomposed about one level before concept is required
- Can show connectivity of function - mass (material), momentum (force), energy (power), information (data), information (commands)
- Is more difficult to represent than form (because “invisible”)

Architecting Sequence

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

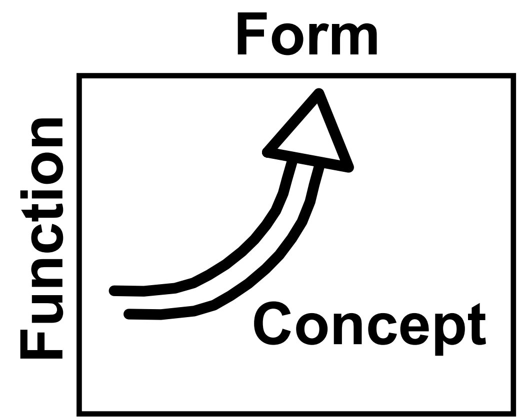


In reverse engineering, you know the form, and are trying to infer the function (and presumably eventually the goals)

Concepts

- Defined - informally
- Defined - formally
- Examples

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

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"
- Products based on different concepts are "disjoint".

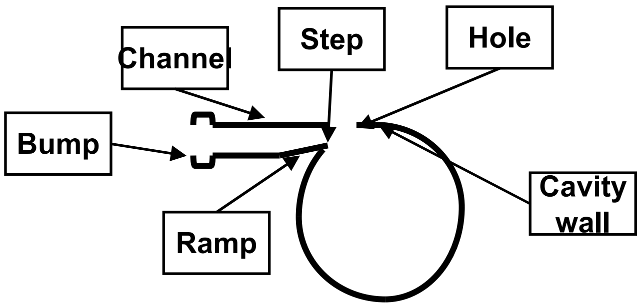
Exercise – 2 min

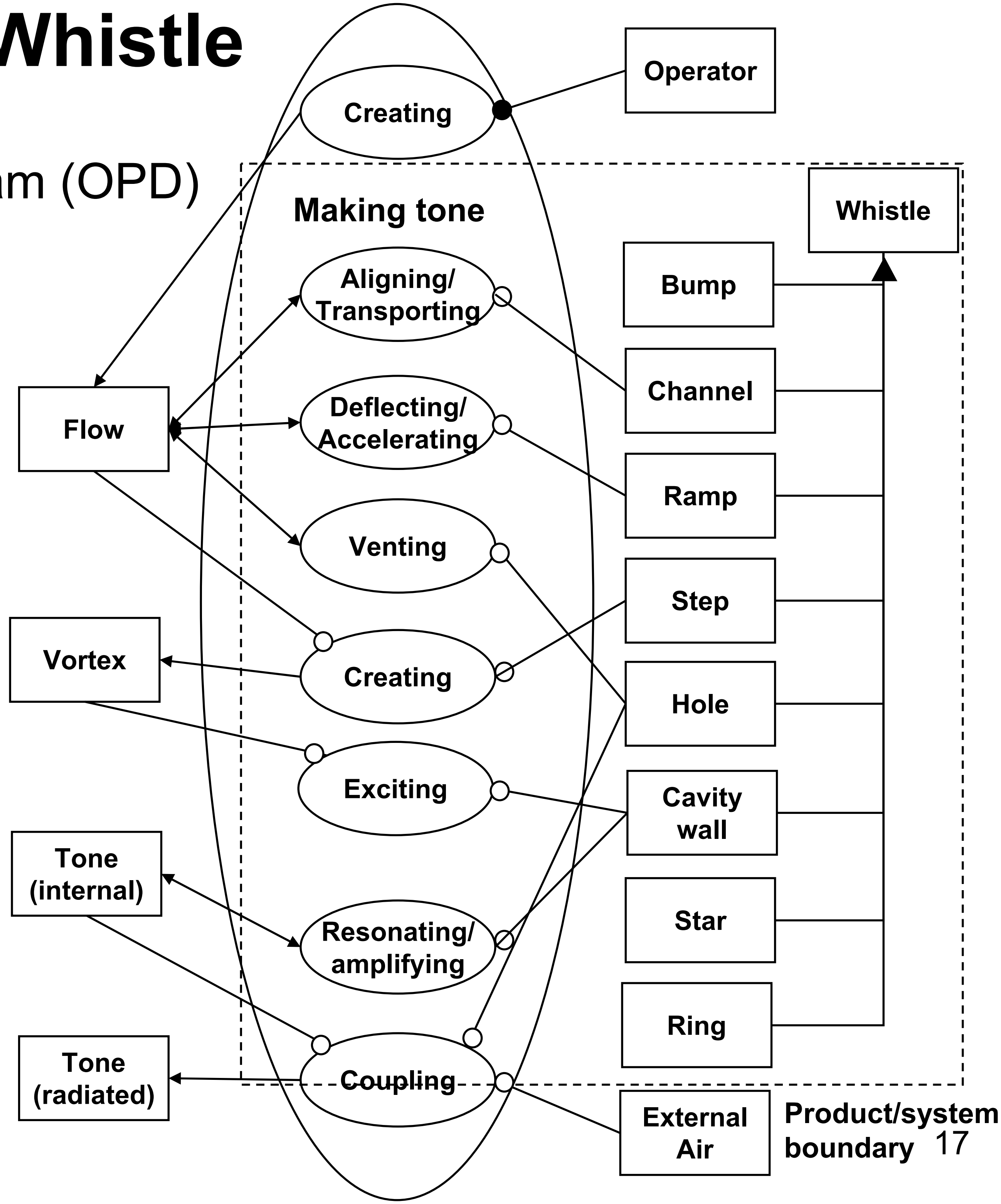
- Describe the concept of one of the following items:
  - Whistle
  - Automobile
  - Aircraft
  - Communications Satellite
  - International Space Station
  - Lecture

Concept: Whistle

Object-Process-Diagram (OPD)

Sketch

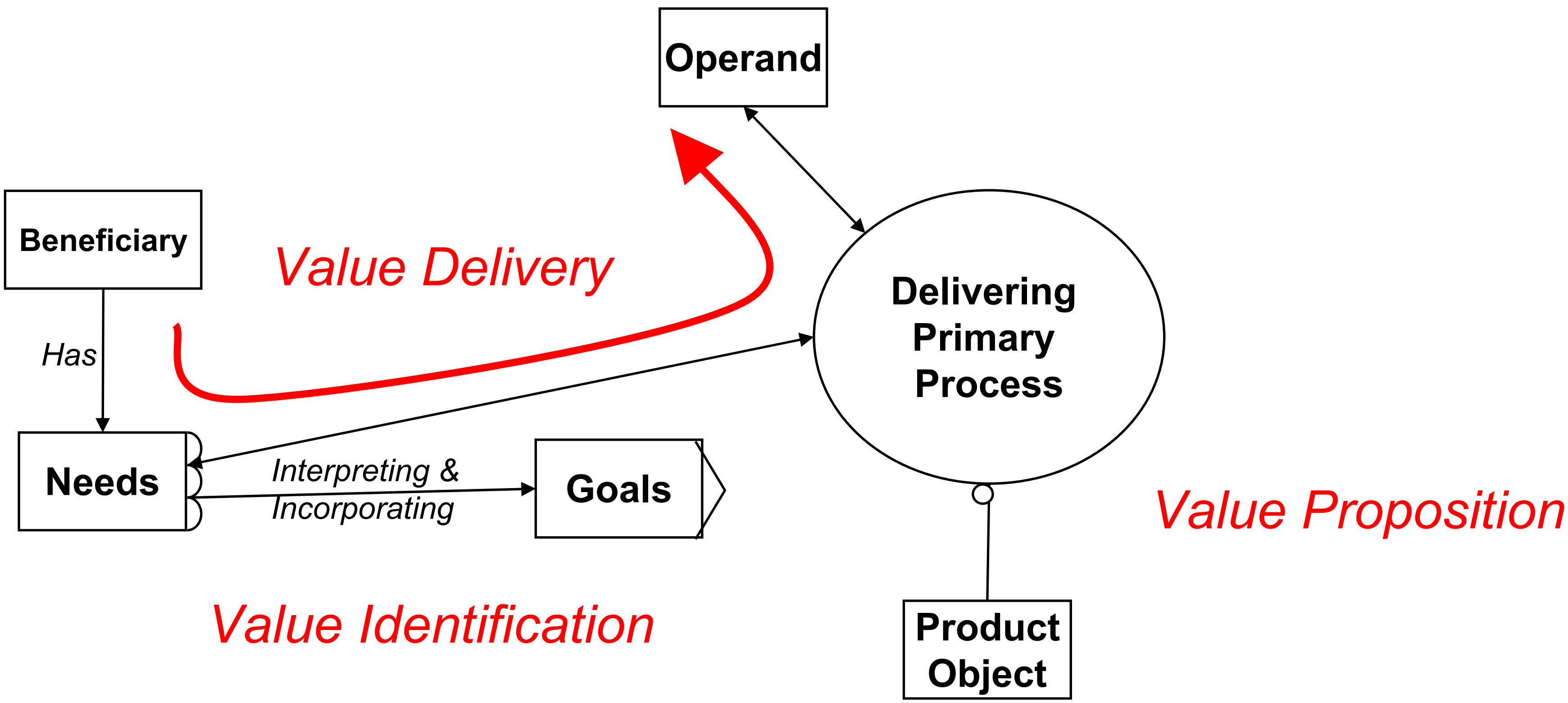




Refrigerator Case Study

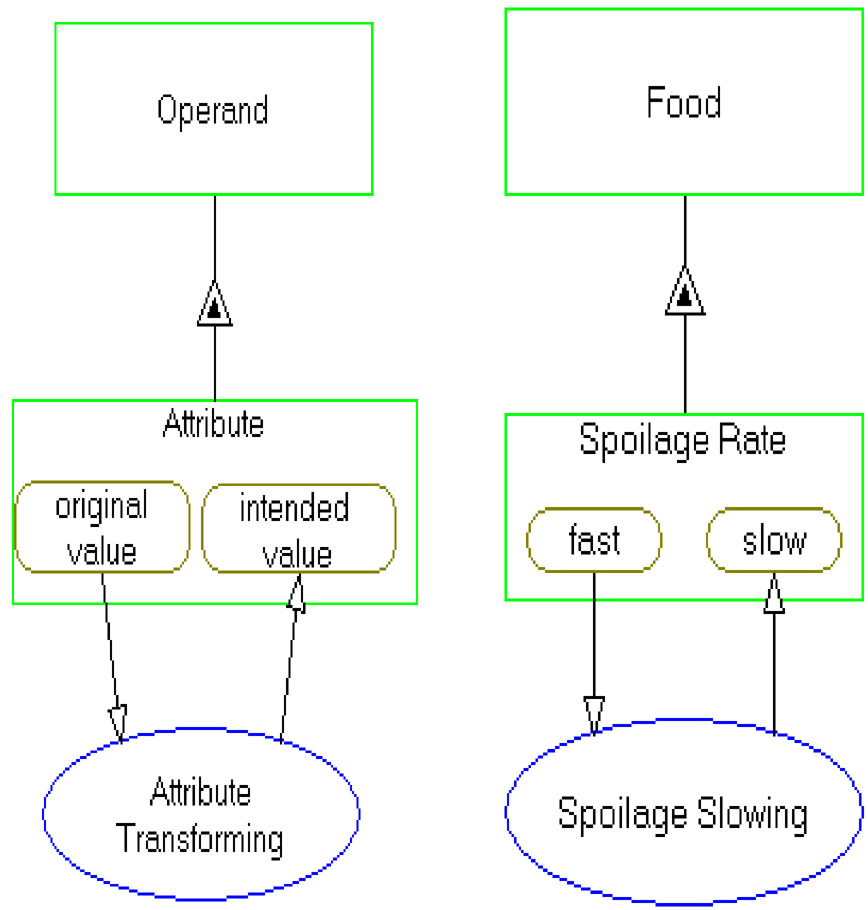
Value - A Formal Definition

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.



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



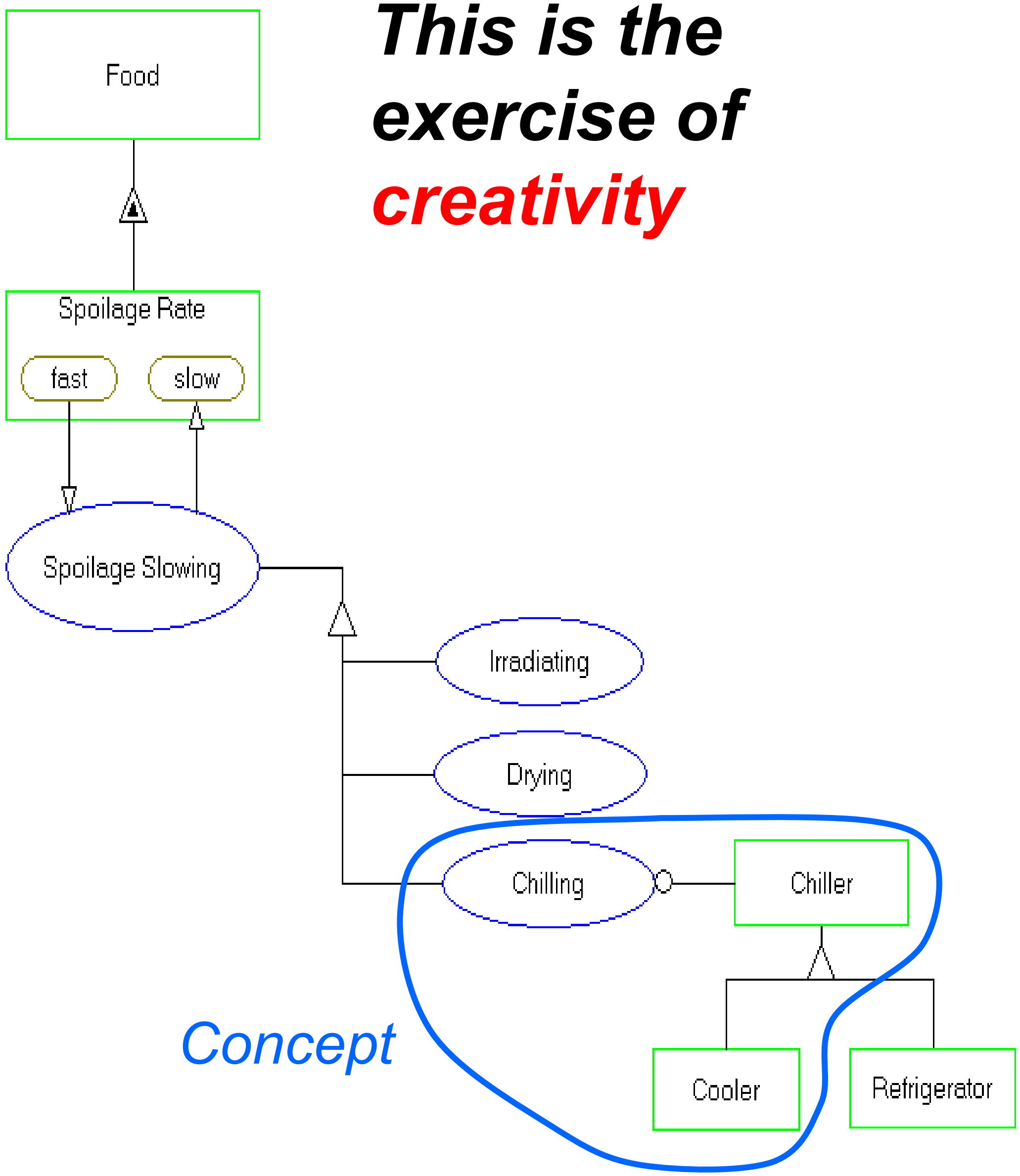
Note: For "Production Systems" the value could be found not in an operand whose attributes are affected but in a resultee that is created

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

Focus Creativity : Concept

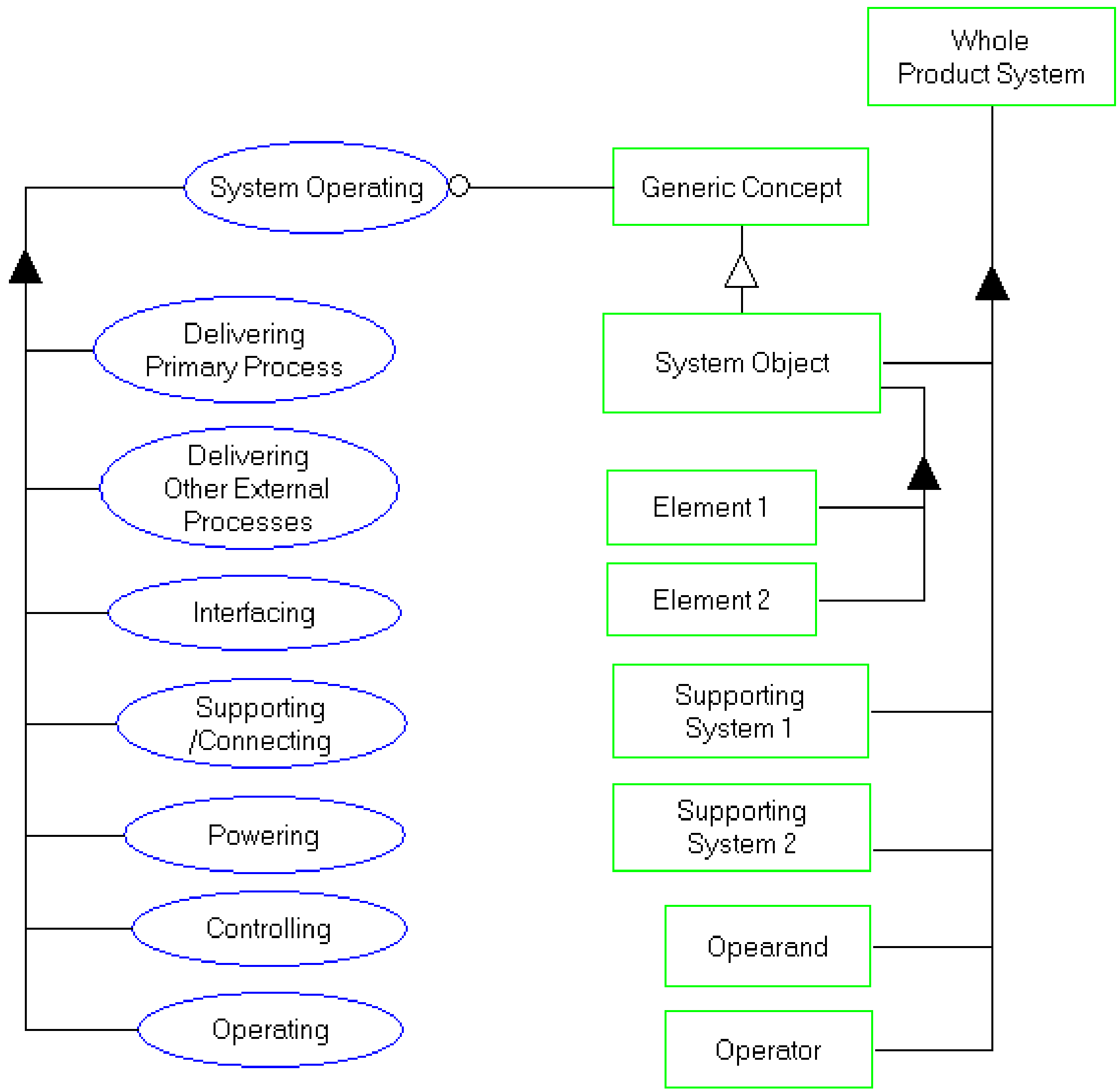
- Concept: a system vision, which embodies working principles, a mapping from function to form
- Choose from among the system operating processing that specialize to the desired solution neutral, value related process
- Specialize the related generic concept to the product form

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



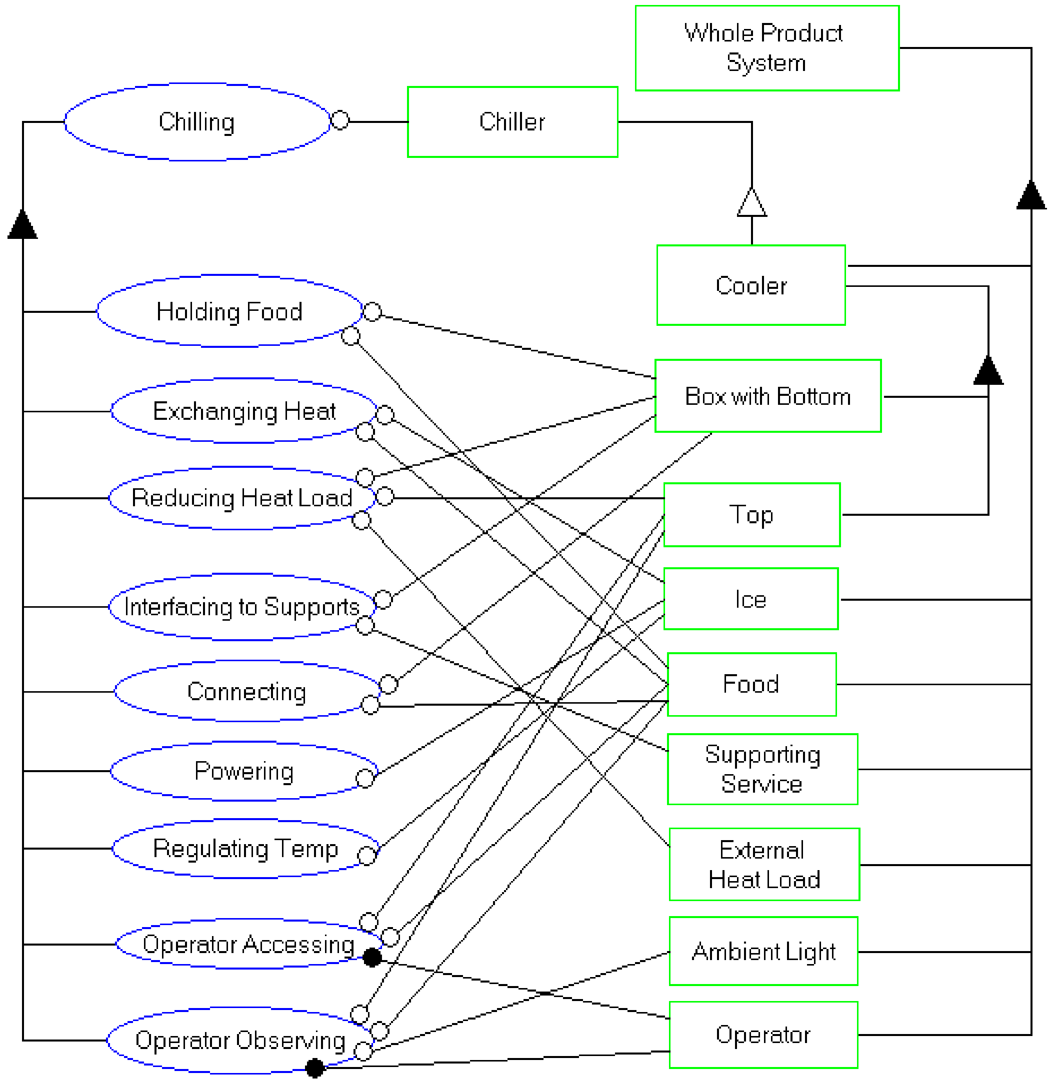
Form and **Function** -Cooler

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

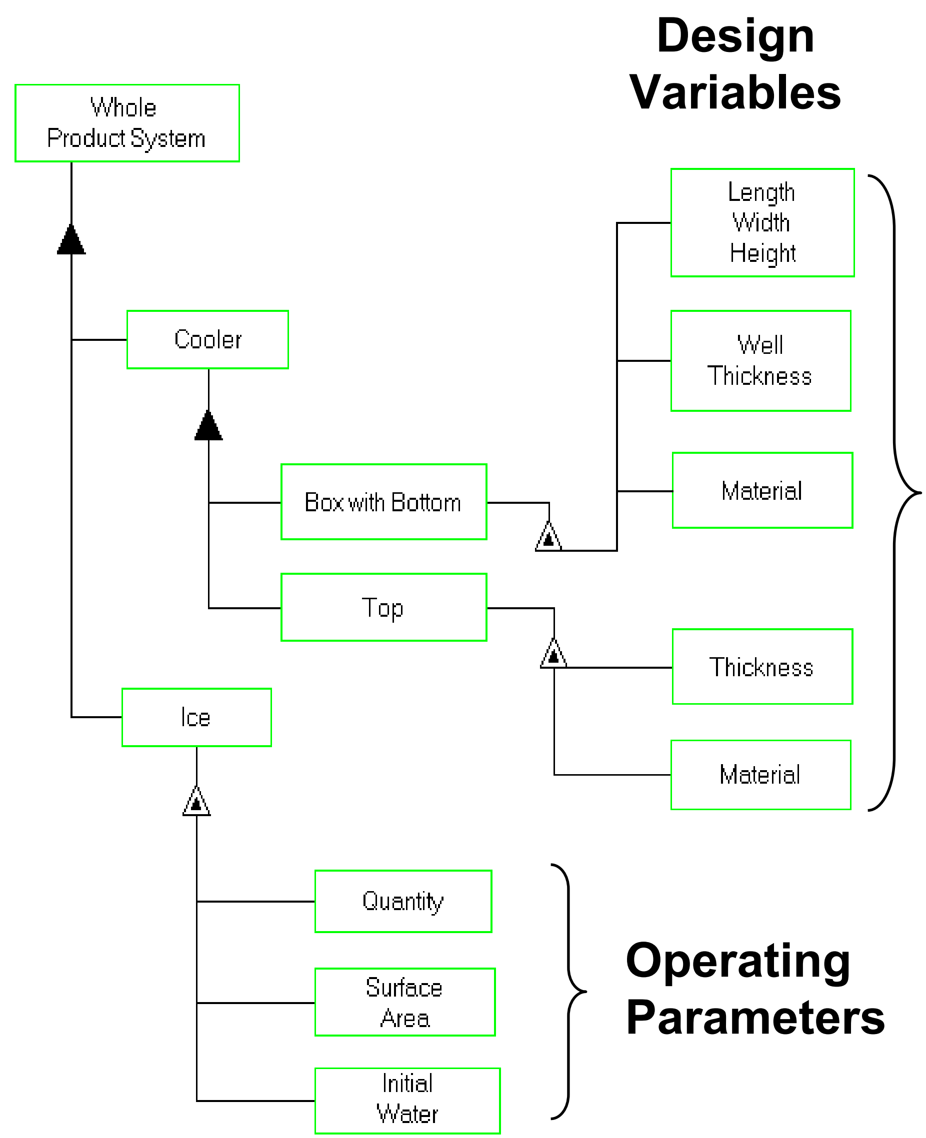
Map objects to processes to determine object-process architecture



Establishing the complexity of the object-process architecture

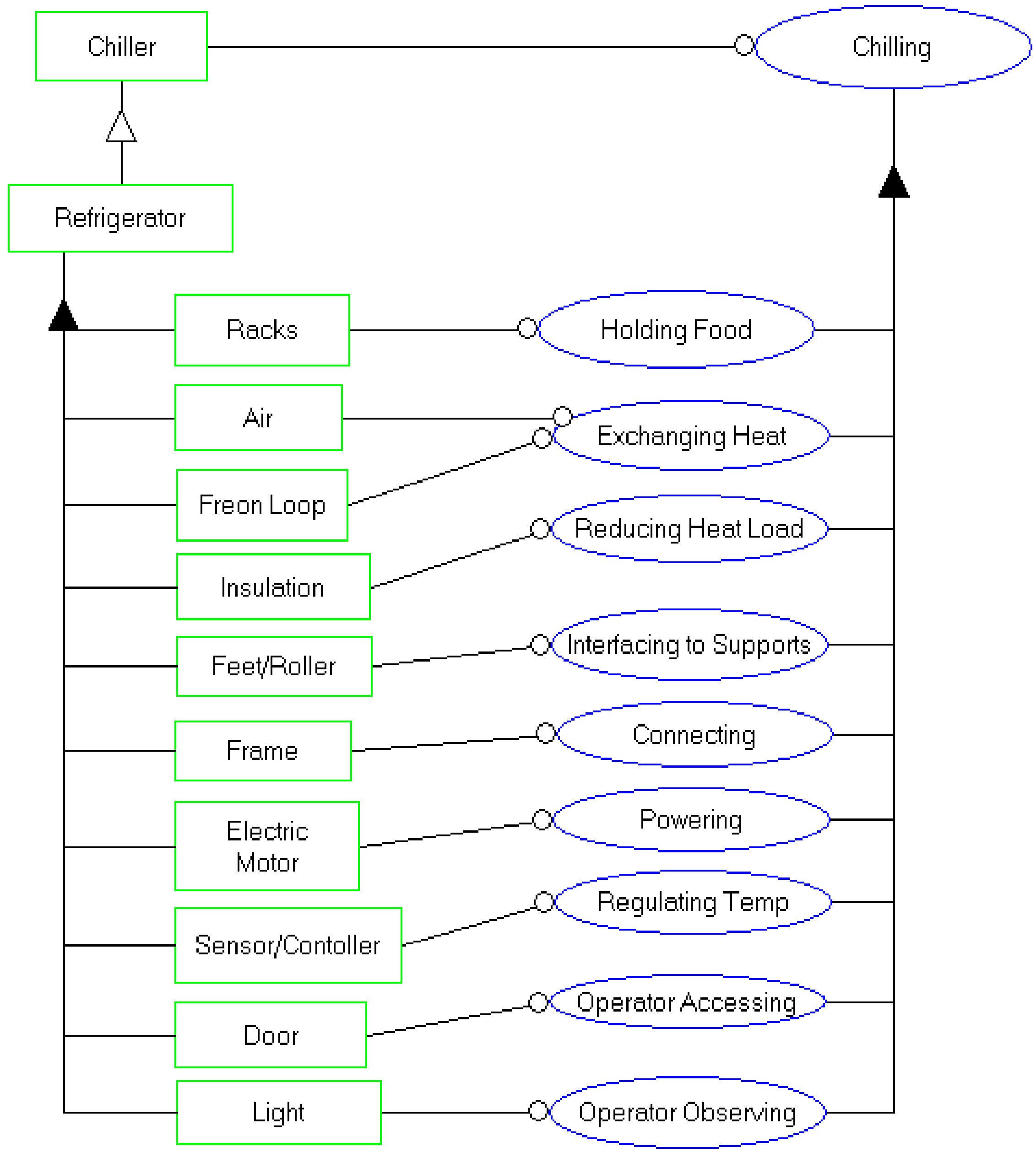
Design vs. Architecture

- Architecture selects the concept, decomposition and mapping of form to function
- Architecture <u>establishes</u> the <u>vector of design variables and operating parameters</u>
- 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

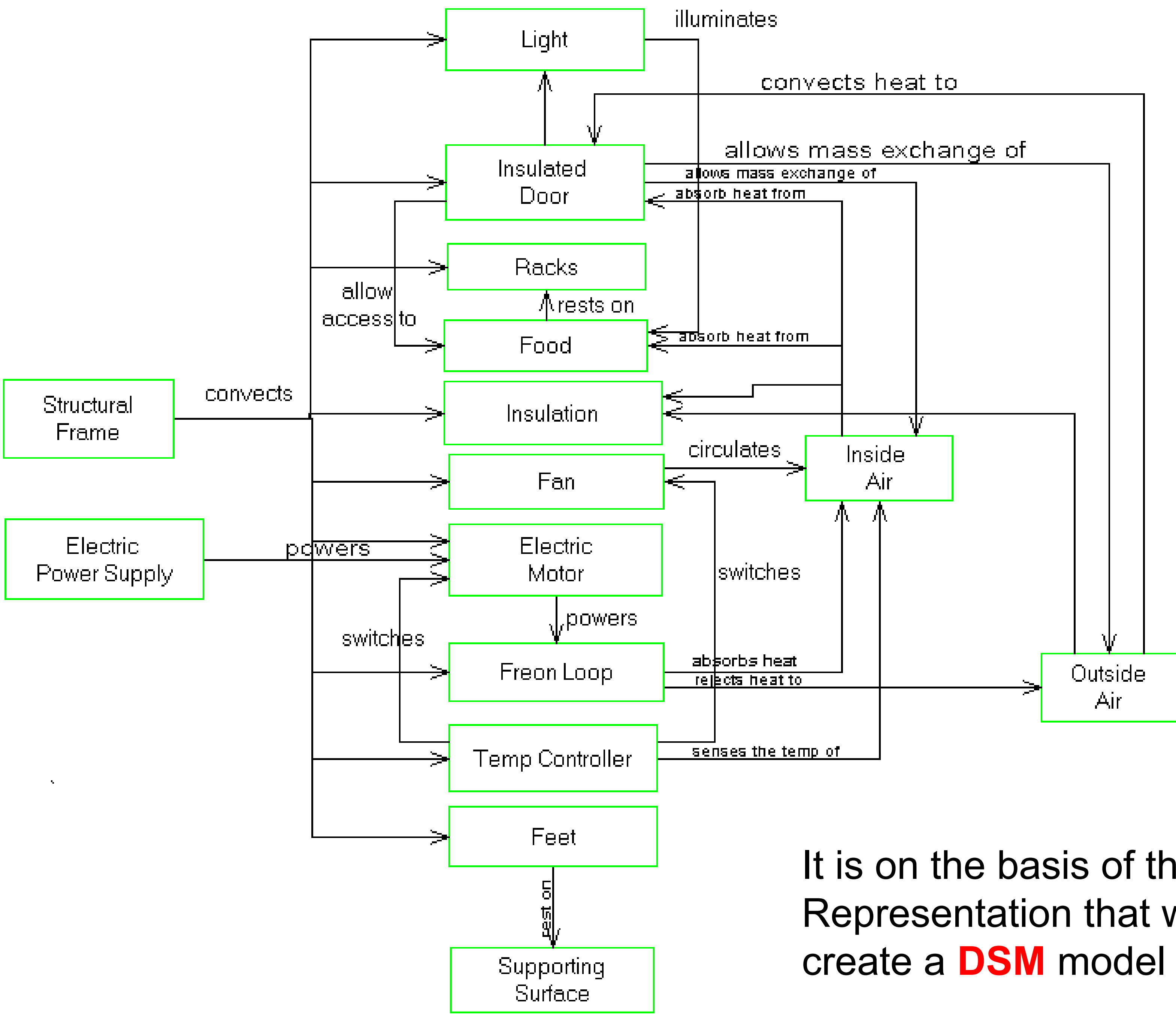


Form and Function - Refrigerator

- 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



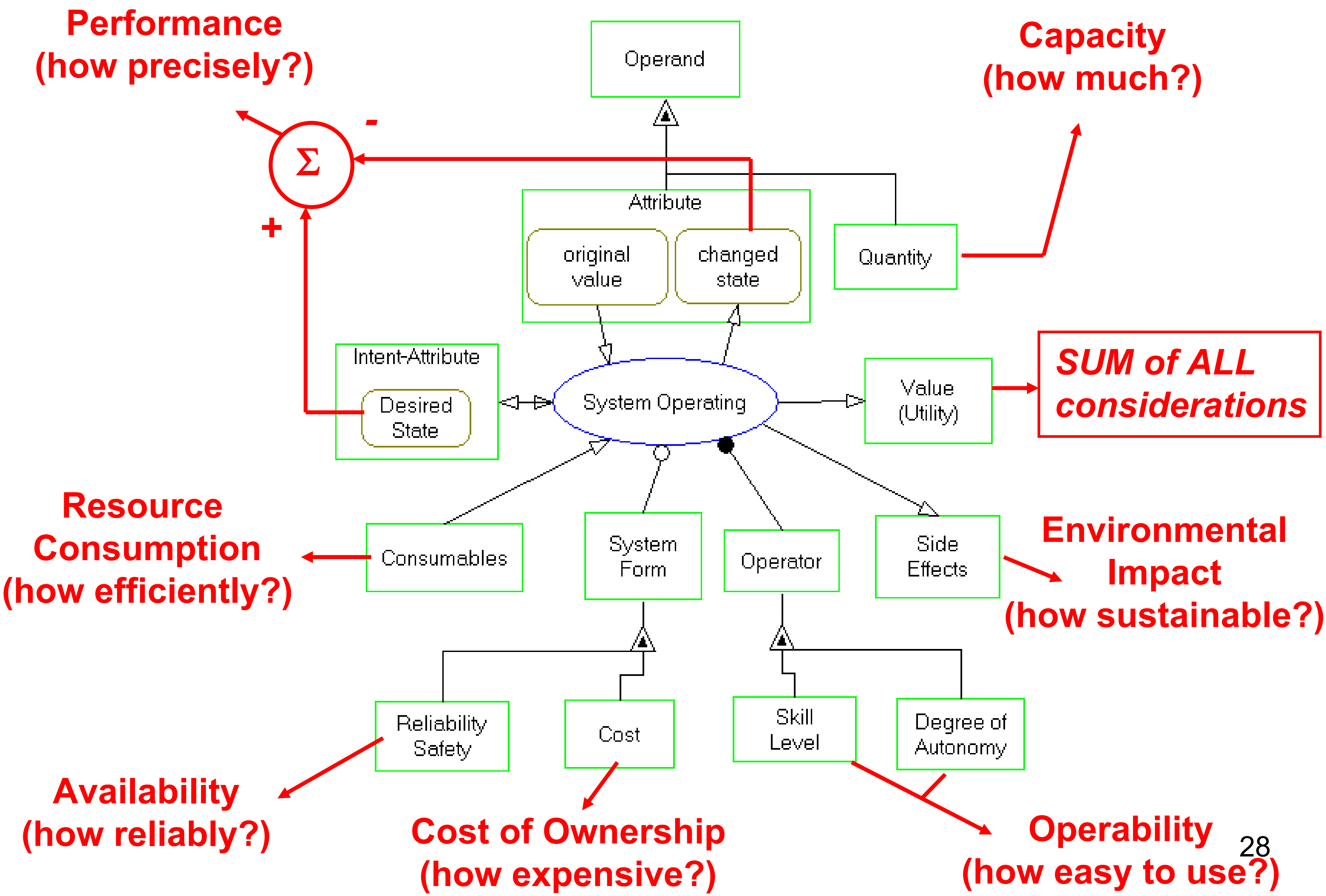
It is on the basis of this Representation that we can create a **DSM** model

Classes of Links

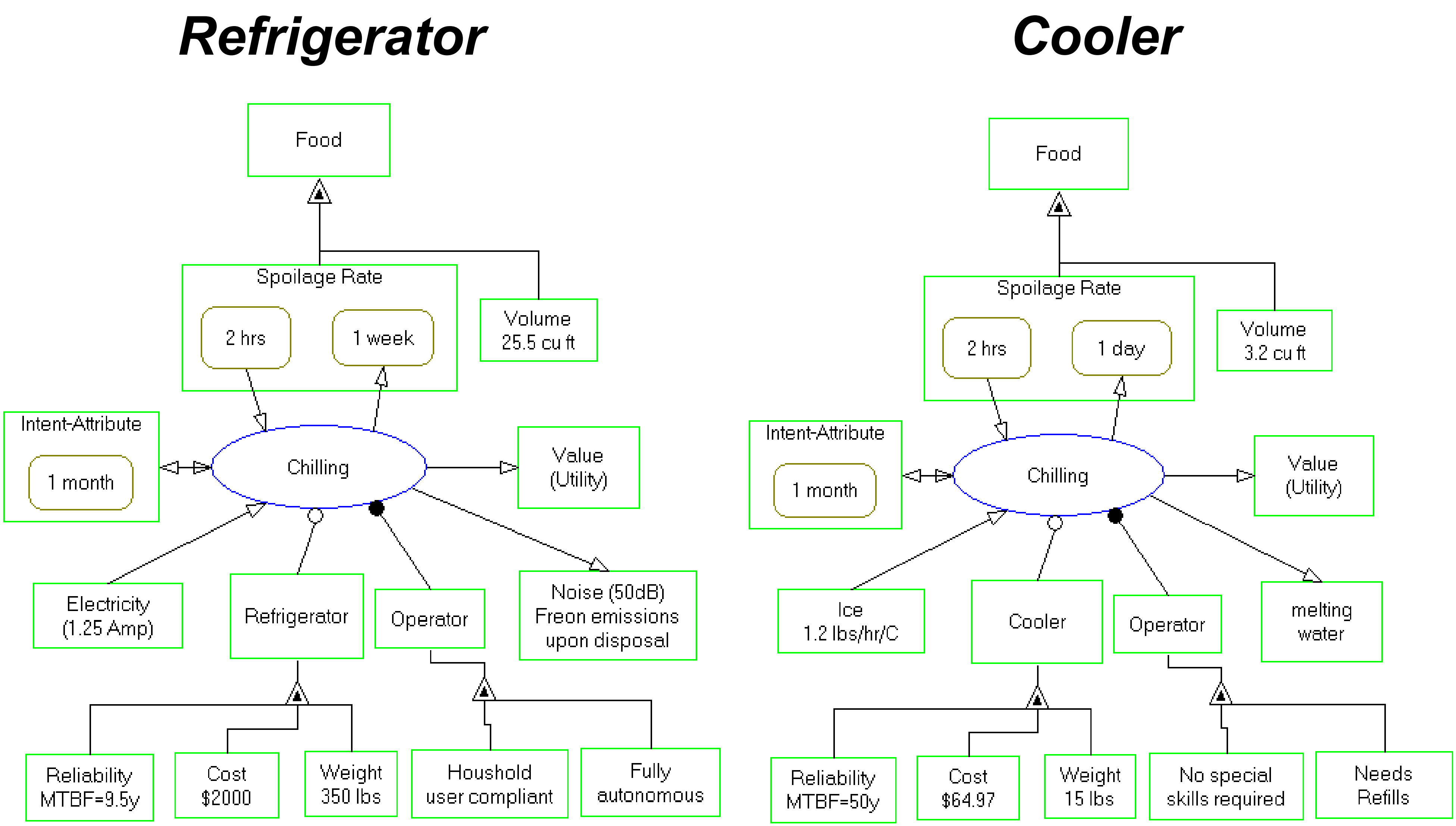
Link Class	Operand	Process	Instrument Form
Physical Connection	Forces, Torques [N, Nm]	Force or Torque Transmitting	bolts, washers, rivets, spot welds…
Energy Flow	Work [J]	Electricity or Heat Transmitting	copper wires, microwaves, …
Mass Flow	Mass [kg]	Fluid, Gas or Solid Matter Transmitting	fuel lines, air ducts, exhaust pipes …
Information Flow	Bits [-]	Data or Command Transmitting	micro-switches, wireless RF, humans

Note: In many cases, in order for an energy, mass or information flow to exist, there also needs to be a physical connection, but not always

Basic Metrics for System "Goodness"



Refrigerator versus Cooler

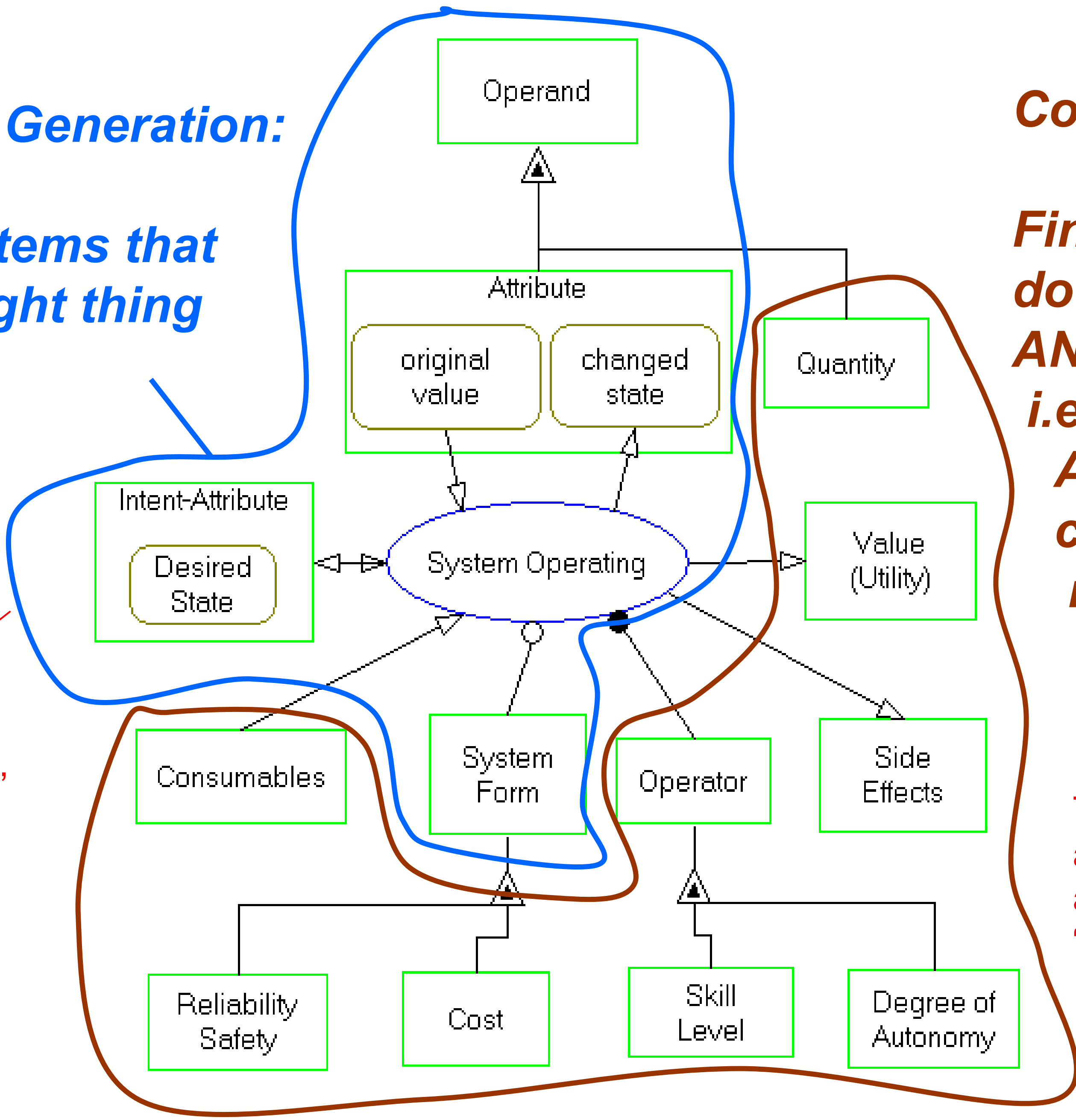


Which of these systems would you choose?

Concept Generation versus Selection

Concept Generation:

Find systems that do the right thing



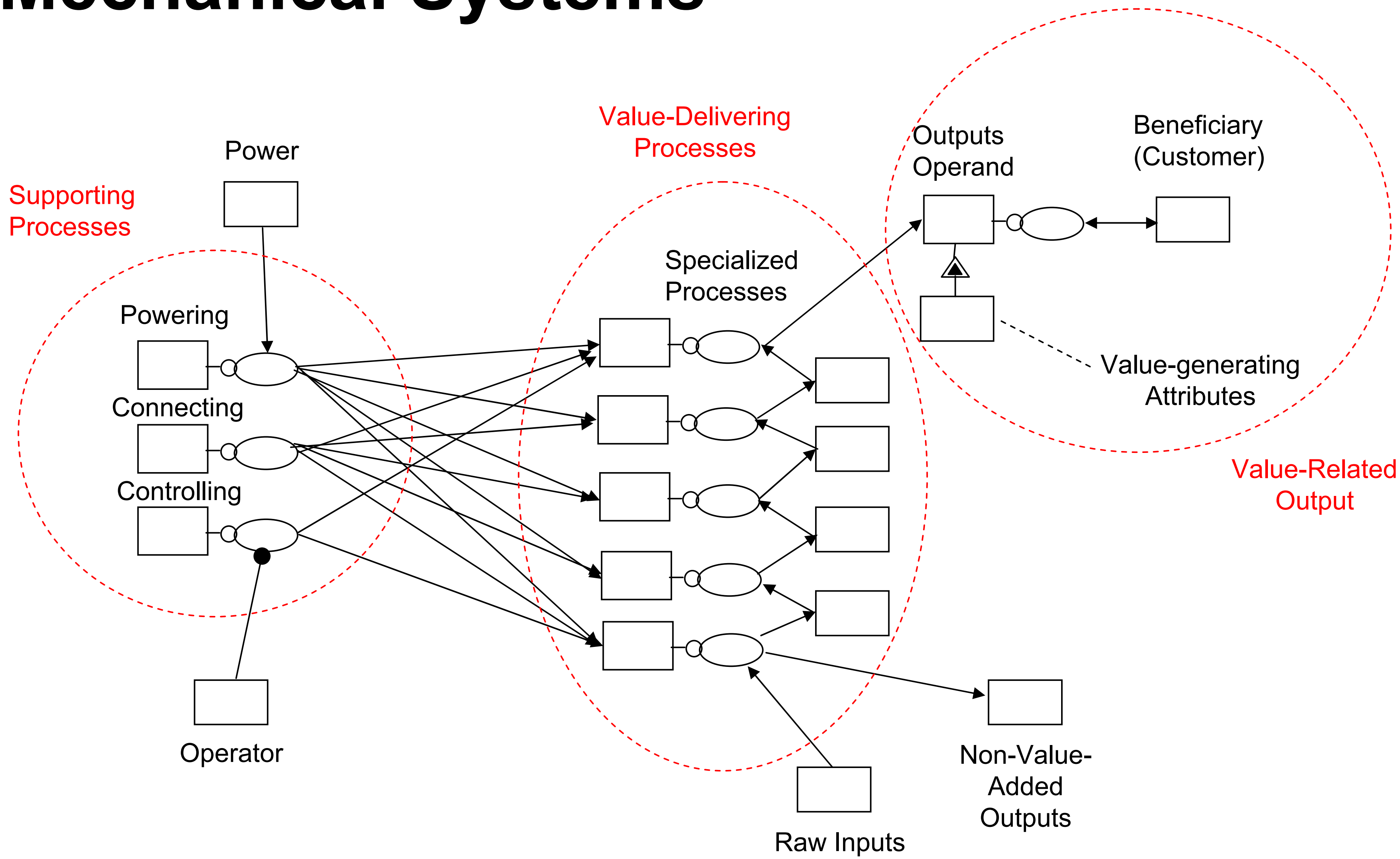
“Disruptive Technologies”

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

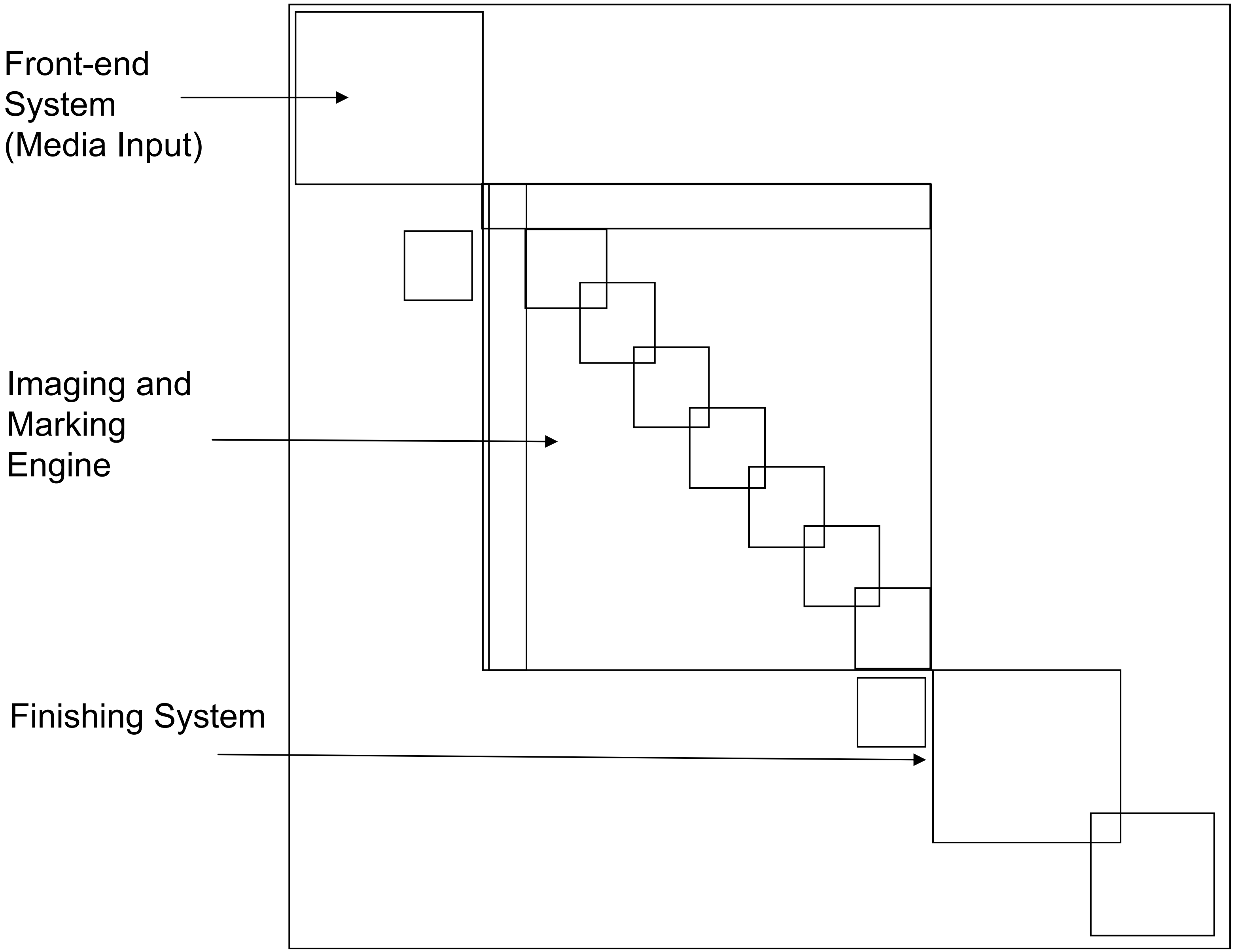
Technology Infusion affects these attributes mainly “Improving Technologies”

General Structure of Complex Electro-Mechanical Systems



Example of High Level Product Architecture (Xerox)

iGen3



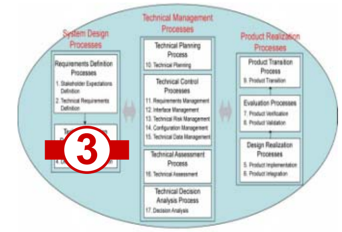
Role Definition of a System/Product Architect

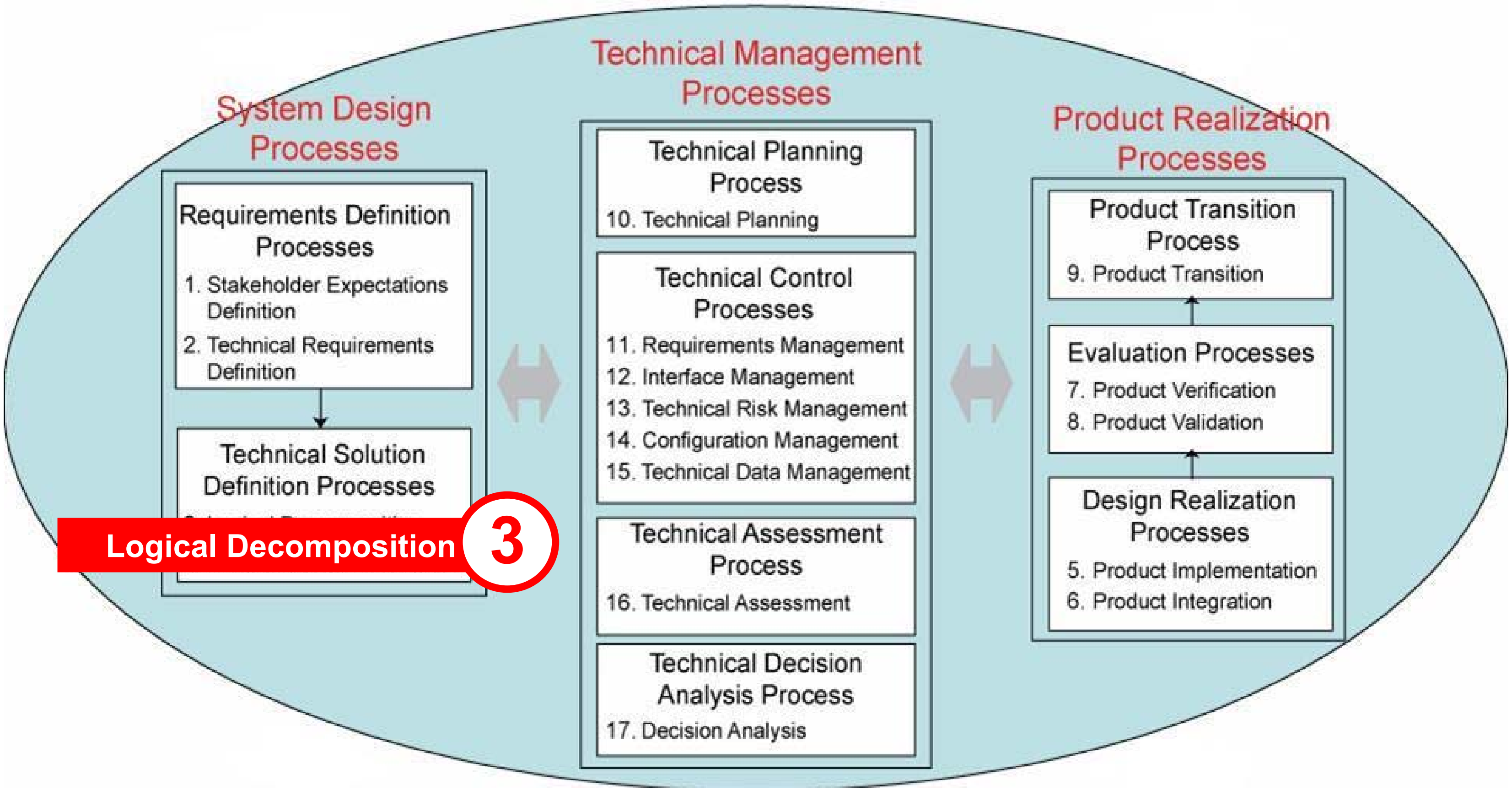
- The architect performs the most abstract, high level function in product development
- The architect is the driving force of the conceptual phase
- The architect
  - Defines the boundaries and functions
  - Creates the Concept
  - Allocates functionality and defines interfaces and abstractions
  - The architect is not a generalist, but a specialist in simplifying complexity, resolving ambiguity and focusing creativity
- This is <u>The Job</u> of the architect
- Does it by thinking holistically about all other attributes of good product

Systems Architecture - Summary

- Architecture requires consideration of form and function, related through concept
- Starting with the operand, its transformation identifies concepts which deliver value
- 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

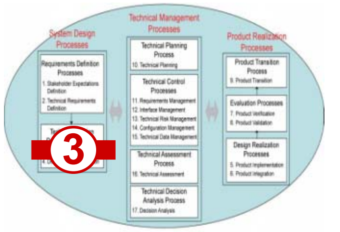
NASA SE Handbook:
Logical Decomposition Process





- **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."**

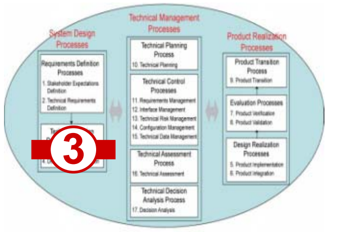
Role of Logical Decomposition



Provide detailed understanding of problem to be solved

Don’t leave any functions out!

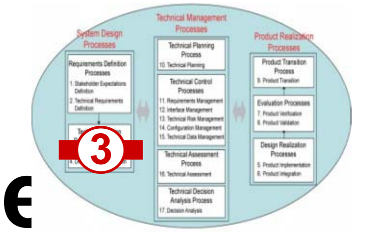
Logical Decomposition Purpose

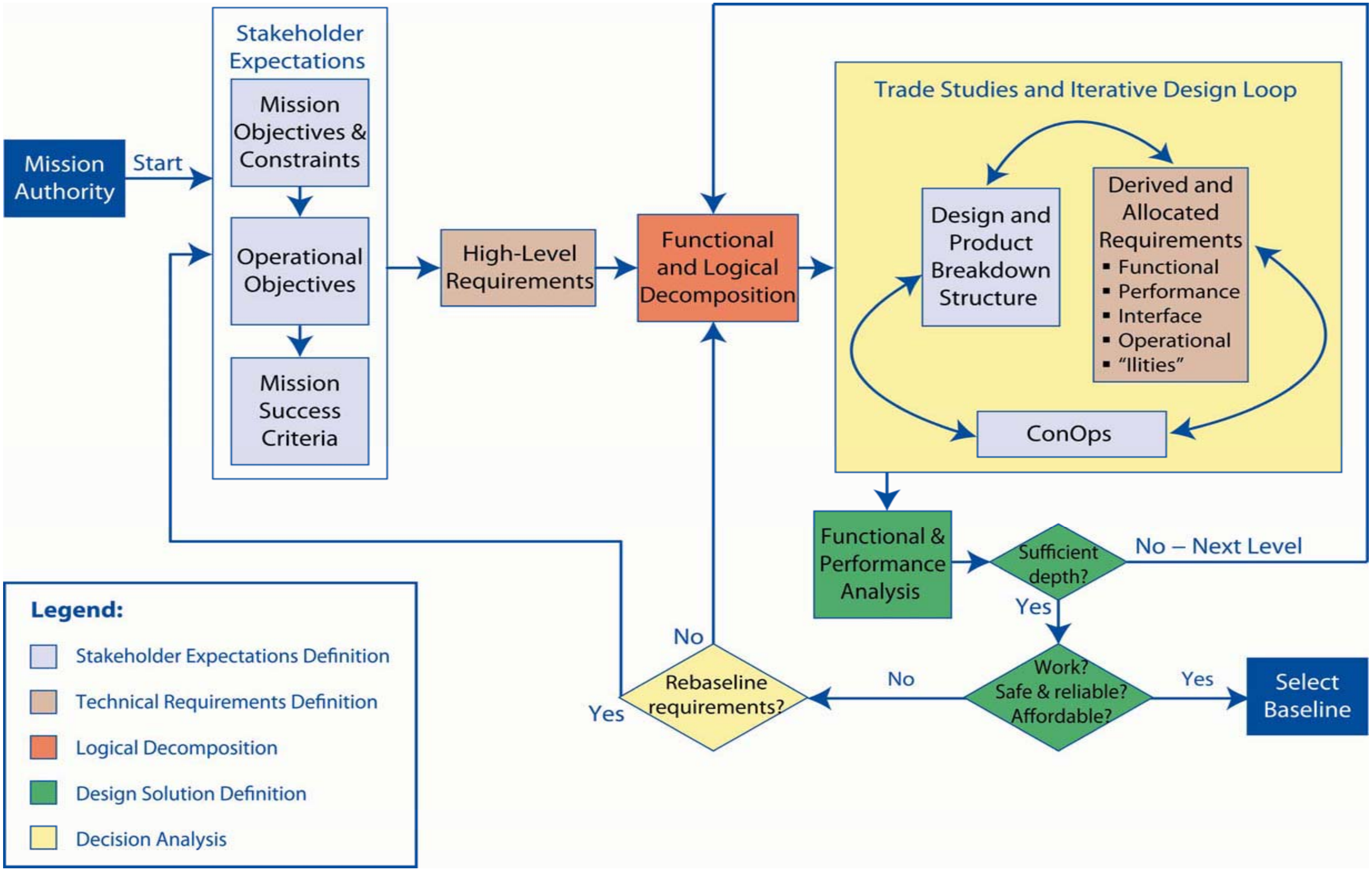


- The Logical Decomposition Process is used to:
  - Improve understanding of the defined technical requirements and the relationships among the requirements (e.g. functional, behavioral, and temporal)
  - Transform the defined set of technical requirements into a set of logical decomposition models and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

ARCHITECT THE SYSTEM

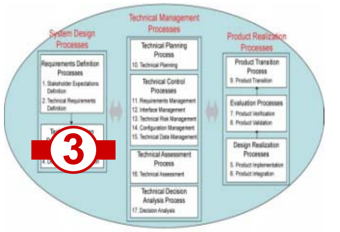
Interrelationships
Among the System Design Processe





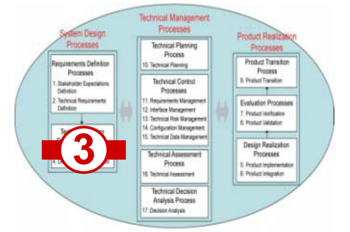
Source: NASA, SP-2007-6105, Figure 4.01

Logical Decomposition Importance



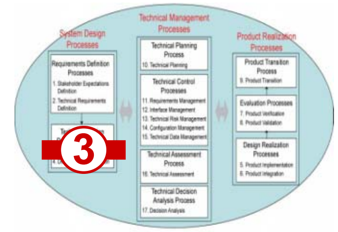
- It is the primary method used in system architecture development and functional requirement decomposition.
- It is the systematic process of identifying, describing, and relating the functions a system must perform to fulfill its goals and objectives.
- Three key steps in performing functional analysis are:
  - Translate top-level requirements into functions that must be performed to accomplish the requirements.
  - Decompose and allocate the functions to lower levels of the product breakdown structure.
  - Identify and describe functional and subsystem interfaces.
- It is the 1st step in getting the right design.

Logical Decomposition Process



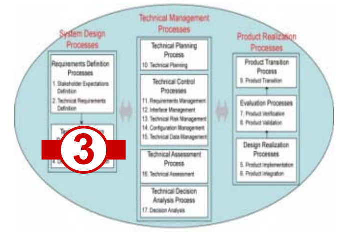
- The Logical Decomposition Process encompasses the formation of models, the allocation of Technical Requirements to them and using results of the analysis process the development of Derived Technical Requirements
- The design approach resulting from the Logical Decomposition Process:
  - Partitions a system into self-contained, logical groupings of elements to enable ease of change, achieve technology transparency and mitigate the risk of obsolescence
  - Uses rigorous and disciplined definitions of interfaces and, where appropriate, define the Key Interfaces within a system using widely supported, open system standards
    - USB

System Architecture Model Development



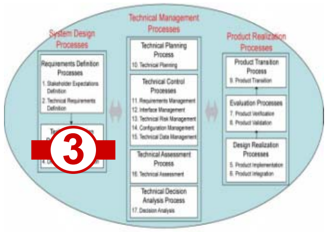
- The key first step in the Logical Decomposition Process is establishing one or more **system architecture models.**
  - The system architecture activity defines the underlying structure and relationships of hardware, software, communications, operations, etc.
  - Functional interfaces and relationships between partitioned subsystems and elements are defined as well
- The system designer uses functional analysis to begin to formulate a conceptual system architecture from the top-level (or parent) functional requirements and constraints
- The system architecture can be seen as the strategic organization of the functional elements of the system laid out to enable the roles, relationships, dependencies, and interfaces between elements to be clearly defined and understood

Decomposition Methods and Models

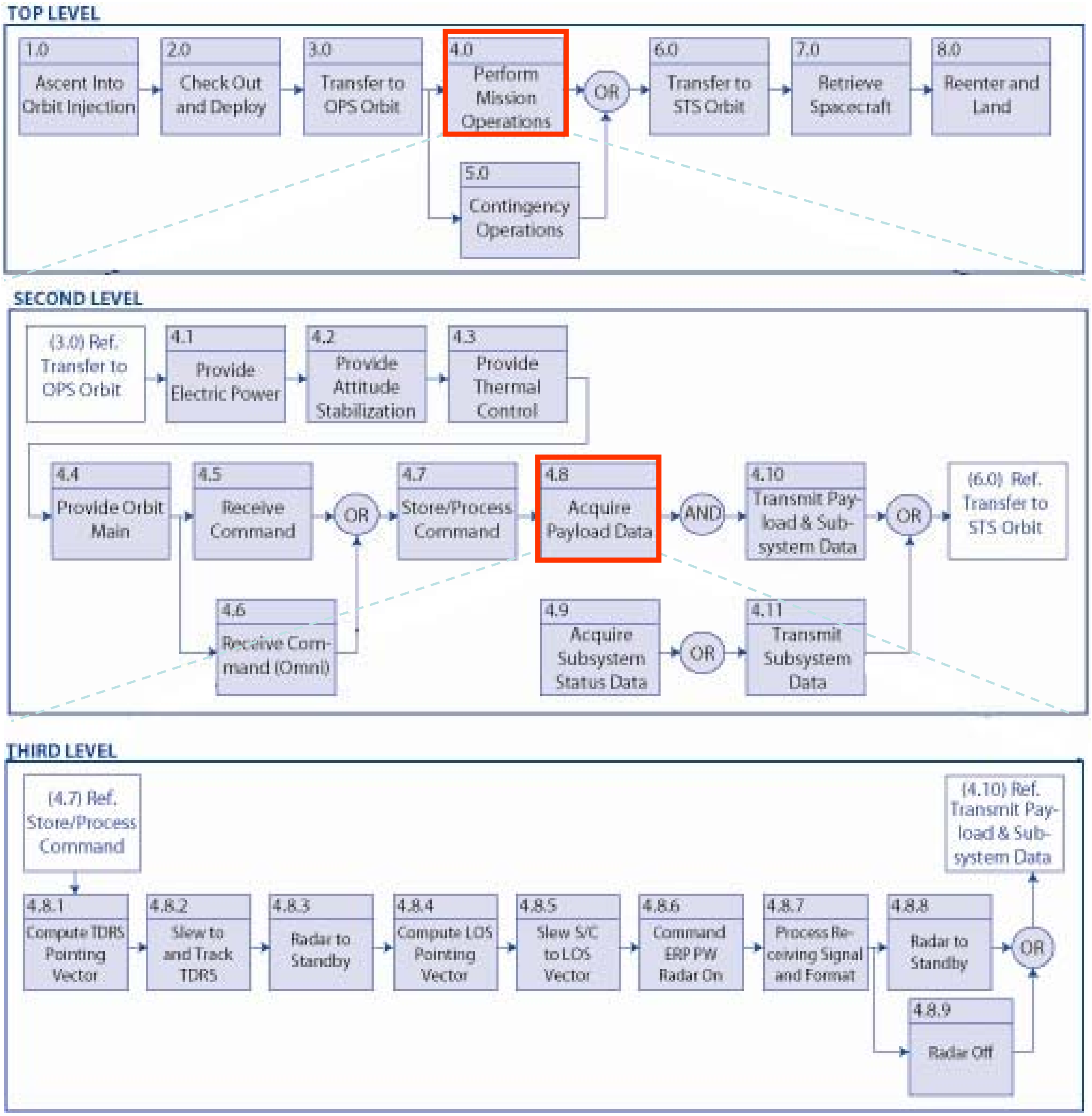


- The defined technical requirements can be decomposed and analyzed by:
  - Functions
  - Time
  - Behaviors
  - Data Flow
  - Objects
  - States and Modes
  - Failure Modes and Effects
- The models may include:
  - Functional Flow Block Diagrams
  - Timelines
  - Data Control Flow
  - Behavior Diagrams
  - Operator task sequencing
- Analysis of decompositions and requirement allocations is based on cost, schedule, safety and risk analyses

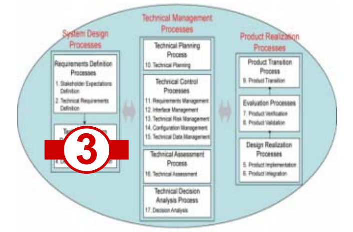
Functional Flow Block Diagram



Source: NASA/SP-2007-6105



Example of Decomposition Models



Timing Diagram

State Diagrams

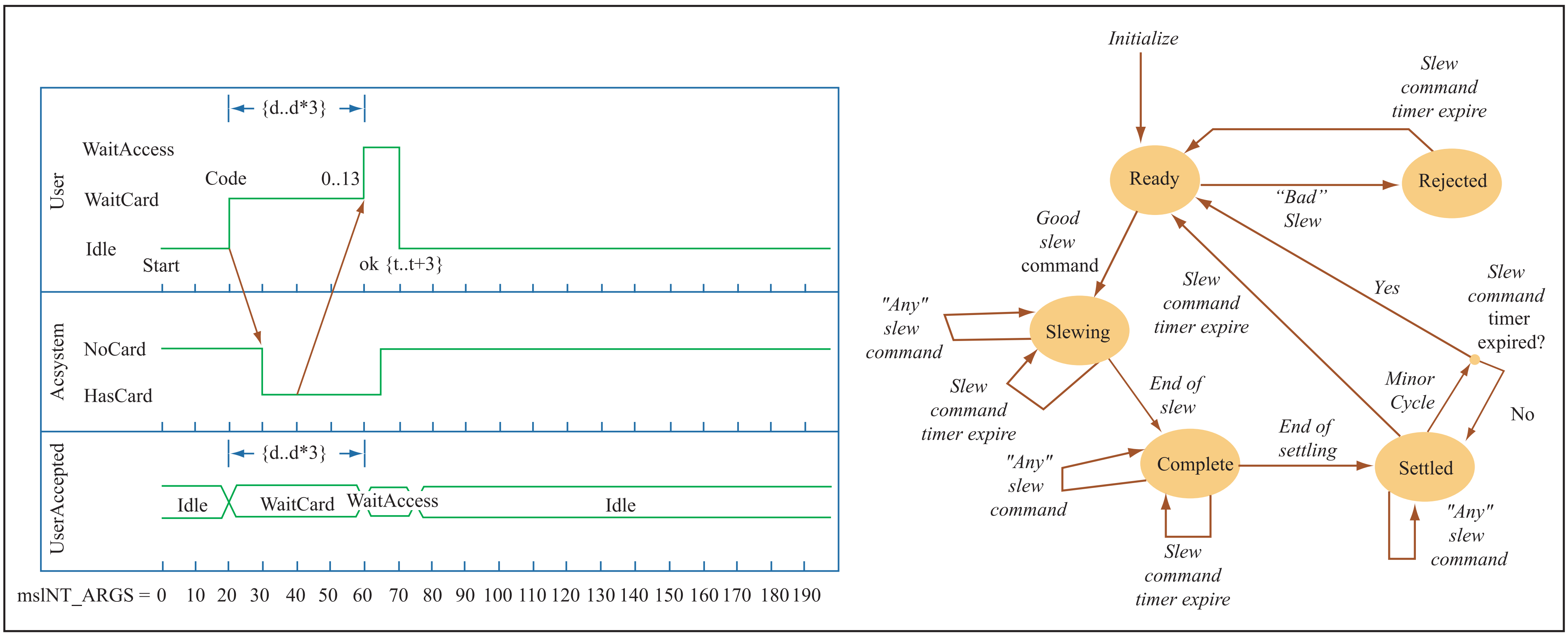
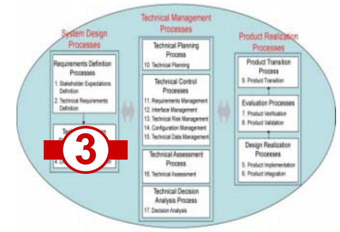


Image by MIT OpenCourseWare.

Example of Timeline Analysis



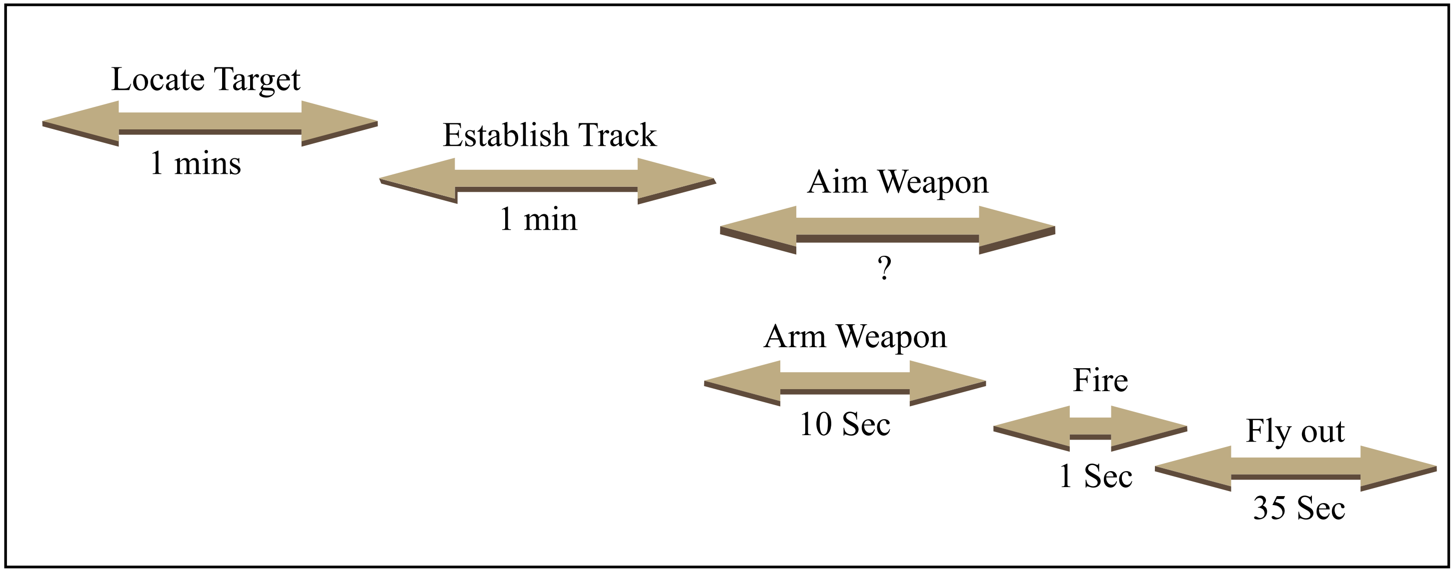
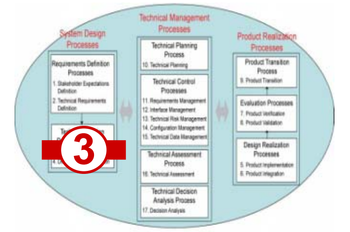


Image by MIT OpenCourseWare.

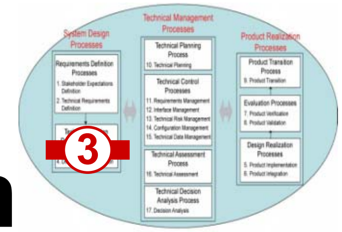
- The system shall destroy a target within 5 minutes of receipt of order.
  - The system shall locate the target within 2 minutes of receipt of order.
  - The system shall establish track within 1 minute of locating the target.
  - The system shall arm the weapon within 10 seconds of establishing track.
  - The system shall fire the weapon within 1 second of completing the aim of the weapon.
  - The weapon shall fly out to the target within 35 seconds of being fired.

Bi-Directional Traceability Analysis



- Use of traceability matrices are often used to ensure traceability throughout the Logical Decomposition Process
- Each sub-function should be checked to ensure traceability back to a technical requirement and that each requirement is implemented through at least one function
  - If there a function with no linkage to a requirement, then the designer has added a function that the user has not requested
  - If there are requirements with no linkage to a function, then the designers have not implemented all the requirements and the system may not meet those requirements during testing

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

1. Define One or More Logical Decomposition Models
2. Allocate Technical Requirements to Logical Decomposition Models to Form a Set of Derived Technical Requirements
3. Resolve Derived Technical Requirement Conflicts
4. Validate the Resulting Set of Derived Technical Requirements
5. Establish the Derived Technical Requirements Baseline

Output

To Design Solution and Requirements and Interface Management Processes

- Derived Technical Requirements

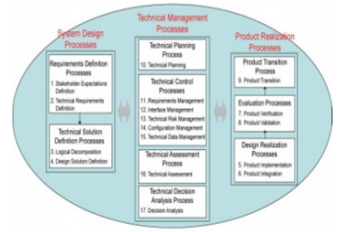
To Design Solution and Configuration Management Processes

- Logical Decomposition Models

To Technical Data Management Processes

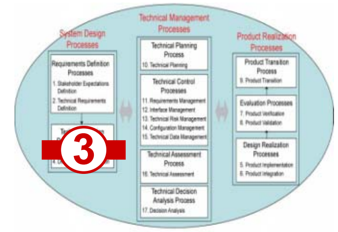
- Logical Decomposition Work Products

Benefits of the Logical Decomposition Process



- During the logical decomposition process, conflicts can be identified and resolved
- The logical decomposition methods can help understand the interaction between requirements
- Helps to establish a set of risk, cost, schedule, and performance criteria in planning trade-off analysis for conflict resolution
- Ensures that all the requirements are allocated to one or more functions

Logical Decomposition Summary



- The Logical Decomposition Process transforms the defined system to lower level functions and requirements
- Logical Decomposition Process begins by establishing one or more system architecture models
- Functional analysis is used to perform the logical decomposition of the system architecture model or models
- Logical Decomposition Process is recursive and iterative and continues until all desired lower levels of the system have been defined

16.842 Fundamentals of Systems Engineering
Fall 2009