Session 4
System Architecture
Concept Generation
Reminder: A2 is due today!

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Topic</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 (group)</td>
<td>Team Formation, Definitions, Stakeholders, Concept of Operations (CONOPS)</td>
<td>12.5%</td>
</tr>
<tr>
<td>A2 (group)</td>
<td>Requirements Definition and Analysis</td>
<td>12.5%</td>
</tr>
<tr>
<td></td>
<td>Margins Allocation</td>
<td></td>
</tr>
<tr>
<td>A3 (group)</td>
<td>System Architecture, Concept Generation</td>
<td>12.5%</td>
</tr>
<tr>
<td>A4 (group)</td>
<td>Tradespace Exploration, Concept Selection</td>
<td>12.5%</td>
</tr>
<tr>
<td>A5 (group)</td>
<td>Preliminary Design Review (PDR) Package and Presentation</td>
<td>20%</td>
</tr>
<tr>
<td>Quiz (individual)</td>
<td>Written online quiz</td>
<td>10%</td>
</tr>
<tr>
<td>Oral Exam (individual)</td>
<td>20’ Oral Exam with Instructor 2-page reflective memorandum</td>
<td>10%</td>
</tr>
</tbody>
</table>
The “V-Model” of Systems Engineering

16.842/ENG-421 Fundamentals of Systems Engineering

1. Stakeholder Analysis
2. Requirements Definition
3. System Modeling Languages - MBSE
4. System Architecture Concept Generation
5. Tradespace Exploration Concept Selection
6. Design Definition Multidisciplinary Optimization
7. Verification and Validation
8. System Integration Interface Management
9. Commissioning Operations
10. Lifecycle Management
11. PFR
12. FRR

“V-Model”

Numbers indicate the session # in this class

*optional
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
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

© The COOL house plans company. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

Contemporary

Source - www.coolhouseplans.com

© The COOL house plans company. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.
Architecture - Informational

© Canon Inc. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.
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
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

Vehicle Powertrain Configuration Schematic (ISG Hybrid)

Credit: Dr. Carlos Gorbea, TU Munich

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

Schematic showing Form

Credit: Dr. Carlos Gorbea, TU Munich

© Carlos Gorbea. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.
Partner Exercise (2 min)

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

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
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

Creativity Models
- Leonardo da Vinci
- Albert Einstein
- Thomas Edison

Brainstorming

Group Dynamics
- Brainstorming
- PowWows
- MindBoggling Workouts

Stimulants
- Six Hats
- de Bono
- Bio-inspired design
- random inputs
- Provocations
- Challenges
- "motion"
- alcohol
- environment
- drugs

Structured Processes
- TRIZ
- Stimulants
- MindMapping

Creativity in SA
- 3 themes: creativity, ambiguity, complexity
- Types of innovation: high leverage

Models
- Fukuzawa
- Fuller
- Albert Einstein
- Thomas Edison
- Leonardo da Vinci

36
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

• 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…
"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
- **Corporalità** - 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

<table>
<thead>
<tr>
<th>Alternatives</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Factor B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Factor C</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
N = \prod_{i=1}^{m} n_i
\]

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

© AIAA. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.
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

<table>
<thead>
<tr>
<th>Device#</th>
<th>Abbrev</th>
<th>Device Description</th>
<th>Block Name</th>
<th>Parameters</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Single_Hull</td>
<td>Hull</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>2</td>
<td>Double_Hull</td>
<td>Hull</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>3</td>
<td>Tripe_Hull</td>
<td>Hull</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>4</td>
<td>Conventional_Chase</td>
<td>Chase</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>5</td>
<td>Integrated_Chase</td>
<td>Chase</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>6</td>
<td>Conventional_Skirt</td>
<td>Skirt</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>7</td>
<td>Total_Lift_Hydrofoil</td>
<td>Hydrofoil</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>8</td>
<td>Partial_Lift_Hydrofoil</td>
<td>Hydrofoil</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>9</td>
<td>Vertical_Lift_Fan</td>
<td>Vertical_Propulsor</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>10</td>
<td>Single_Air_Prop</td>
<td>Horizontal_Propulsor</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>11</td>
<td>Double_Air_Prop</td>
<td>Horizontal_Propulsor</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>12</td>
<td>Single_Water_Prop</td>
<td>Horizontal_Propulsor</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>13</td>
<td>Double_Water_Prop</td>
<td>Horizontal_Propulsor</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>14</td>
<td>Single_Waterjet</td>
<td>Horizontal_Propulsor</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
<tr>
<td>15</td>
<td>Double_Waterjet</td>
<td>Horizontal_Propulsor</td>
<td>0</td>
<td>1</td>
<td>(unused)</td>
</tr>
</tbody>
</table>

### Rules

```
Y DefineRule
Comparison | #BlockInstances
Hull <= Constant
Hydrofoil <= Constant
Skirt = Constant
Chase = Constant
Horizontal_Propulsor = Constant
Vertical_Propulsor = Constant
```

© Larry Zeidner. All rights reserved. This content is excluded from our Creative Commons license. For more information, see [http://ocw.mit.edu/help/faq-fair-use/](http://ocw.mit.edu/help/faq-fair-use/).
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.