Module Purpose: System Architecture

- Place system architecture development in context with needs analysis, conops, functional analysis and system design.

- Understand what system architectures are and some techniques for how they are developed.

- Acknowledge that architecture development is usually an inductive process that benefits from heuristics and the experience of the systems engineer creating the architecture (who is also known as the system architect).
“It must be remembered that there is nothing more difficult to plan, more doubtful of success, nor more dangerous to manage than the creation of a new system.”

- Niccolo Machiavelli, The Prince

**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.
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.

Developing Candidate System Architectures is Recursive and Iterative

- 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?
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?

Four Deductive or Inductive Methods Support Synthesis and Discovery

♦ Science-based, deductive methods:
  • Normative
    • Hard rules are provided (from somewhere), success is defined by following the rules
    • “If it doesn’t look like what we are doing now it must be wrong.”
  • Rational
    • Solutions derived from objectives
    • General systems problem solvers, optimization and formal techniques
    • Rule based

♦ Art-based, inductive methods:
  • Participative
    • Solution from group process, design by group consensus. Stakeholders involved
  • Heuristic
    • Lessons learned based
    • Develop solutions through soft rules that are driven by experience
Architecting Focuses on Refining the Problem to Be Solved While Developing Conceptual Solutions

- The development of a system architecture, also called ‘architecting’, is a systems engineering responsibility. It is the art and science of purpose determination and concept formulation.
- The essence of architecting is structuring, simplification, compromise, and balance.
- This balance is achieved by appropriate compromise between:
  - System requirements
  - Function
  - Form
  - Simplicity
  - Robustness
  - Affordability
  - Complexity
  - Environmental imperatives, and
  - Human factors
- Candidate architectures are compared using these factors and a baseline, or agreed to system architecture is chosen.
  - A choice is made despite the typically large uncertainties and occasionally ambiguous customer priorities.

Pause and Learn Opportunity

Have the students read the article on the Apollo architecture decision to use Lunar Orbit Rendezvous (Apollo_LOR_1971.pdf).
At the board, outline the alternative architectures that were under consideration for the Apollo missions.
  - Earth-orbit rendezvous
  - Direct ascent
  - Lunar-orbit rendezvous
Discuss the pros and cons of each and why LOR became the preferred architecture.
Pause and Learn Opportunity, part 2

Extend the discussion to include NASA’s current plans on returning crews to the Moon using a combination of Earth-orbit rendezvous and Lunar-orbit rendezvous.

What are the resulting architecture elements?
What are the pros of this approach?

Use the speech by M. Griffin to get a better understanding of the NASA architecture (MG-STA-speech/ESAS-arch_1/22/08.doc).

View the architecture representation with the graphic on the next slide.
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

Describing a Space System Architecture

♦ No one figure or diagram can capture a system’s architecture - it requires different ‘views’ or perspectives.

♦ Architecture descriptions are what we produce:
  • Spacecraft renderings and subsystem block diagrams
  • Space system communication flow diagrams
  • Functional flow diagrams - sometimes captured in functional flow block diagrams (FFBDs; as discussed in Functional Analysis Module)
  • Subsystem interface diagrams - frequently captured in N-squared diagrams (as discussed in Interfaces Module)

♦ By analogy with a building architecture, these are the blueprints, elevations, floor plans, budgets, wiring plans, etc.
The James Webb Space Telescope
Communications Architecture

- The launch vehicle injects observatory into an L2 transfer trajectory.
- The observatory operates at L2 point for 5 years with a goal of 10 years, providing imagery and spectroscopy in the Near and Mid Infrared wavebands.
- The Ground Segment receives downloads of science data and sends command uploads during daily 4 hour contacts.
- Ground Segment uploads plans to the Observatory once a week and the observatory autonomously executes these plans.

Ground Segment

NASA Provided Communication Asset for Early Orbit Phase

Launch Segment
Creating a system architecture is a systems engineering function that is the first step in translating a defined problem into a solution.

Creating an architecture is a recursive, iterative process that begins with the problem statement, creates some candidate solutions, assesses their merits and chooses one.

Architecture creation is not a deterministic process, but understanding the strengths, weaknesses and adaptability of heritage or analogous systems is a valuable first step.

In working with the stakeholders, the function or performance requirements of the system may be modified to create a better match between the problem statement and the candidate solution.

Like many systems engineering functions, architecting is one of balancing competing factors and choosing a preferred solution with uncertain and sometimes ambiguous information.

No one view captures an architecture. Many views are used to capture the system structure defined in terms of system elements, interfaces, processes, constraints, and behaviors.
Backup Slides
for System Architecture Module

Building Architectures
Illuminate by Analogy

- The architect works for the client and with the builder.
- You expect the architect to help develop requirements.
  - Both architects and systems engineers build self-consistent, balanced problem-solutions pairs.
- Architects produce abstracted designs.
  - Floor plans, elevations, cost estimates, etc. are not complete building plans, but they are necessary for complete building plans.
- Architecture descriptions and the architecture itself are different.
  - The floor plan is not the architecture, nor is the elevation, nor is the cost estimate.
- A good architecture representation is not just the physical structure, there are many views.

Mark Maier and Elsbeth Rechtin - The Art of Systems Architecting, CRC Press, 2000
The Three Views of the DOD Architecture Framework

Elements of Pre Phase A Mission Architecture
**Product Architecture**

- **Product architecture** is the arrangement of the physical elements of a product to carry out its required functions.
- It is in the Embodiment design phase that the layout and architecture of the product must be established by defining what the basic building blocks of the product should be in terms of what they do and what their interfaces will be between each other. Some organizations refer to this as *system-level design*.
- There are two entirely opposite styles of product architecture, modular and integral:
  - Modular: components (chunks) implement only one or a few functions and the interactions are well defined.
  - Integral: implementation of functions uses only one or a few components (chunks) leading to poorly defined interactions between components (chunks).
- In integral product architectures components perform multiple functions
- Products designed with high performance as a paramount attribute often have an integral architecture.