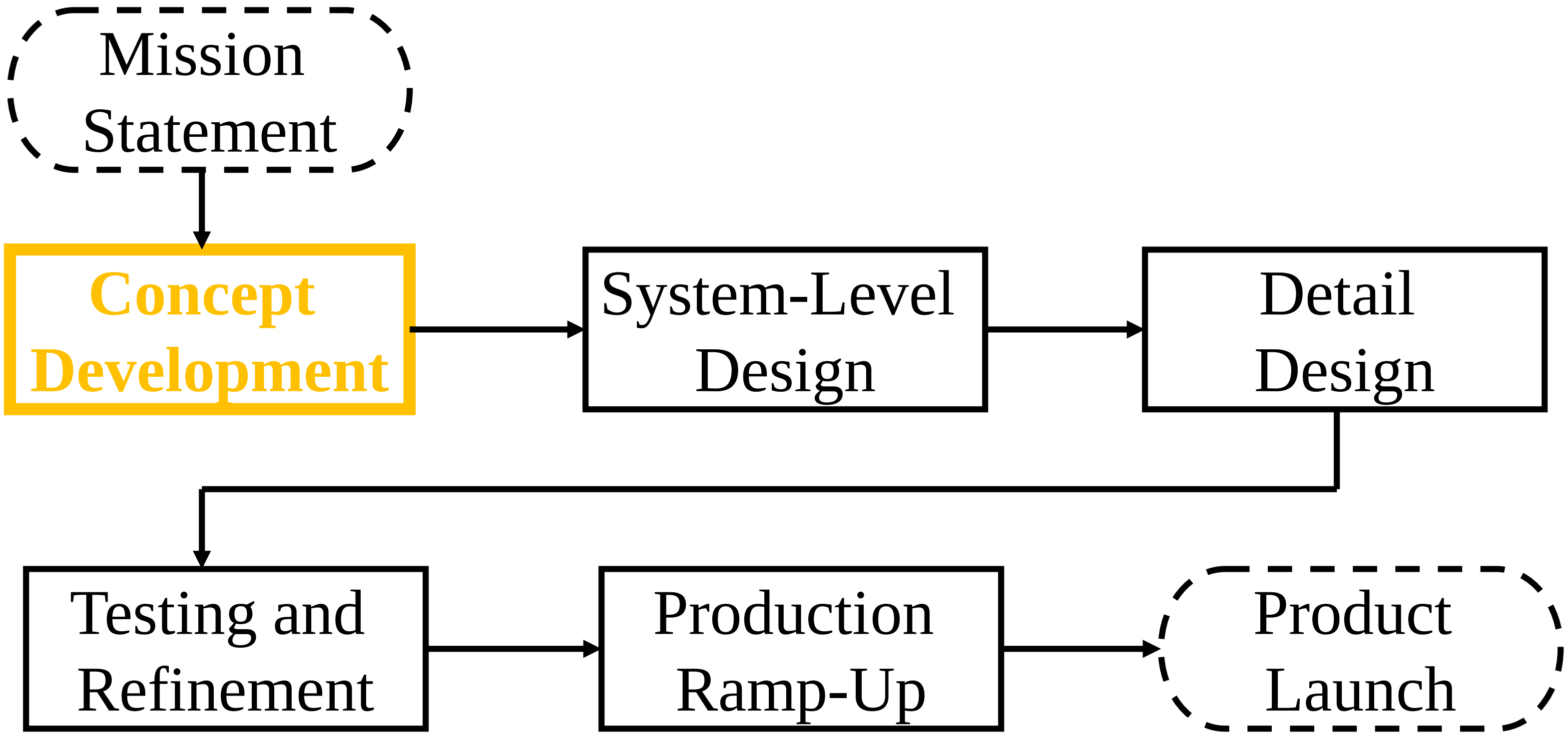
Elec 4309 Senior Design

Wendell H Chun

Oct. 10, 2017

Yesterday: Concept Selection



A Generic Product Development Process

Architecture

- The word “architecture” is derived from the Greek word “architecton”, which means master mason or master builder
- Webster’s Dictionary defines architecture as:
  - The art or science of designing or building structures
- The structure (in terms of components, connections, and constraints) of a product, process, or element – The Art of Systems Architecting
- An Architecture is the highest-level concept of a system in its environment - IEEE

Architecture

- Architecture – The fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution - P141 Standard
- Systems Architecture – The fundamental and unifying system structure defined in terms of system elements, interfaces, processes, constraints, and behaviors – INCOSE SAWG
- Architecture – The organizational structure of a system of CSCIs, identifying its components, their interfaces and a concept of execution among them

Architecture

- The architecture of a system defines its high-level structure, exposing its gross organization as a collection of interacting components.
- Components needed to model a architecture include:
  - **Components, Connectors, Systems, Properties and Styles.**

What Is an Architecture?

- It is the fundamental and unifying system structure defined in terms of system elements, interfaces, processes, constraints, and behaviors.
  - Source: International Council on Systems Engineering (INCOSE) System Architecture Working Group
- It is the structure of components, their relationships, and the principles and guidelines governing their design and evolution over time.
  - Source: Department of Defense (DOD) Architecture Framework v1.0
- A system architecture is the link between needs analysis, project scoping and functional analysis and the first descriptions of the system structure.

What is System Architecture?

INCOSE Systems Engineering Handbook (SEH) 3.2

- [4.3.2.1 Architectural Design Concepts]:
  - "System … architectures depict the summation of a system's entities and capabilities at levels of abstraction that support all stages of deployment, operations, and support." – C. Wasson
  - "the selection of the types of system elements, their characteristics, and their arrangement"
- [4.3.1.5 Architectural Design Process Activities]:
  - "logical sequencing and interaction of system functions or logical elements"
  - "Identify interfaces and interactions between system elements"

System Architect

- The architect is a member of the team that is responsible for designing and building a system
- The architect's contribution comes in the very early stages of the systems engineering process:
  - When the operational concept is defined
  - The basic structure of the system is conceptualized
- A system architect, not only knows about the individual components, but also understands the interrelationships among the components

Systems Architecting

- Systems Architecting has been defined as the process of creating complex, unprecedented systems
- Building systems in today's world is tenuous at best:
  - Requirements of the marketplace are ill-defined
  - Rapidly evolving technology provides new services at a global level instantly
  - Uncertainty is increasing about the way the system will be used, the components that will be incorporated and the interconnections that will be made

Systems Architecting

- Generating a system architecture as part of the systems engineering process can be seen as a deliberate approach to deal with the uncertainty that characterizes these complex, unprecedented systems

Developing A System Architecture

Creating an architecture is the beginning of the system design process and establishes the link between requirements and design. The typical architecture development sequence is:

1. Establish initial system requirements by needs analysis, project scoping, and the development of the concept of operations (CONOPS).
2. Define the external boundaries, constraints, scope, context, environment and assumptions.
3. Develop candidate system architectures as part of an iterative process using these initial requirements.
4. For each architecture, compare the benefits, costs, risks and the requirements that drive their salient features and consider modifying (with stakeholder involvement) their CONOPS, system performance and even their system functions to improve the solution-problem proposition.

Traditional Approach to Systems Architecting

- Many methodologies have been developed to support a traditional system development model:
  - Define the requirements
  - Consider several options
  - Emerge with a well-defined design through a process of elimination
  - Based on structured analysis and design

Developing Candidate System Architectures is Recursive and Iterative

Needs Analysis

- ✓ What needs are we trying to fill?
- ✓ How are current solutions insufficient?
- ✓ Are the needs completely described?

CONOPS

- ✓ Who are the intended users?
- ✓ How will the system be used?
- ✓ How is this use different from heritage systems?

Functional Requirements

- ✓ What capabilities are required?
- ✓ At what level of performance?
- ✓ Are segment interfaces well defined?

System Architectures

- ✓ What is the overall approach?
- ✓ What elements make up this approach?
- ✓ Are these elements complete, logical, and consistent?

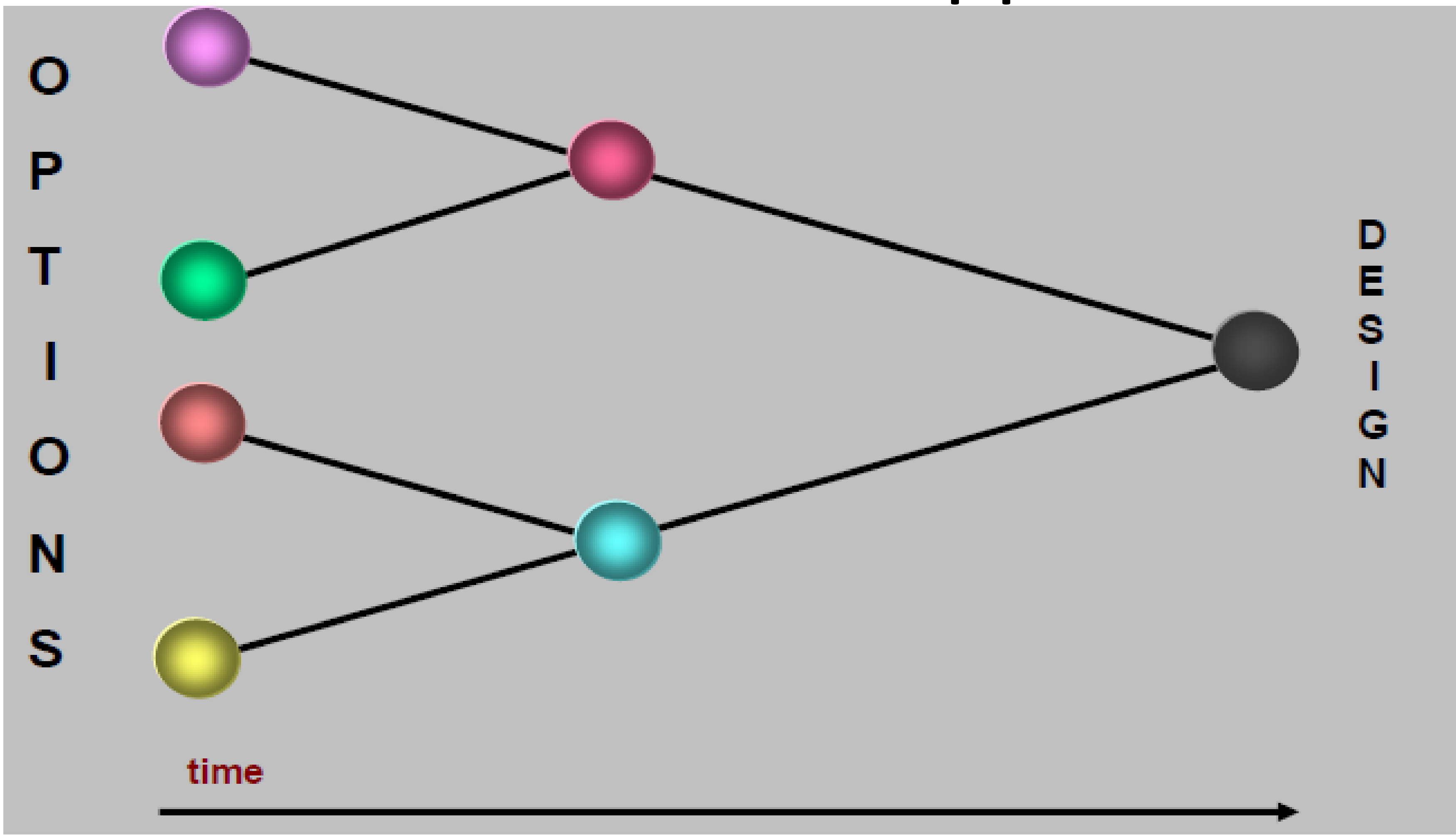
Work With Customer to Potentially Modify Problem Statement Based on Solution Options

Work With Customer to Potentially Modify Problem Statement Based on Solution Options

Traditional Approach to Systems Architecting

- Effective when the requirements are well defined and remain essentially constant during the system development period
  - Cannot handle change well:
    - If the implementation of the system is long – on the order of years – the requirements change because of changing needs and new technology offers different alternatives and opportunities

The Traditional Approach



Last Word is “Design”

Design as verb is nearly the same as architect

- But more comfortable

Design as noun is nearly the same as architecture

- But not “first, having authority”

But “Architect” is way cooler than “Designer”

Safe to use **design** when talking about *most* System Architecture activities and products:

- Presentation title: “From Concept to **Design**”
- SEH 4.3: “Architectural **Design** Process”

Why?

"… *to synthesize a solution that satisfies system requirements.*"

- [SEH 3.2: 4.3.1.1 Purpose]

"… *developing the system architecture is one of the most important responsibilities of the systems engineer.*"

- [SEH 3.2: 4.3.2.1 Architectural Design Concepts]:

How to Solve Difficult Problems

To ensure solution meets requirements

- Requirements and Scenario Tracing

To break up into more manageable pieces

- Functional and Physical Decomposition

To know what to measure during implementation

- Technical Performance Measures (TPM)

To understand COTS tradeoffs

- Requirements and Function Allocation & Element Interfaces

To help avoid mistakes and prepare for V&V

- Valves in Space! Stairway to Nowhere! Units Mismatch!

Documenting System Architecture

Collection of **Artifacts** that communicate the system's structure, behavior, and qualities

Primary artifact type is the **Model**

- Graphical: IDEF0, SysML, AADL, N2, FFBD, etc.
- Math & Stats: equations of motion, Monte Carlo, etc.
- Physical: wind tunnel model, usability mockups, etc.

Artifacts are organized by Architectural **Views**

Architectural Views

One Architecture, Many Views

- No more “Functional Architecture”, “Physical Arch.”, etc.
- A view emphasizes artifacts that communicate a certain aspect of the system

By Perspective and Semantics:

- Functions, Inputs, Outputs, … (Functional)
- Physical Equipment, Cables, Topology, … (Physical)
- Users, Workflow, External Systems, … (Operational)

By Scope or Scale:

- Environment, System, Element, …

By Life Cycle:

- Development, Operations, Training, Logistics, Refinement, Retirement, …

Notation Syntax and Semantics

Diagrams:

- Represent elements and relationships between them
- Follow Notation's Syntax and Semantics rules

Syntax rule examples:

- All lines must connect at both ends
- All boxes must be connected by at least one line

Semantics rule example:

- Each box must be connected to at least one input, one output, and one mechanism

Informal Lines and Boxes

Plain Ol' Lines And Boxes (POLAB)

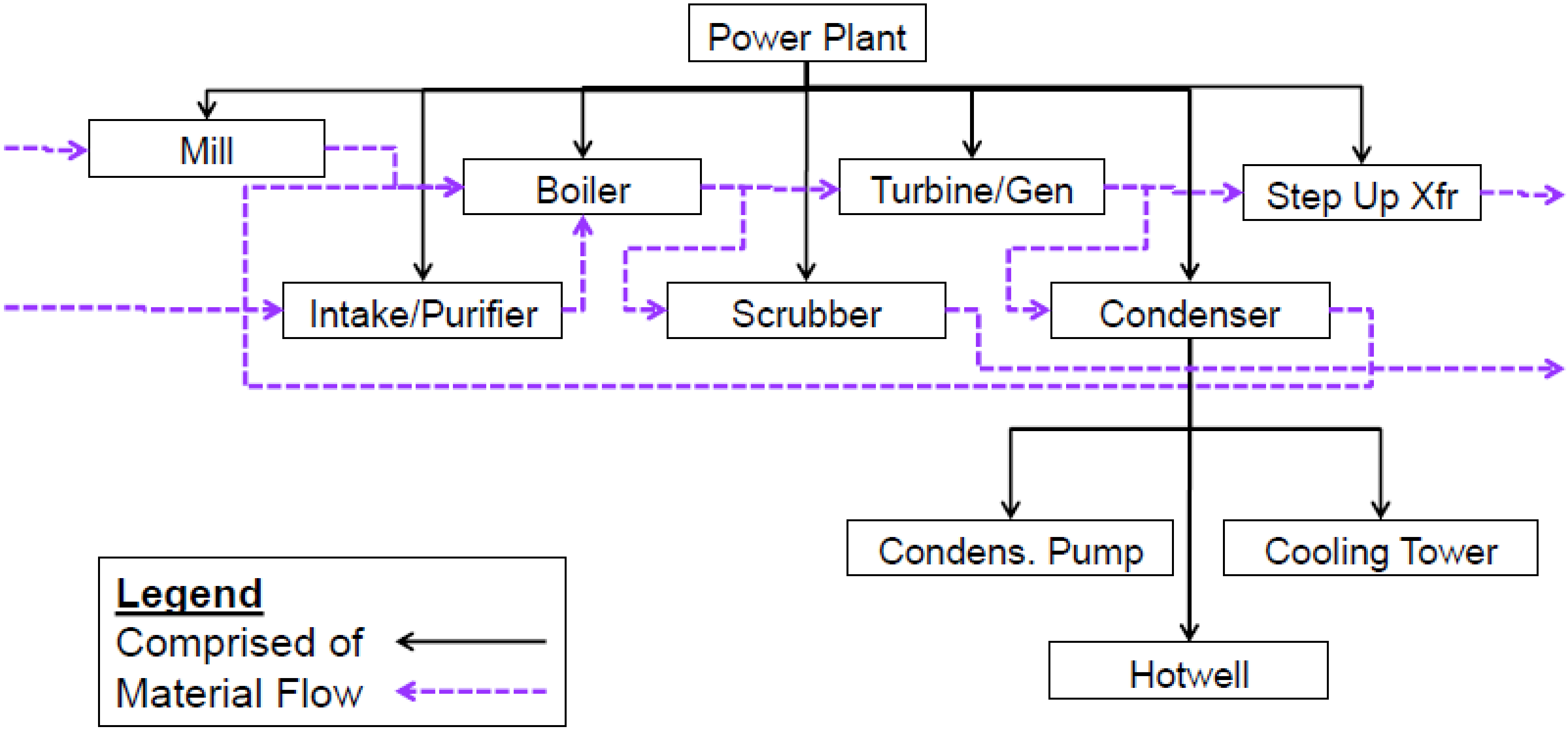
- Informal notation
- "Back of envelope"
- "Whiteboard prototyping"

Semantics:

- Anything you want
- But make them very clear, preferably with a Legend so you don't forget

POLAB Example

Physical decomposition of Power Generating Plant



We Know What we Want, Now What?

We have our System Concept with:

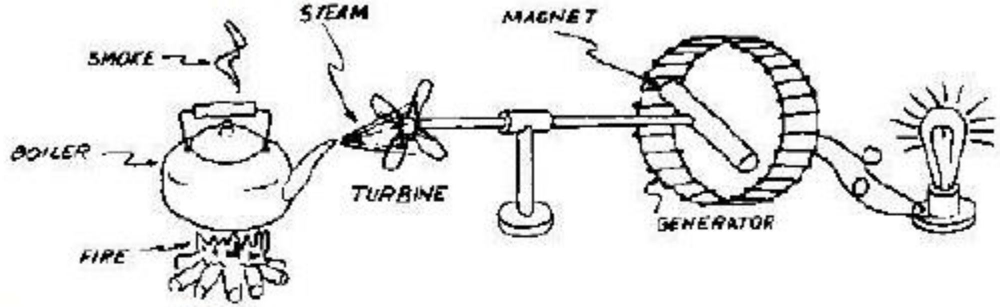
- Usage Scenarios
- System Requirements
- Performance Objectives

Can't we just start building now?

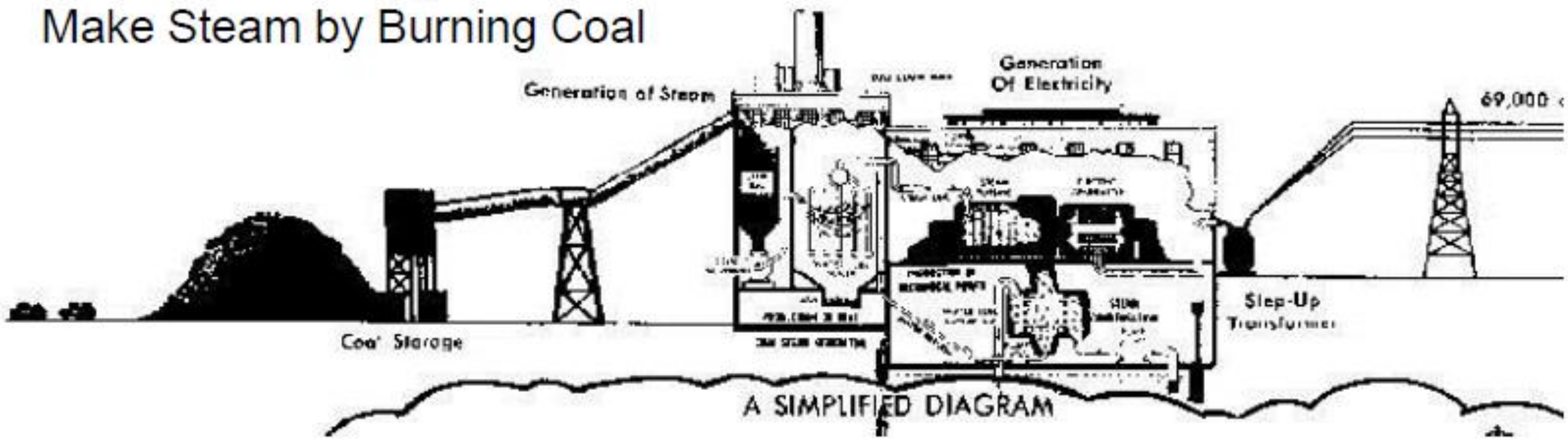
We need a way to turn **Concept** into **Design**

Example Concept

Generate Electricity by Steam-Powered Turbine



Make Steam by Burning Coal



- *Welcome to The New Castle Power Plant*, Pennsylvania Power Co., late 1960's

Example Usage Scenario #1

1. River water is strained and filtered
2. Water is purified
3. Pure water is boiled
4. Steam turns the turbine
5. Spent steam is cooled into condensate
6. Condensate is reused by pumping back to boiler

Architectural Design Process [SEH3.2:4.3]

Define the Architecture

- Logical sequencing/interaction of functions or logical elements
- Partition and allocate system requirements with associated performance requirements
- Identify internal and external interfaces and interactions
- Define V&V Criteria for the system elements

Analyze and Evaluate the Architecture

- Evaluate alternative design solutions

Document and Maintain the Architecture

- Establish traceability between requirements and system elements

Approach

- Start with overall System Function, then Context
- Top-down *and* Bottom-up Functional Analysis
- Allocate requirements to functions
- Keep thinking, "how will we test this?"
- Function to Physical Allocation
- Physical Analysis, COTS selection

Remember: It's all one big exercise in trade-offs, so generate and evaluate as many alternatives as feasible

[SEH 3.2: 4.2] implies use of an **iterative** approach, example is Specification Tree [3.4.2].

Insights gained during architecture development and evaluation may lead to **changes to requirements** and functional interfaces

Functions

Function is a process that transforms **inputs** into **outputs**

Describes an **action** taken by the system or by one of its elements

A function is known by a **verb** or **verb-noun pair**

A function can be decomposed or composed

Data, information, energy or material exchanged between functions in order to accomplish action

Functional Analysis

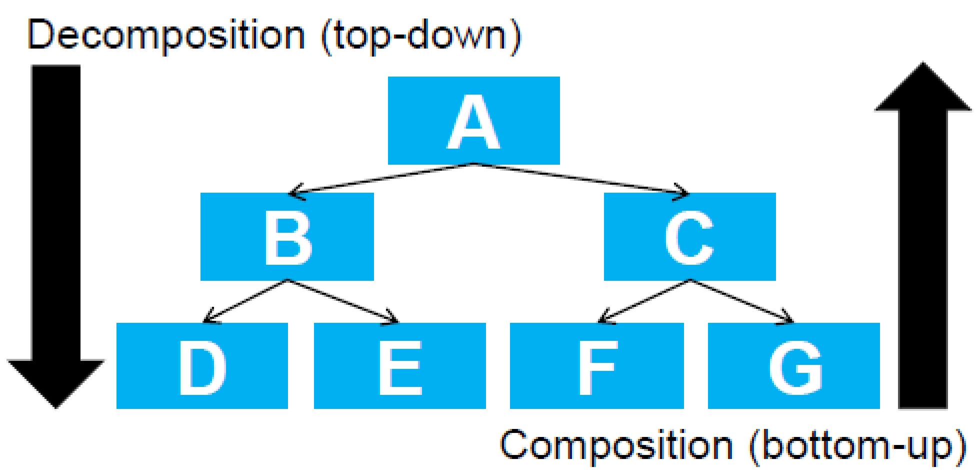
Decomposition permits innovation,

- Careful of Pirsig knife

Composition/Synthesis exploits existing systems, COTS and experience

- Careful of Christiansen org chart trap

Use both to create **best** solution!



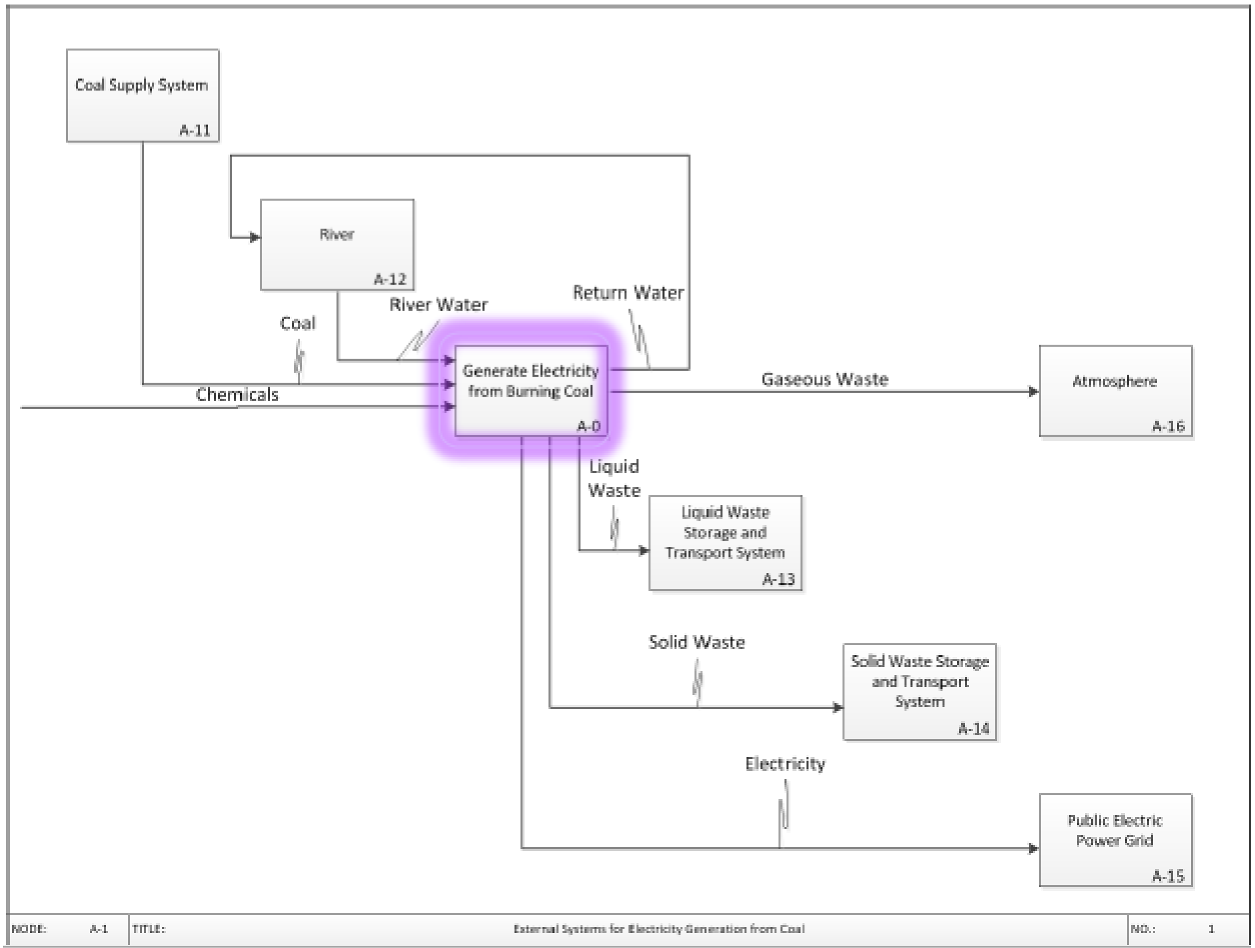
Functional View of System

Entire system's functionality summed up in one box

Law of Conservation of Inputs, Controls and Outputs

- every input, control, and output of a particular function must appear on the decomposition of that function, and there can be no new ones.

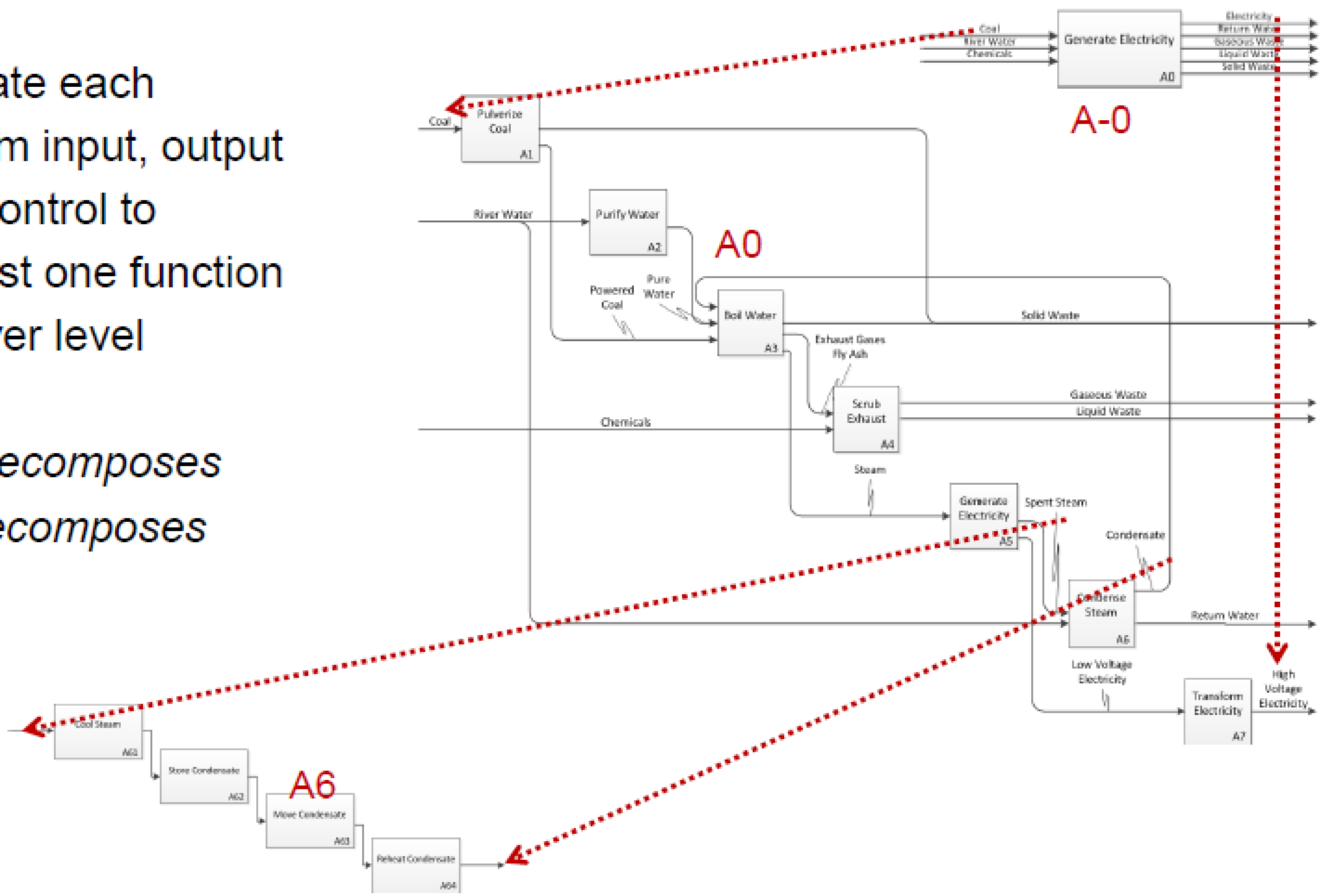
Context of External Systems



Functional Decomposition (Top-down)

Allocate each system input, output and control to at least one function in lower level

"A6 decomposes A0 decomposes A-0"



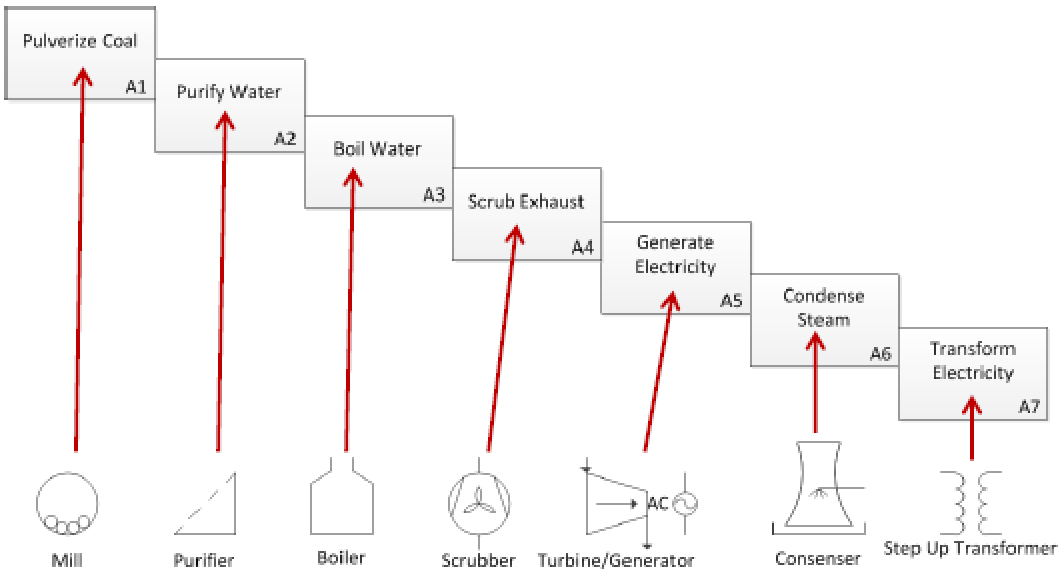
Functional Composition (Bottom-up)

4. Any ICOs “left over” form this level’s ICOs

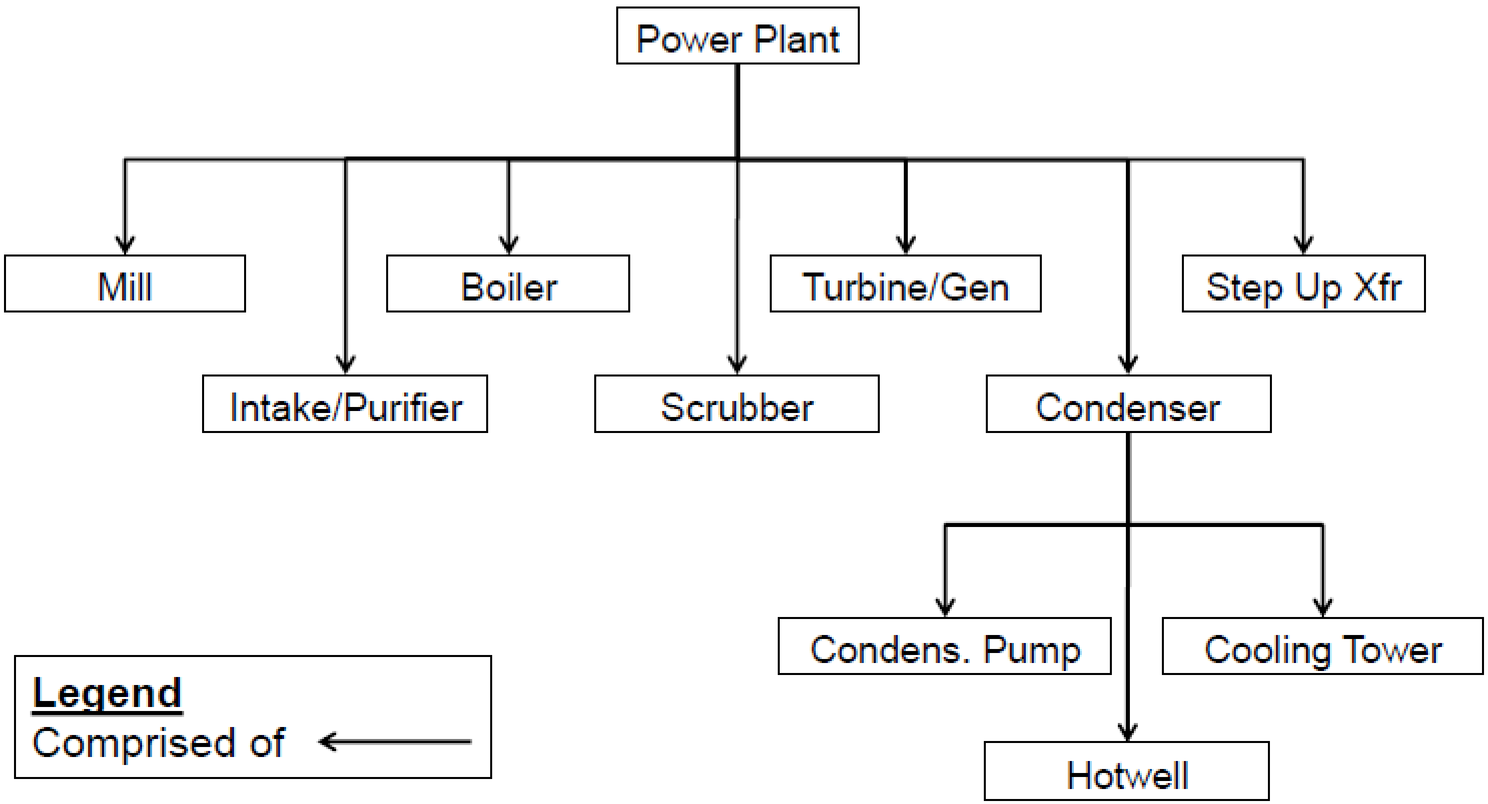
3. “Compose” inputs, controls, outputs (ICOs) for each existing element

2. Create functions from them

1. Start with existing elements



Physical Decomposition



Function to Physical Allocation

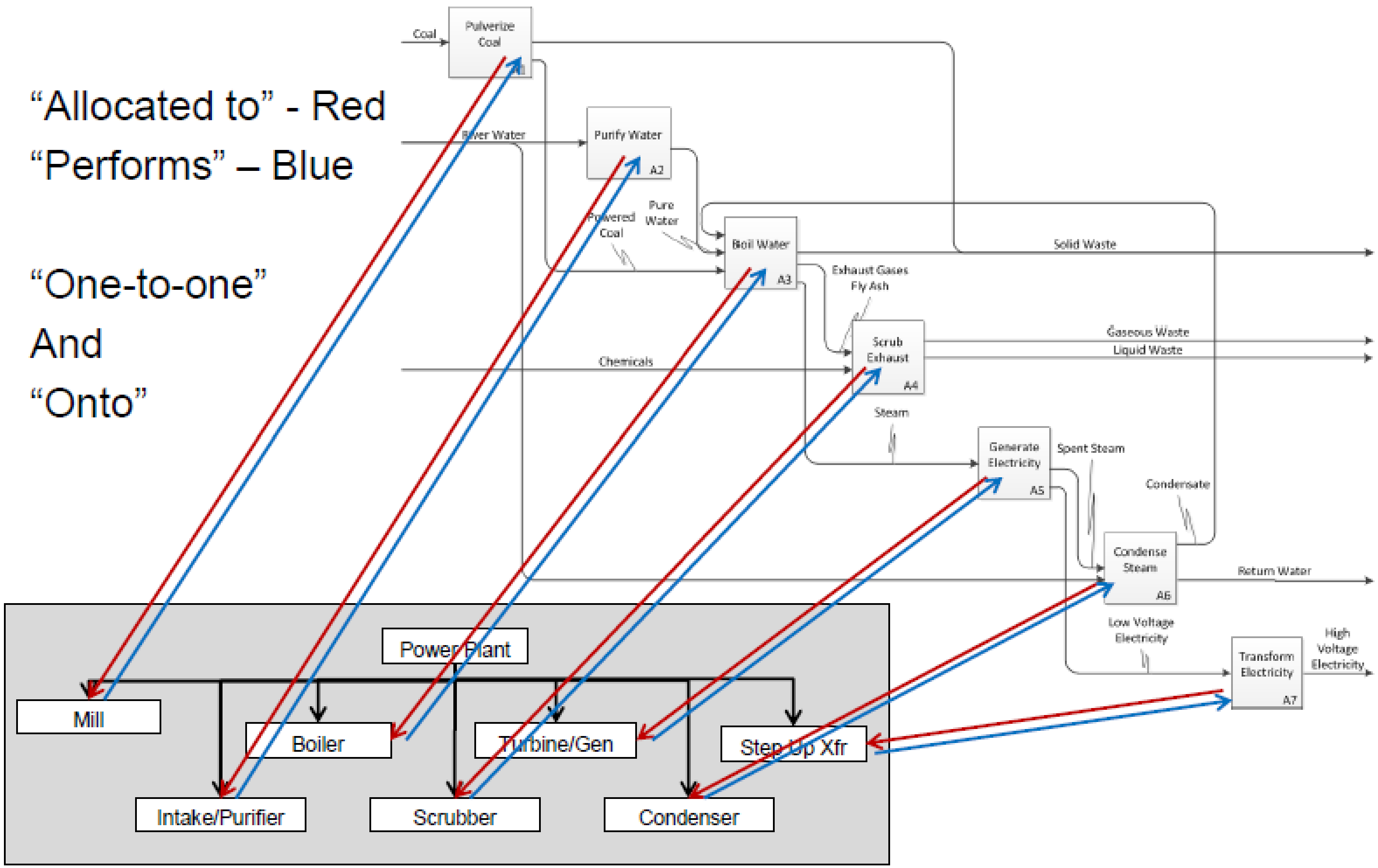
"Allocated to" - Red

"Performs" – Blue

"One-to-one"

And

"Onto"



So How Do We Create Architectures?

There are two primary techniques to create architectures, both benefit from understanding the *performance and limitations of heritage systems*.

- Synthesis:
  - Modifying or combining existing systems to satisfy stated needs
  - Requires logic and good knowledge of existing systems
  - What functions do I need to get the job done?
  - Can I combine existing systems without too much baggage?
- Discovery:
  - Leverage knowledge of existing architectures to 'discover' a new one
  - Requires knowledge of existing systems and abstraction skills
  - Is there an analogous system in another domain?
  - What are the good or bad properties of a given architecture?

More Tools and Techniques

Patterns

Creativity Techniques

- Morphological Analysis
- Living Systems Theory
- Hatley-Pirbhai Modeling

Model-based SE tools: CORE, Rhapsody, etc.

Architecture and Design Patterns

"Constitute high-level structures appropriate to the design of the major elements of the system of interest.

Express the relation between the context, a problem, and a solution, documenting attributes and usage guidance.

Architecture patterns are time-proven in solving problems which are similar in nature to the problem under consideration."

- Cloutier, Robert (2009). Patterns and Systems Architecture, http://www.incose.org/orlando/Attach/200902/Cloutier%20Patterns%20Orlando%20INCOSE%20-%20no%20backup.pdf
- Alexander, Christopher (1977). A Pattern Language: Towns, Buildings, Construction
- Gamma, E., Helm R., Johnson R., Vlissides J. (1995). Design Patterns: Elements of Reusable Object-Oriented Software
- Buschmann F., Meunier R., Rohnert H. & Sommerlad P. & Stal M. (1996). Pattern-Oriented Software Architecture: A System of Patterns
- http://www.patterns4se.com/

Morphological Box

Itemize generic aspects of system or element

List alternate choices for each generic aspect

Kind of like an *a la carte* menu

Related to [SEH 3.2: 4.3.2.3] **option descriptors**

Example for “Condense Steam” decomposition:

Aspects

Reservoir Volume	Flow Rate	Cooling Technique
500 gal	50 gpm	Traveling Water Screen
700 gal	90 gpm	Cooling Tower
	200 gpm	

Choices

- Buede, D. (2000). *The Engineering Design of Systems: Models and Methods.*
- http://en.wikipedia.org/wiki/Morphological_analysis_(problem-solving)

Living Systems Theory

Processors of matter-energy and information:

- Reproducer, Boundary

Processors of matter or energy:

- Ingestor, Distributor, Converter, Producer, Storage, Extruder, Motor, Supporter

Processors of signals and information:

- Input transducer, Internal transducer, Channel and Net, Timer, Decoder, Associator, Memory, Decider, Encoder, Output transducer

http://en.wikipedia.org/wiki/Living_systems_theory

Selection Architecture Selection Criteria

1. Ability to fulfill mission as defined by requirements
2. Ability to operate within resource constraints
3. Accommodation of interfaces
4. Ability to adapt to future needs
5. Costs over entire life cycle
6. Side effects
7. Measures of risk
8. Measures of quality factors
9. Measures of subjective factors

[SEH 3.2: 4.3.2.2]

Evaluation Techniques

Scenario Tracing

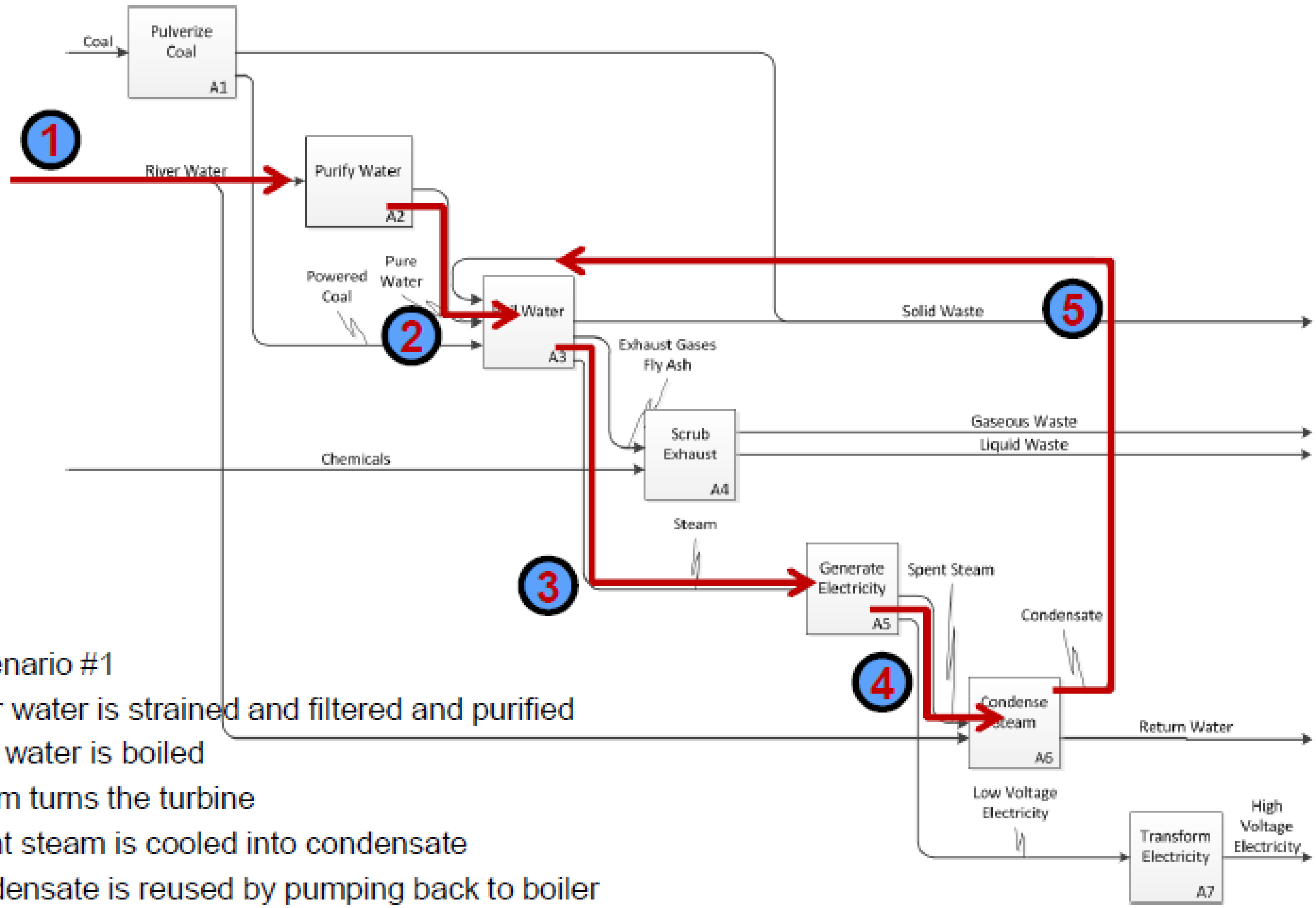
Requirements Tracing

Function Evaluation

Other Techniques

- Modeling and Simulation
- Trade Studies
- Architecture Tradeoff Analysis Method

Scenario Tracing



Usage Scenario #1

1. River water is strained and filtered and purified
2. Pure water is boiled
3. Steam turns the turbine
4. Spent steam is cooled into condensate
5. Condensate is reused by pumping back to boiler

Function Evaluation

Use Scenario Tracing to find:

Shortfalls – absence of functionality

- Absence of functionalities for input set
- Inability to produce desired output
- Insufficient feedback/control to produce desired output
- Trace stops at dead-end

Overlaps – redundancy in functionality

- May be more difficult to find than Shortfalls

Requirements Tracing

	Behavioral Requirements															
	Input Reqs			Output Reqs					Exter	Functional Reqs						
Functions	1.1.1 Accepts coal	1.1.2 Accepts river water	1.1.3 Accepts chemicals	1.2.1 Provides h.v.electricity	1.2.2 Provides return water	1.2.3 Provides gaseous waste	1.2.4 Provides liquid waste	1.2.5 Provides solid waste	1.3.1 Shall use public electric grid	1.4.1 Pulverize coal	1.4.2 Purify river water	1.4.3 Create steam	1.4.4 Scrub exhaust	1.4.5 Generate electricity	1.4.6 Condense steam	1.4.7 Transfor
1. Pulverise Coal	x							x		x						
2. Purify Water		x	x								x					
3. Boil Water								x				x				
4. Scrub Exhaust			x			x	x						x			
5. Generate Electricity														x		
6. Condense Steam		x			x										x	
7. Transform Electricity				x					x							x

Using System Architecture

Technical Performance Measures (TPM)

- Taken from Objective Hierarchy and Decompositions
- Element weight, volume, heat dissipation, etc.
- Response times

Interface Definitions

- Taken from Decomposition boundaries
- Interface Control Documents
- Machine-readable Interface definitions

Qualification Artifacts

- Test plans

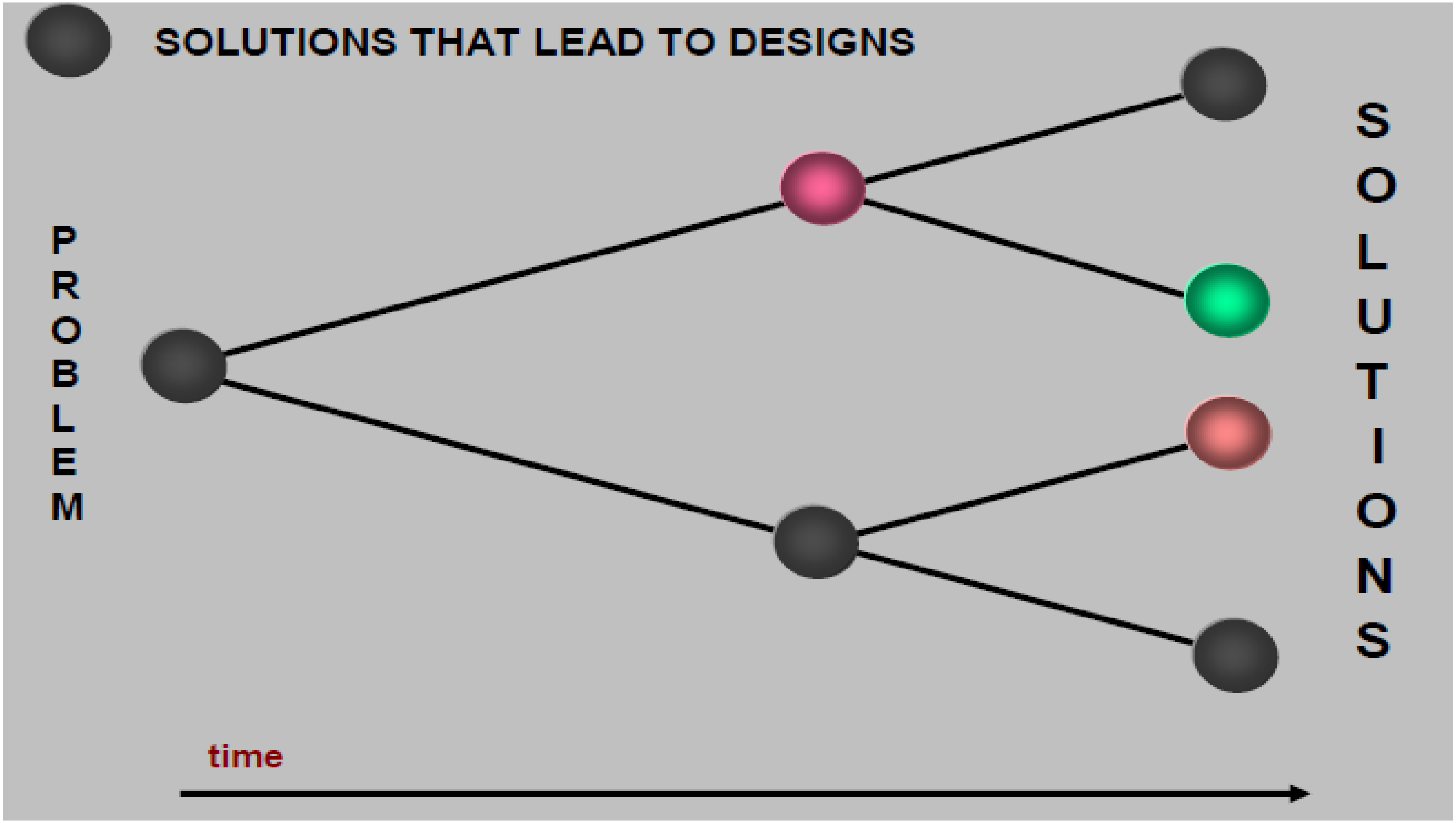
Evolutionary Approach

- New approach that is emerging with roots in systems engineering
- Deals with uncertainty in requirements and in technology, especially for systems with a long development time and expected long life cycle:
  - Evolutionary development
  - Build-a-little, Test-a-little
- Requirements are allowed to be more abstract and therefore subject to interpretation
- Alternative solutions are explored and pursued further as new technology options become available

Evolutionary Approach

- Intermediate designs are saved
- Some intermediate designs are implemented as prototypes but not operationally implemented while others are implemented in traditional ways
- Advantages of Object-Oriented approach:
  - Allows flexibility in the design as it evolves over time
- Disadvantages of Object-Oriented approach:
  - Requires some early elimination of technology alternatives in the absence of reliable information

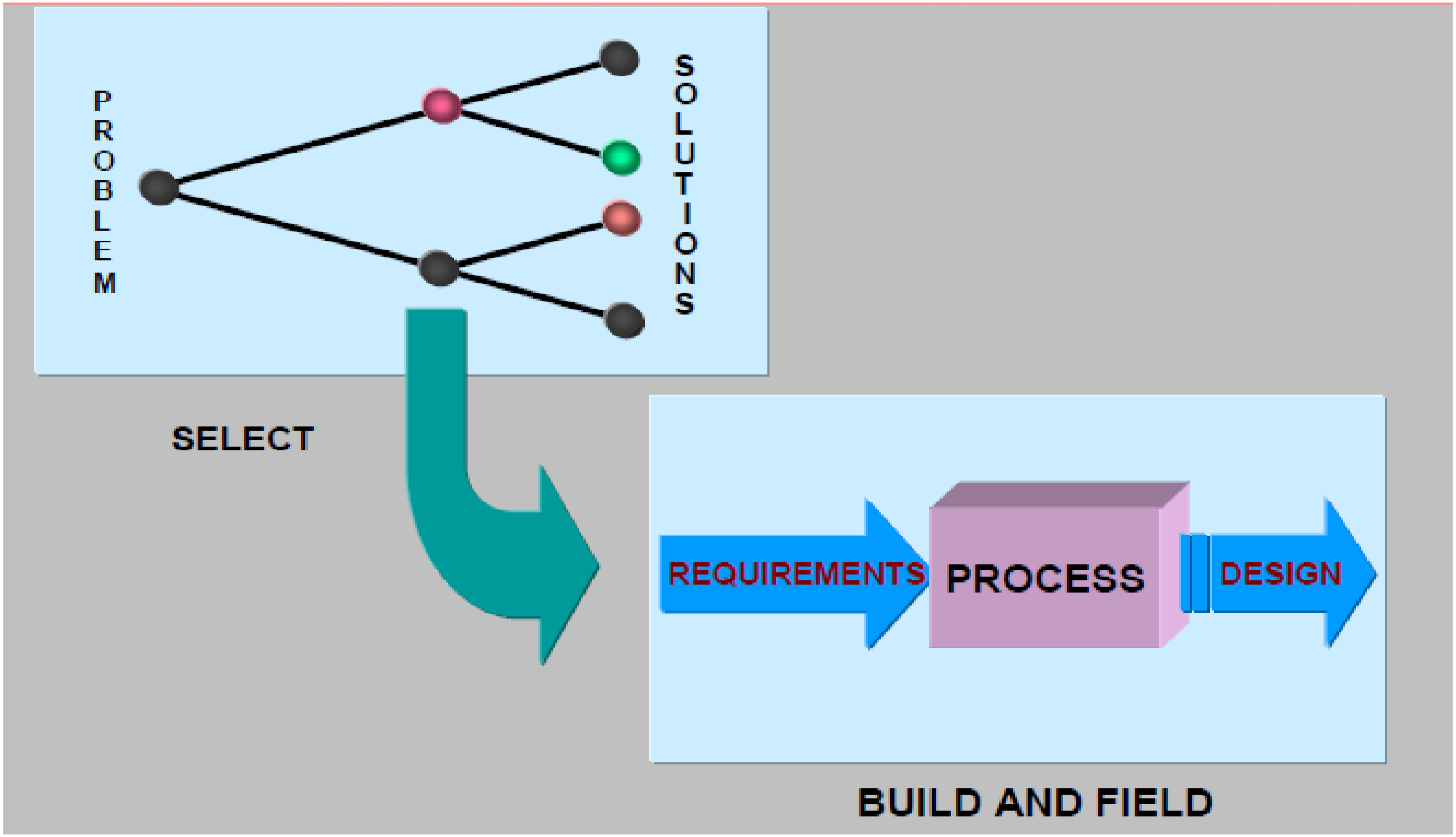
Evolutionary Approach



Select, Build, and Field

- At any time in the development process, when there is a need to build a system, the available solution that best meets the current requirements is selected and implemented using any systems engineering approach

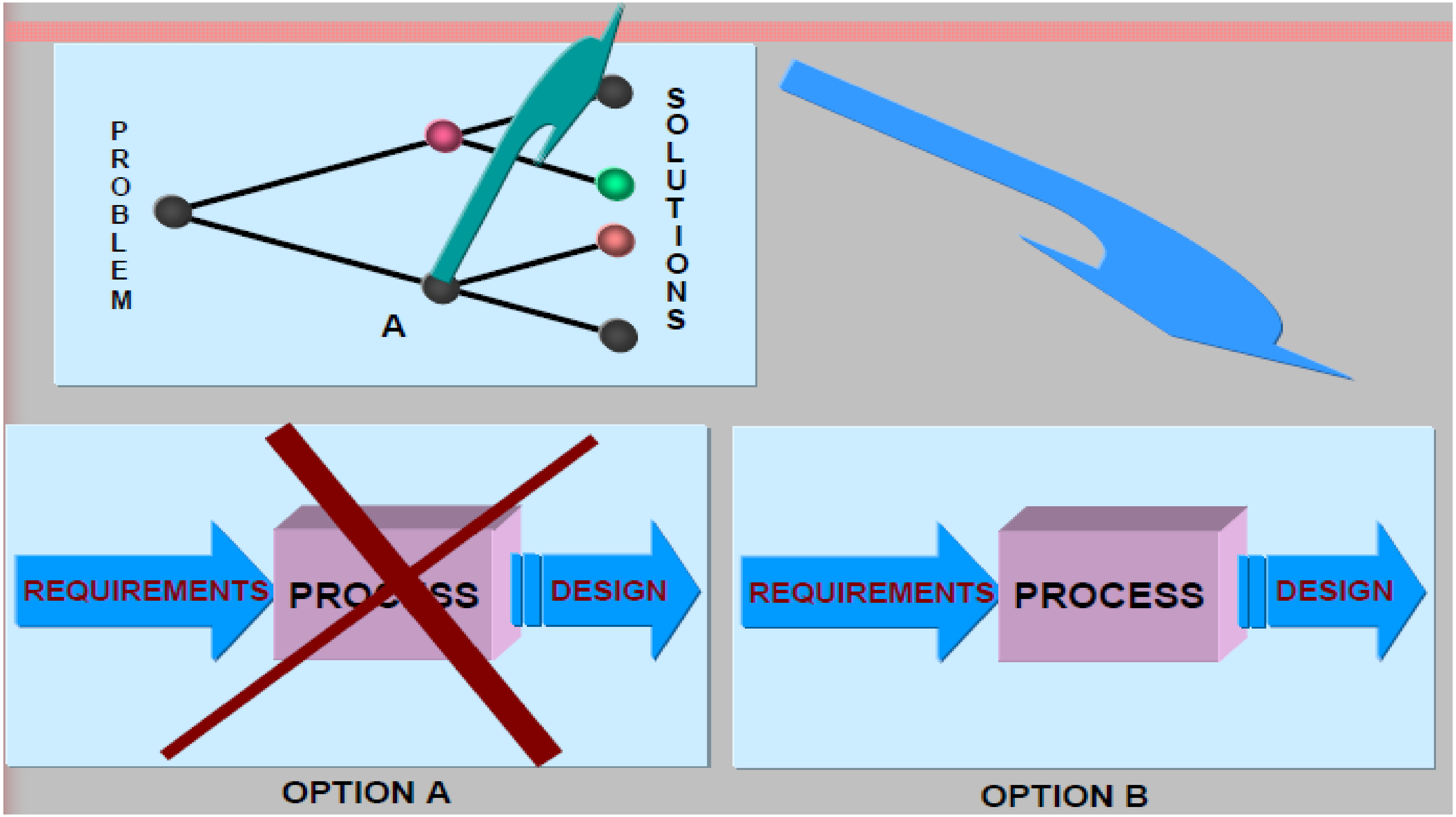
Select, Build, and Field



The Challenge of Coping with Change

- If the implementation is long, then the situation shown next prevails, with the unfortunate consequences that very little, if any, from the work on Option A is used for Option B

The Challenge of Coping with Change



How to Define an Architecture

- Defining an architecture, especially of an information system, requires the following items to be described:
  - Processes exist that need to take place in order that the system accomplish its intended functions
  - The individual processes transform either data or materials that “flow” between them
  - The processes or activities or operations follow rules that establish the conditions under which they occur
  - The components that will implement the design (hardware, software, personnel, and facilities must be described)

How to Define an Architecture

- Define the Functional Architecture
  - A **functional architecture** is:
    - A set of activities or functions that are arranged in a specific order and when activated, achieves a set of requirements
    - Divide and allocate the functional requirements into different sub-functions and modes of operation

How to Define an Architecture

- Define the Physical Architecture
  - A **physical architecture** is:
    - A representation of the physical resources
    - Expressed as nodes that constitute the system and their connectivity
    - Expressed in the form of links

How to Define an Architecture

- Define the **technical architecture:**
  - A minimal set of rules governing the arrangement, interaction, and interdependence of the parts or elements that must ensure that a conformant system satisfies a specified set of requirements
  - Provides the framework upon which engineering specifications can be derived, guiding the implementation of the system
  - Analogous to the building code that provides guidance for new buildings to be able to connect to the existing infrastructure by characterizing the attributes of that infrastructure

Operational Concept

- An important task in the architecture development process is to define the operational concept:
  - A concise statement that describes how the goal will be met
  - How will the system look and act in the operational environment
- Operational Concept Definition Parts:
  - How the system operates
  - Where in the operating environment the system will be distributed
  - How long the system must operate
  - How effective the system's performance must be

Operational Concept

- An operational concept is a shared vision from the perspective of the system's stakeholders of how the system will be:
  - Developed
  - Produced
  - Deployed
  - Trained
  - Used and maintained
  - Refined
  - Retired

Operational Concept

- The operational concept includes a collection of scenarios – one for each group of stakeholders for each relevant phase of the system's lifecycle:
  - Each scenario addresses one way that a particular stakeholder will want to use, deploy, fix, etc., the system and how the system will respond to produce a desired end
  - Scenario - a sequence of events which might occur that includes the interaction of the product with its environment and users, as well as the interaction among its product components

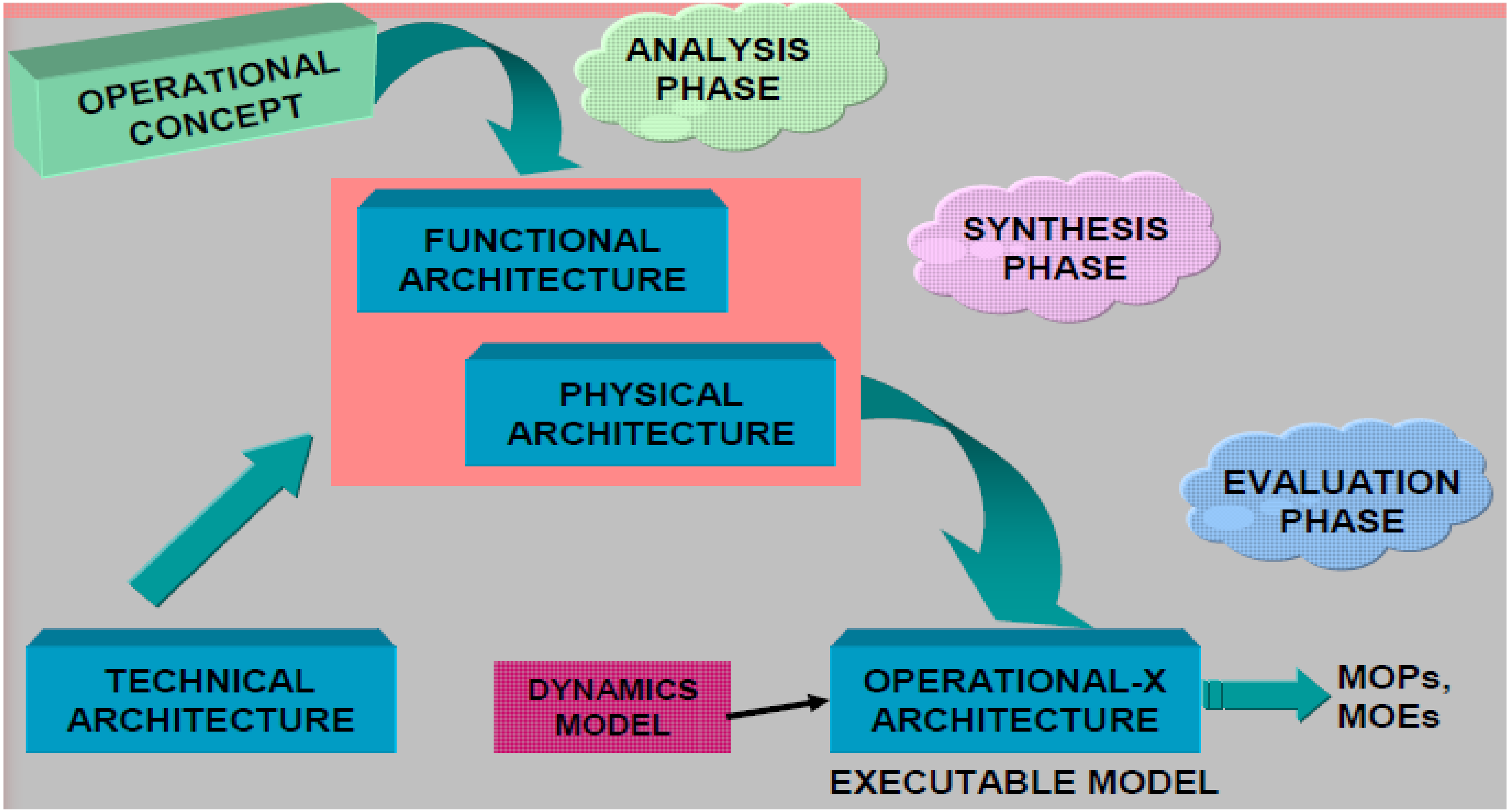
Executable Model

- The functional, physical, and technical architectures are static representations that attempt to describe the dynamic behavior of the architecture
- In order to analyze the behavior of the architecture and evaluate the performance characteristics, an executable model is needed

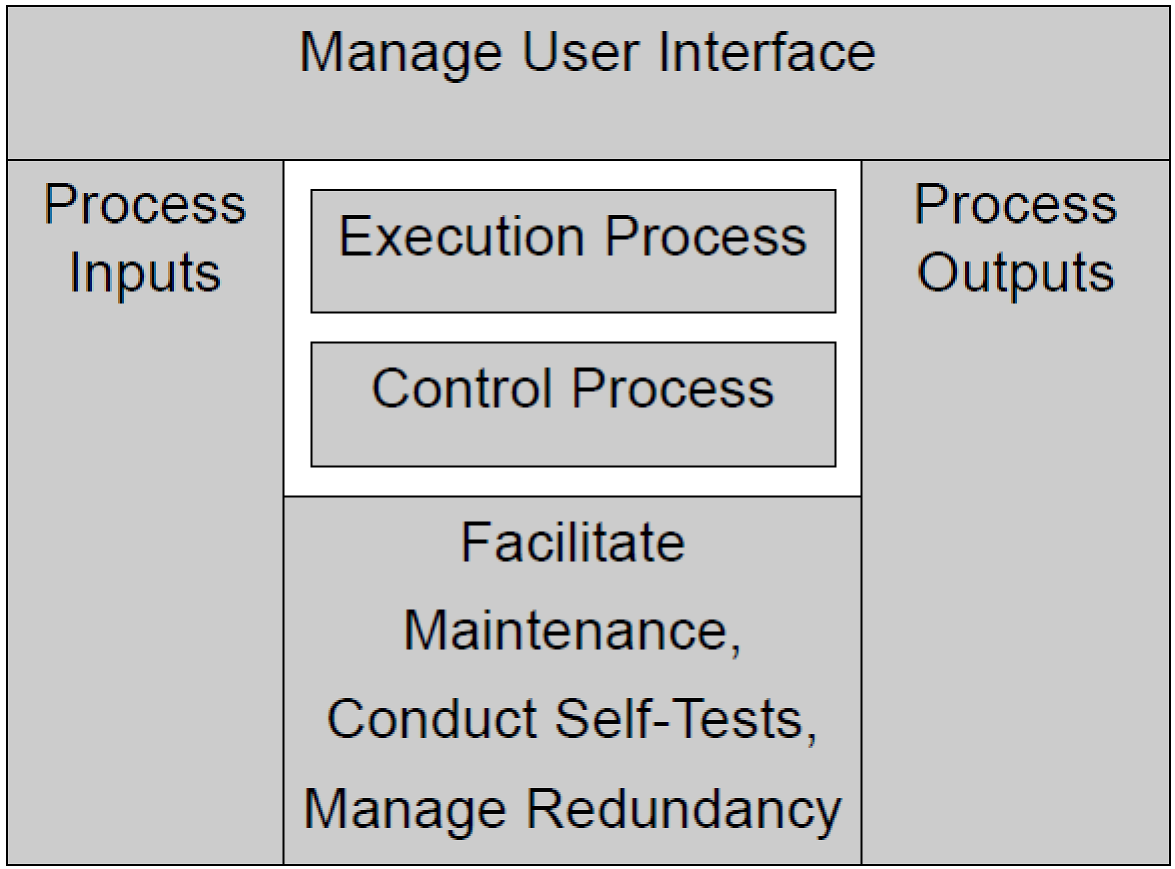
Architecture Development Process

- The architecture development process consists of three phases:
  - Analysis Phase – The static representatives of the functional and physical architectures are obtained using the operational concept to drive the process and the technical architecture to guide it
  - Synthesis Phase – The static constructs are used, together with descriptions of the dynamic behavior of the architecture to obtain the executable operational X-architecture (X = executable property)
  - Evaluation Phase – Measures of performance (MOP) and measures of effectiveness (MOE) are obtained

Three-Phase Process of Architecture Development



Hatley-Pirbhai Modeling



http://en.wikipedia.org/wiki/Hatley-Pirbhai_modeling

Architecture vs. Design

- A system architecture creates the conceptual structure within which subsequent system design occurs.
- Developing a system architecture and developing a system design are systems engineering functions that support system synthesis, but they have different uses.
- System architecture is used:
  - To establish the framework (i.e., constrains the trade space) for subsequent system design
  - To support make-buy decisions
  - To discriminate between alternative solutions
  - To ‘discover’ the true requirements or the ‘true’ priorities
- System design is used:
  - To develop system components that meet functional and performance requirements and constraints
  - To build the system
  - To understand the system-wide ripple effects of configuration changes

Summary

- **Architecture** – The fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution
- A **system architect**, not only knows about the individual components, but also understands the interrelationships among the components

Summary

- A **functional architecture** is:
  - A set of activities or functions that are arranged in a specific order and when activated, achieves a set of requirements
- A **physical architecture** is:
  - A representation of the physical resources
  - Expressed as nodes that constitute the system and their connectivity
  - Expressed in the form of links

Summary

- An important task in the architecture development process is to define the **operational concept:**
  - A concise statement that describes how the goal will be met
  - How will the system look and act in the operational environment
- A **technical architecture** is a minimal set of rules governing the arrangement, interaction, and interdependence of the parts or elements that must ensure that a conformant system satisfies a specified set of requirements

Summary

- The functional, physical, and technical architecture are **static representations** that attempt to describe the dynamic behavior of the architecture
- In order to analyze the behavior of the architecture and evaluate the performance characteristics, an **executable model** is needed