CMPE450 Lecture 01
Systems Engineering 0.101: A quick overview

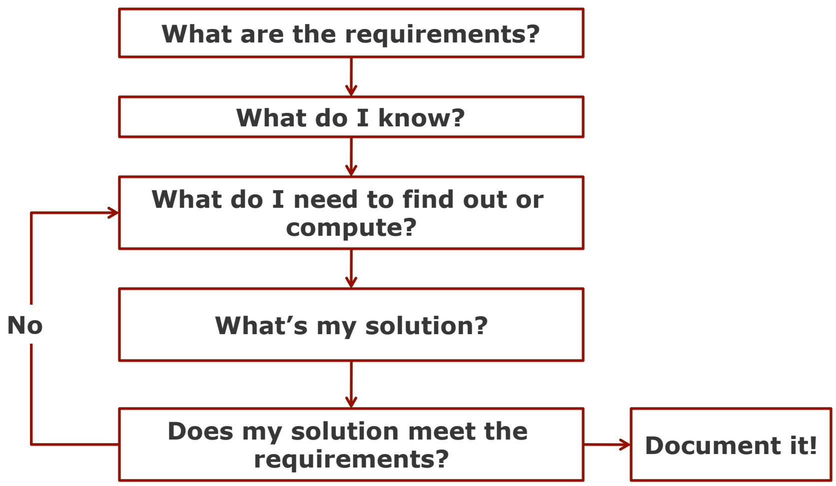
Part 1: What does my customer want?

Let's start at the very beginning

- How do I have any idea what it is I'm supposed to build?
- Do not skip this step…
- …otherwise you're running around building/ designing something (which is fun)…
- …with no idea at all when you will be done…
- …or if your work will be useful.
- You are an not an artist (not a bad thing).
- …you're an engineer, so your product must be useful!

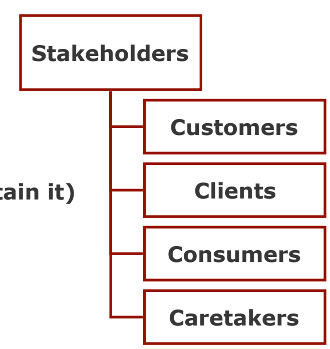
LaBerge’s Law #3

- Every engineering problem is a **design** problem.
- In fact *designing solutions* is what engineering do!



What’s a system? Who cares?

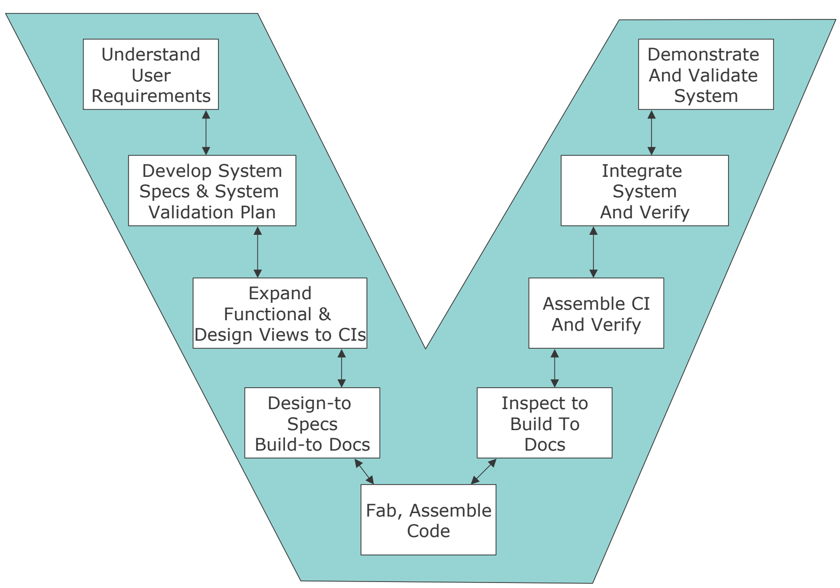
- Q: What is a system? (from a SE perspective)
  - A: “a collection of elements (e.g., hardware, software, people, facilities and procedures) organized to accomplish some common objectives”
- Q: Who defines the objectives?
  - A: the **stakeholders (the 4 c’s)**
    - ***Customers*** (those with the money)
    - ***Clients*** (those with the interest)
    - ***Consumers*** (those who use it)
    - ***Caretakers*** (those who repair or sustain it)



You don’t get paid until you’re complete (and successful)

- **What determines success?**
- **Heuristic: Success is defined by the stakeholder, not the architect.**
- **Heuristic: A system engineer (architect) must maintain a broad perspective of the problem.**
- **Heuristic (Aristotle): In order to understand anything, you must not try to understand everything**

The System Engineering “Vee”



Systems Engineering is a fractal process

- **Fractal**
  - Fractals are non-regular geometric shapes that have the same degree of non-regularity on all scales.
- **Heuristic:** One person's architecture is another person's detail. One person's system is another person's component.
- **Heuristic**: In general, each system level provides a context for the level(s) below
- **Heuristic**: Leave the specialties to the specialist. The level of detail required by the architect is only to the depth of an element or component critical to the system as a whole.
- **Heuristic**: The successful system architect or engineering understands the rudiments of all of the external systems and interfaces.

Create graphical artifacts first!

Create graphics first, text second when depicting artifacts of a system design.

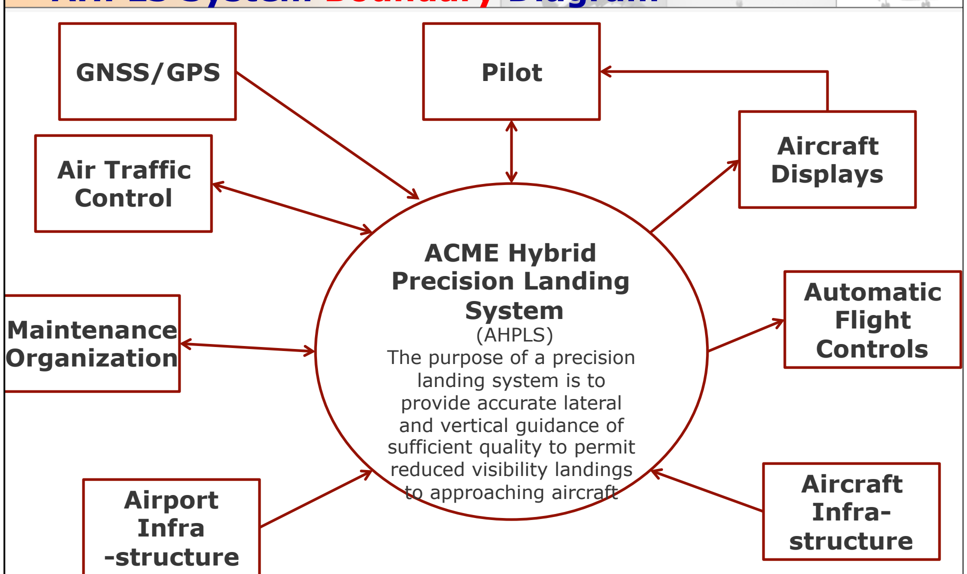
!

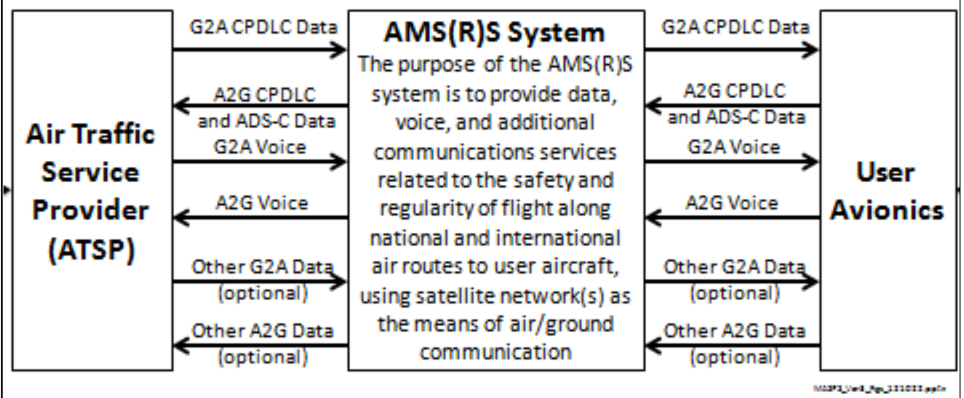
- Graphics assist in communication with customers, coworkers and other stakeholders
- Text documents will flow from the artifacts.
- **Heuristic**: A picture really is worth 1000 words!

System Boundary Diagram (SV-1)

- **Defines the context/scope of the work to be provided as part of your product or system.**
- **Identifies the external adjacent systems**
- **Heuristic: Diagram abstraction should result in 7±2 elements.**

AHPLS System Boundary Diagram





"Minimum Aviation System Performance Standard for AMS(R)S Data and Voice Communications Supporting Required Communication Performance (RCP) and Required Surveillance Performance (RSP) in Procedural Airspace" Washington, DC: RTCA, Inc., 2013.

What do we know about the interfaces?

- Tool is the **Information Exchange Matrix (OV-3)**
- Place to capture **tribal** knowledge
- Very important to keep a **high level** of abstraction
- Identify **problematic** interfaces
  - Interfaces for which we don't know what, how, how much or how fast
- Don't focus on implementation details!

Example attributes for information exchange matrix…

- Source
- Destination
- ID
- Description of the information received/ transmitted
- Media for the interface (wire/wireless/network, etc)
- Analog/Digital
- Size/Volume
- Type (Autonomous/Cooperative/Active)
- Update Rate
- Bandwidth
- Security
- Safety Criticality
- Mission Criticality
- Error Rate

Sample Information Exchange Matrix

- Write down everything you know at this level.
- Keep the level of abstraction appropriate
- **Heuristic**: It doesn't exist if it isn't written down!

	Information Source			Inf. Destination		Information Description & Attributes									
ID	OIE Source	Sub-Function	Producing Activity	OIE Receiver	Consuming Activity	Description	Media	Size/Vol.	Units	Interface Type	Freq.	Security	Safety Critical	Mission Critical	Quality/FOM/Error Rate

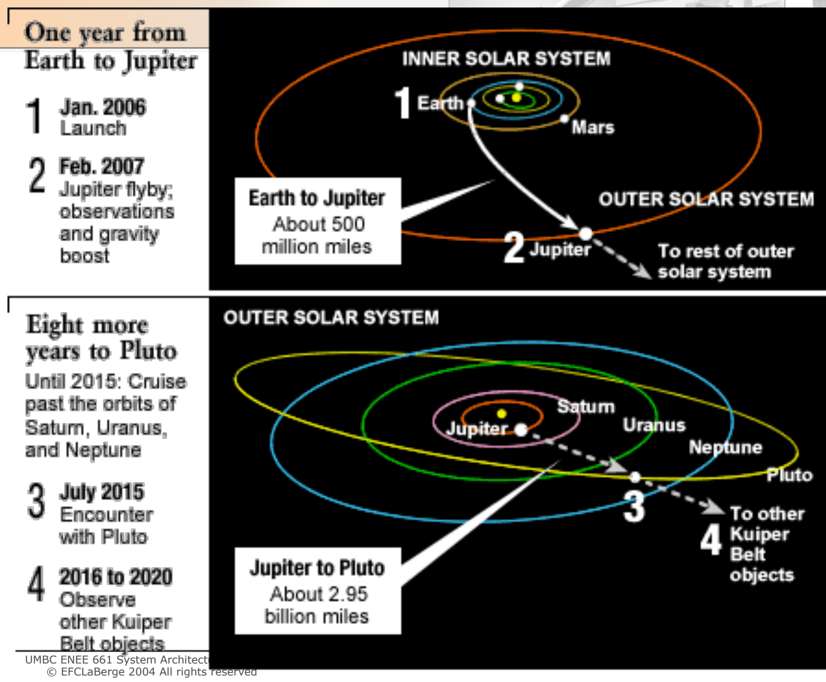
ID	Source	Subfnc	Producing Process	Receiver	Consuming Process	Description	Media	Size/Vol.	Units	Interface Type	Frequency	Security	Safety Crit?	Mission Crit?	Quality/ Error Rate
G2A_CPDLC_Data(g)	ATSP	n/a	Individual Controller	AMS(R)S	SBB Meet Me Points	ACARS Data containing CPDLC information ground source up	Leased	ACARS Packets	various	tbd	varies	CPDLC controlled	Yes	Yes	BER = 0.1 ppm
A2G_CPDLC_Data(g)	AMS(R)S	SBB Meet Me Point	various A/C systems	ATSP	Individual Controller	ACARS Data containing CPDLC information air source down	Leased	ACARS Packets	various	tbd	varies	CPDLC controlled	Yes	Yes	BER = 0.1 ppm
G2A_CPDLC_Data(a)	AMS(R)S	L-Band Transceiver	Digital I/O	Data Mgmt Unit (DMU)	Router	ACARS Data containing CPDLC information ground source up	ARINC429, AFDX, ARINC659, etc.	LAN packets	various	wire or fiber optic	varies	CPDLC controlled	Yes	Yes	BER = 0.1 ppm

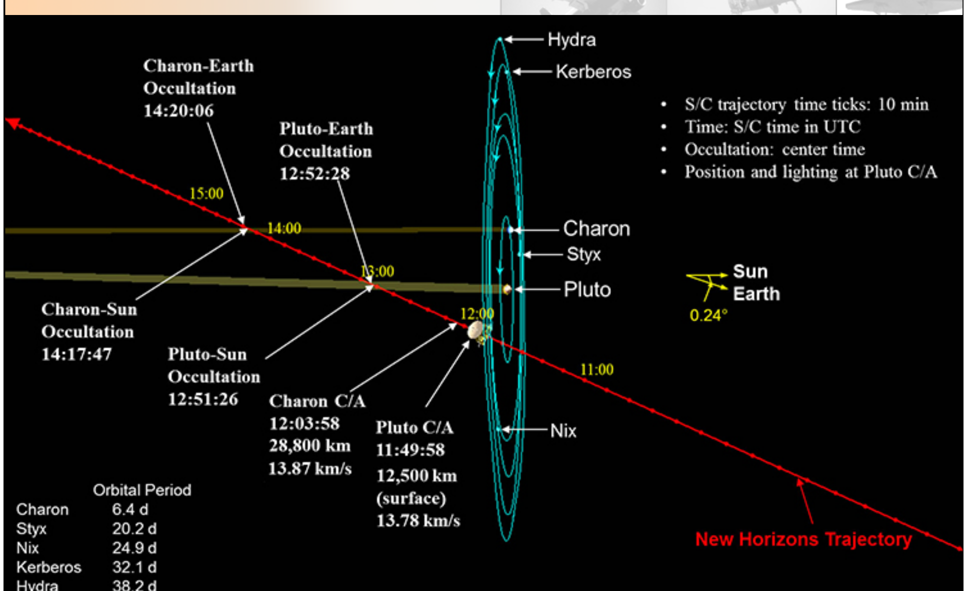
ID	Source	Subfnc	Producing Process	Receiver	Consuming Process	Description	Me(
G2A_CPDLC_Data(g)	ATSP	n/a	Individual Controller	AMS(R)S	SBB Meet Me Points	ACARS Data containing CPDLC information ground source up	Lea:
A2G_CPDLC_Data(g)	AMS(R)S	SBB Meet Me Point	various A/C systems	ATSP	Individual Controller	ACARS Data containing CPDLC information air source down	Lea:
G2A_CPDLC_Data(a)	AMS(R)S	L-Band Transceiver	Digital I/O	Data Mgmt Unit (DMU)	Router	ACARS Data containing CPDLC information ground source up	ARII AFD

Media	Size/Vol.	Units	Interface Type	Frequency	Security	Safety Crit?	Mission Crit?	Quality/ Error Rate
Leased	ACARS Packets	various	tbd	varies	CPDLC controlled	Yes	Yes	BER = 0.1 ppm
Leased	ACARS Packets	various	tbd	varies	CPDLC controlled	Yes	Yes	BER = 0.1 ppm
ARINC429, AFDX, ARINC659, etc.	LAN packets	various	wire or fiber optic	varies	CPDLC controlled	Yes	Yes	BER = 0.1 ppm

How is the system to be used?

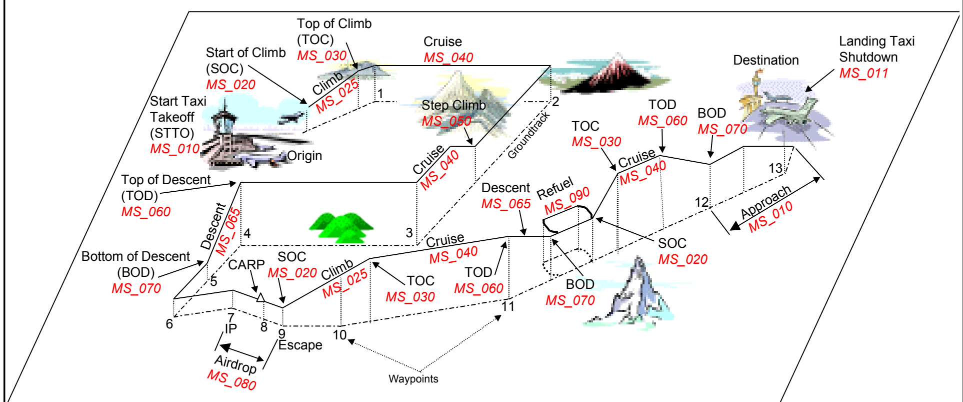
- Sketch out a (series) of "mission" alternatives
  - Drawing doesn't need to be complicated
- Identify the key events and when they occur
- Identify what the consumer actions are
  - E.g. push a button, turn power on, type command on keypad, etc.
- Identify external responses from the system
  - What changes can the user perceive?





	Orbital Period
Charon	6.4 d
Styx	20.2 d
Nix	24.9 d
Kerberos	32.1 d
Hydra	38.2 d

Sample Mission Scenarios Military FMS



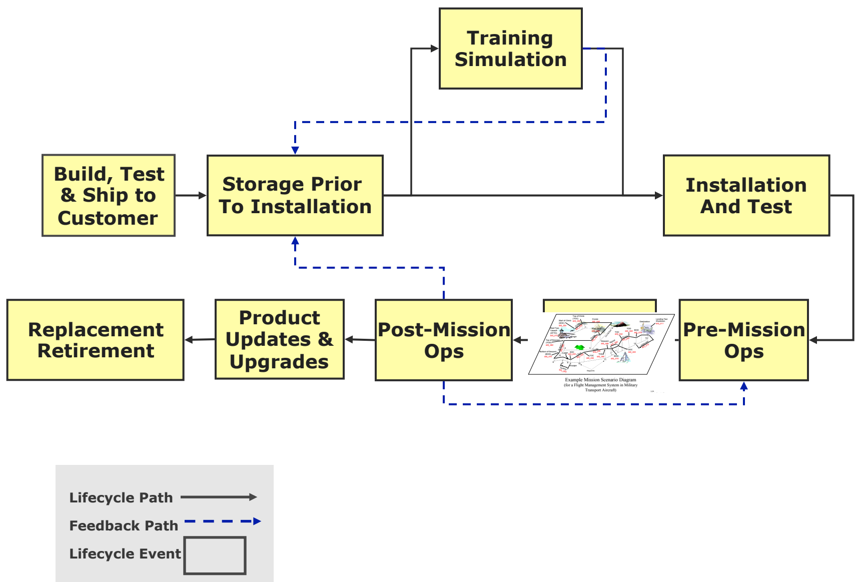
Example Mission Scenario Diagram
(for a Flight Management System in Military Transport Aircraft)

LGR

The operational scenarios

- **Consist of life-cycle events**
  - **Manufacture**
  - **Storage**
  - **Training**
  - **Installation**
  - **Mission Operation (previous graphic)**
  - **Post-mission**
  - **Maintenance**
  - **Upgrades**
  - **Replacement or retirement**

Operational Scenario Overview



Summary

- The importance of the operational view
  - Help clarify user demands
  - Consider all 4 C’s
    - Customer
    - Consumer
    - Client
    - Caretaker
- What’s in the system? (SBD)
- What are the interfaces? (IEM)
- How will we use it? (MSD)
- How will it grow or age? (OSD)

- Next time: The Functional View