

Summary of the Operational View

- 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



Systems Engineering 0.101: A quick overview

Part 2: The Functional View





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!**

Functional data flow & control flow diagrams

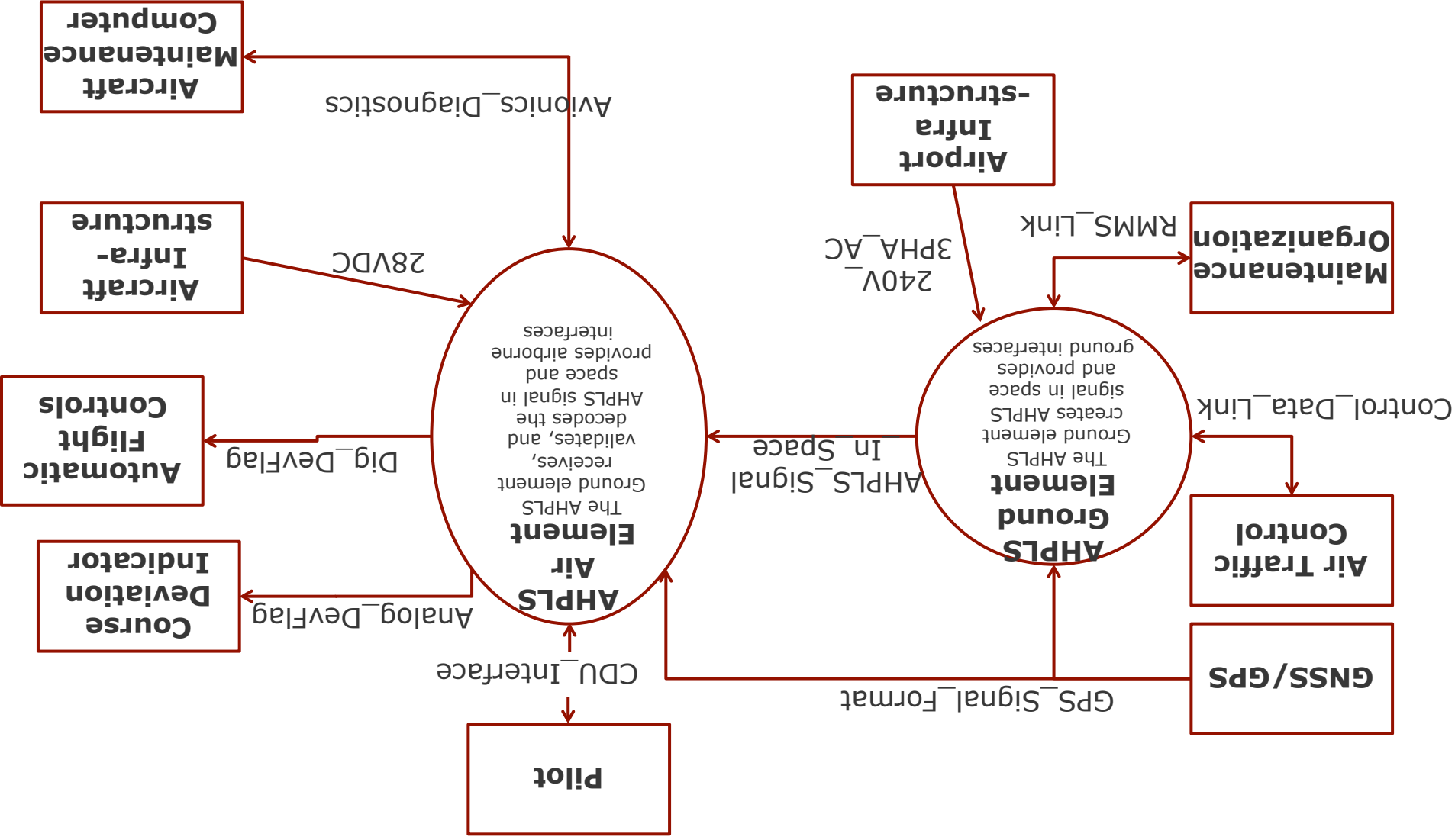
- Data flow diagrams and control flow diagrams are the step below System Boundary diagrams
- Graphical depiction of interactions between functions
- Stay focused on the **functions**, not the **implementation**
- Stay focused on the **functions**, not the **implementation**

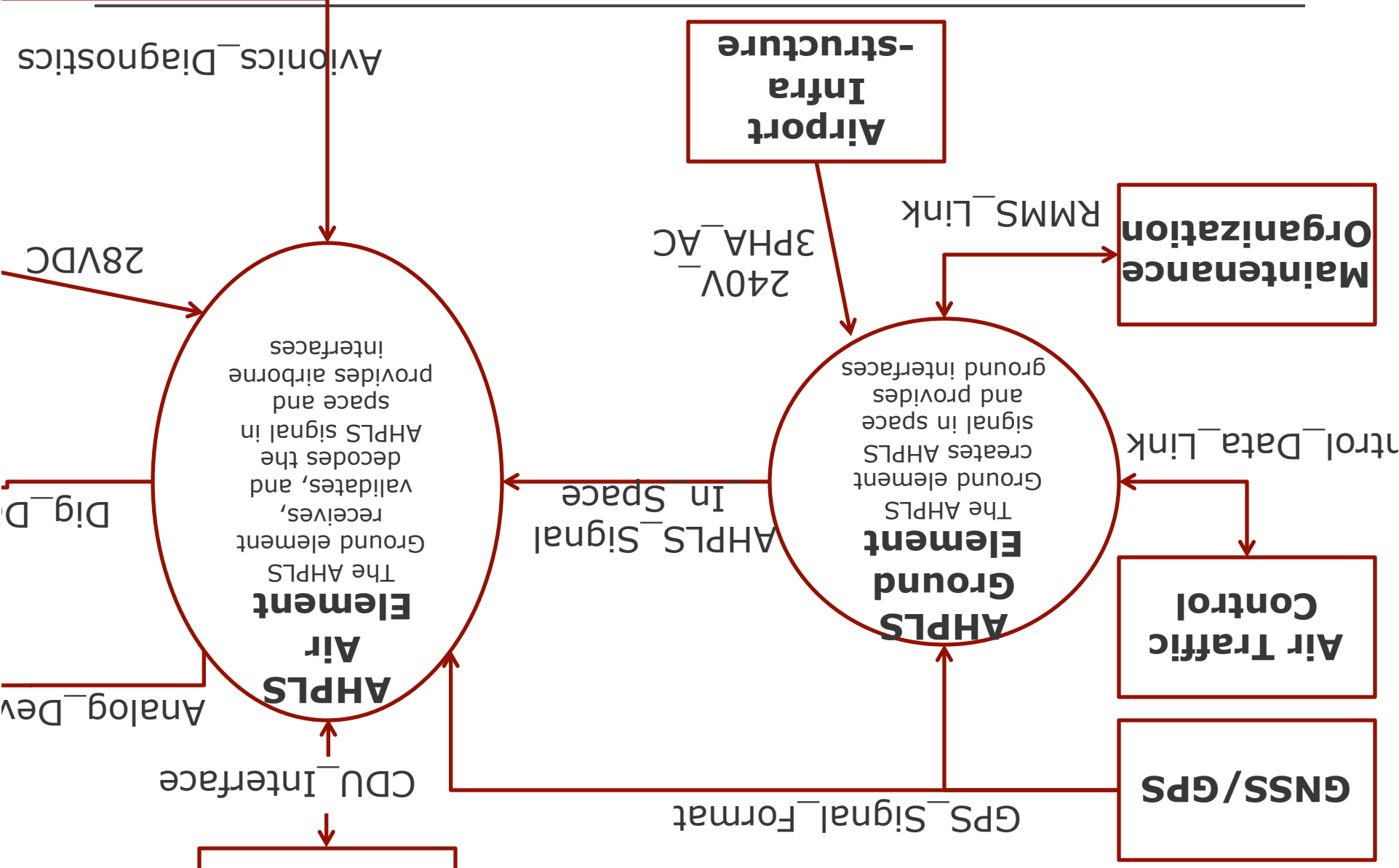
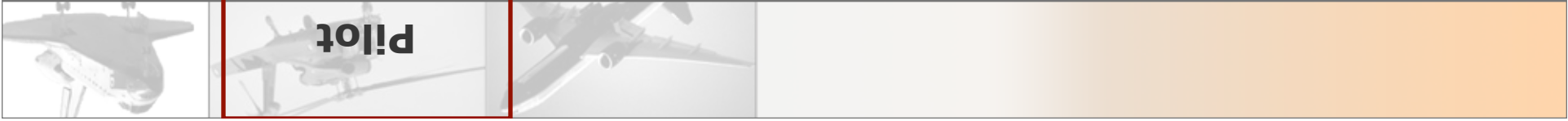
- Functions
- Verb phrases, e.g., "Compute path deviations"
- Show data and control flows differently
- Hatley-Pirbhai use solid & dashed
- IDEF0 uses entry/exit point
- Define the interfaces in an Interface Table or Data Dictionary

Site Decomposition

- The process of breaking down the SBD is known as decomposition
- Usually, the next level is *functional*...
- ...but sometimes the first level is decomposition to sites
- That's the situation for our AHPLS







Functional Data Flow/Control Flow Heuristics

- **Heuristic:** If you haven't defined the interfaces, you don't know anything.

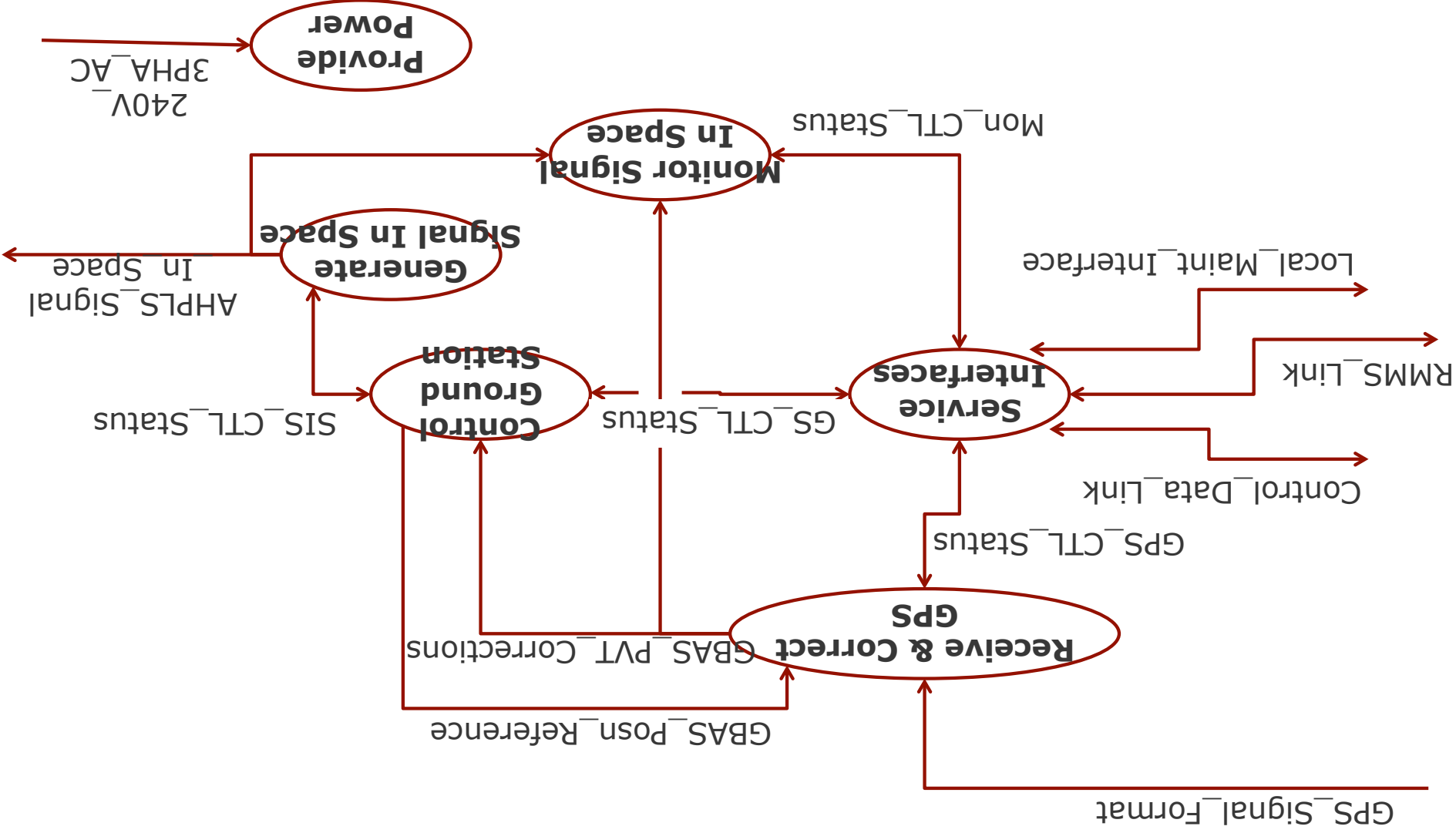
- **Heuristic:** One person's system is another person's component.

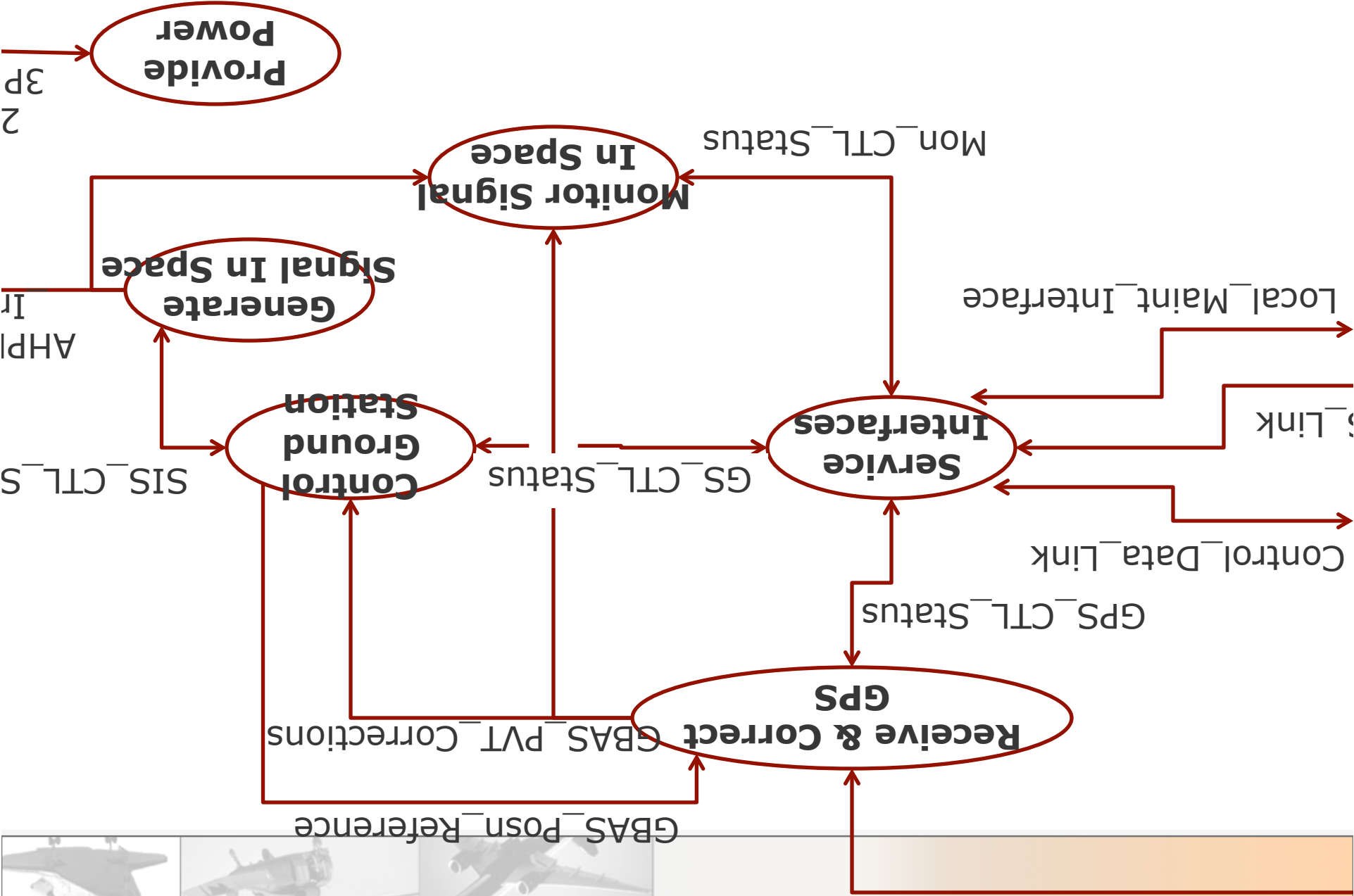
- **Heuristic:** Good quality interface specifications are simple, unambiguous, complete, concise, and focus on substance.

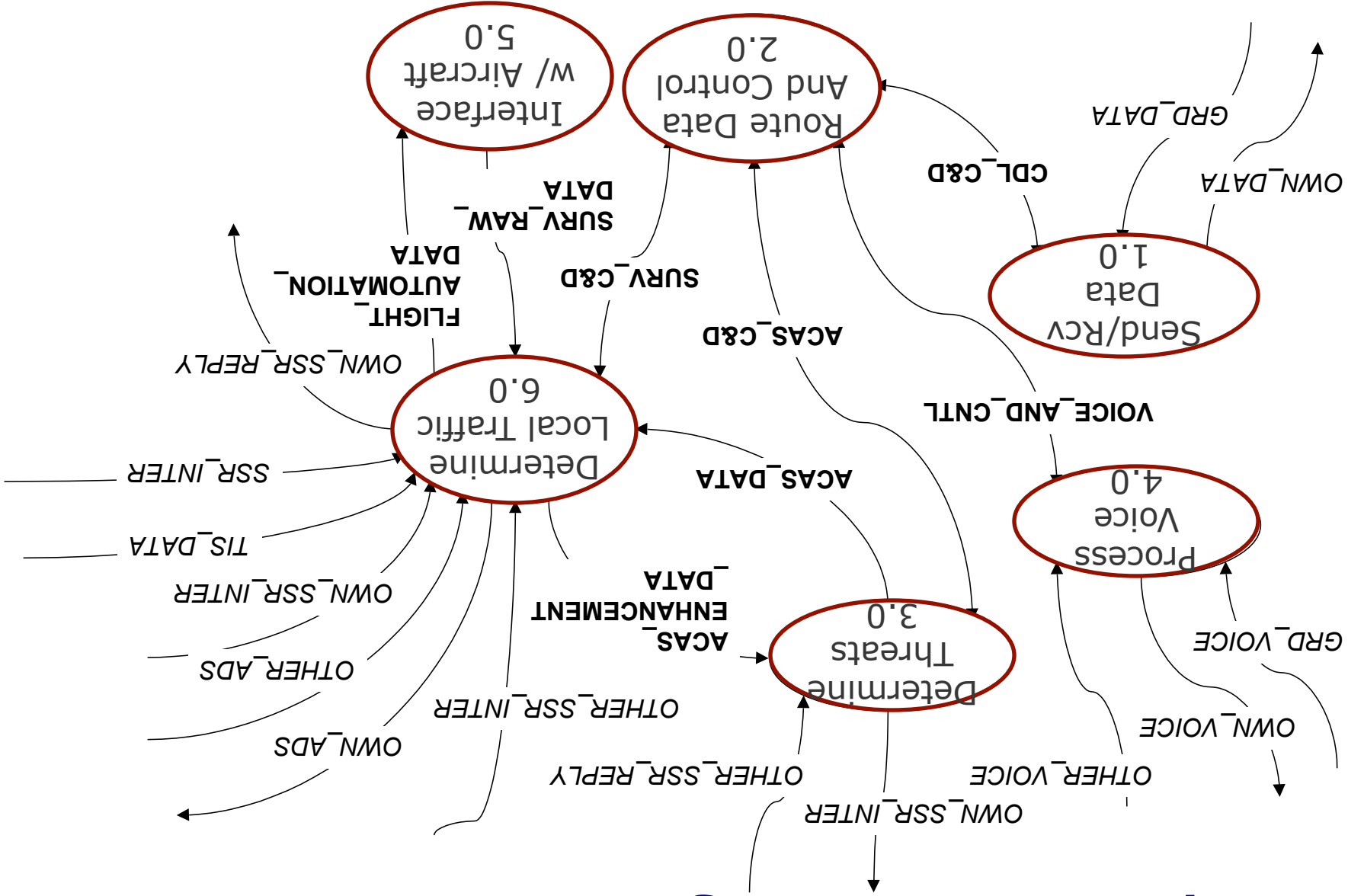
- **Heuristic:** Working documents should be the same as customer deliverables; that is , they should use the customer's language, not engineering jargon.



Ground DFD (first pass)



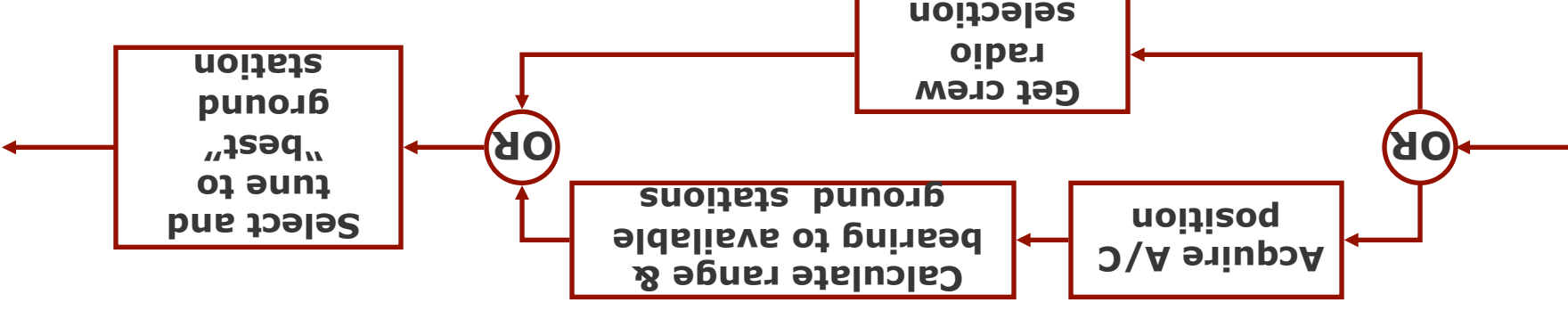




Functional Flow Block Diagram (FFBD)

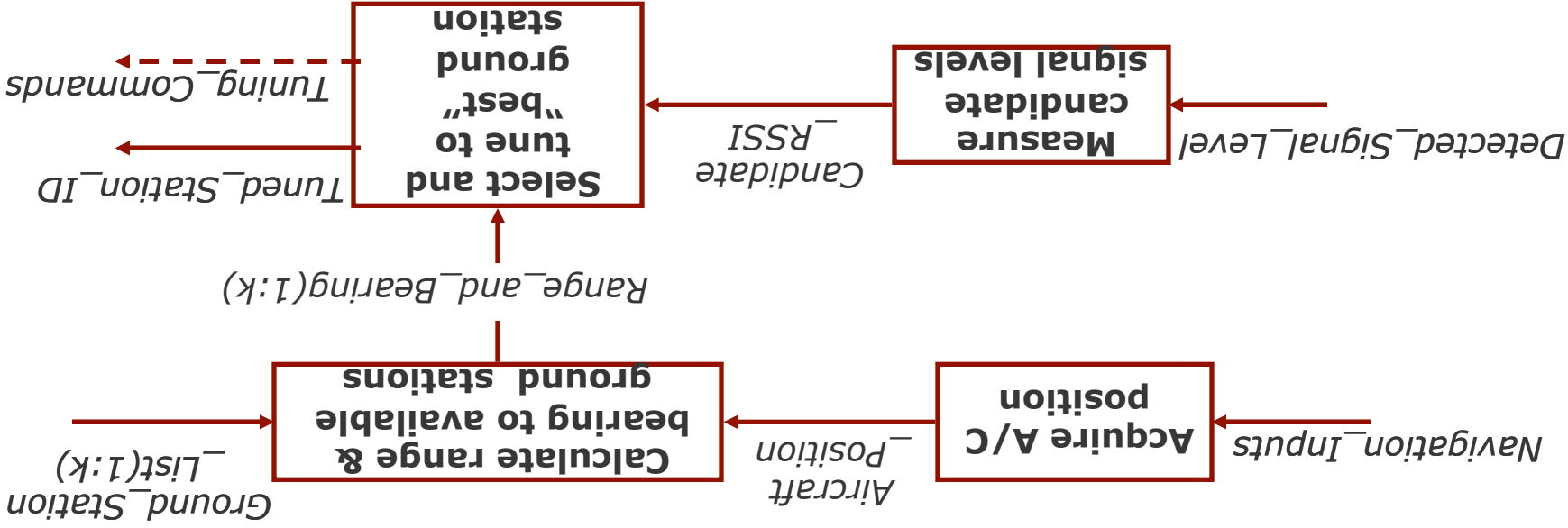


- **Purpose:** Depict serial/concurrent properties of the functionality
- **Design Goal:** Maximize concurrency
- **Consider:** Relationships between functions to determine dependencies
- **Do:** Consider sequence “left-to-right”
- **Don't:** Worry about data flows
- The FFBD is still a **static** model

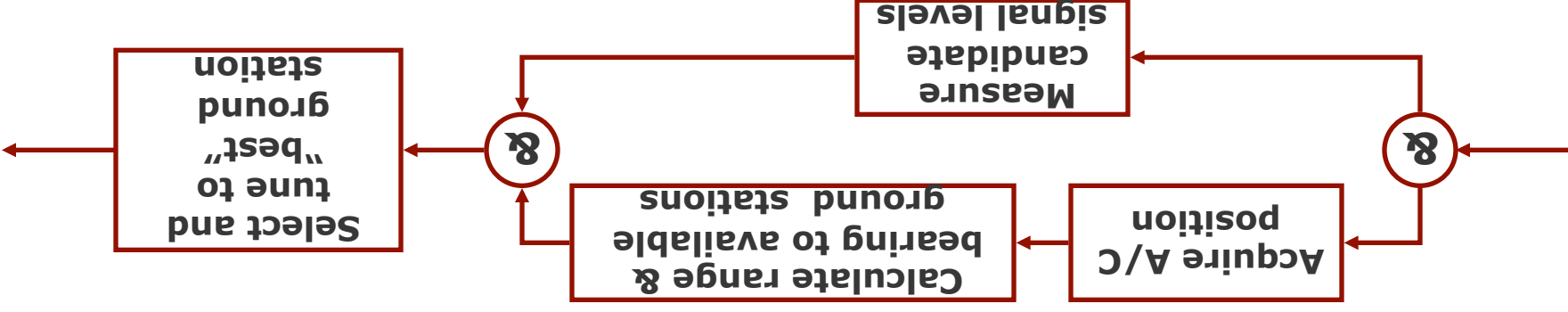




Data Flow/Control Flow Diagram vs. FBD



Data Flow / Control Flow Diagram



Functional Flow Block Diagram

Hints for working with FBDS

- At higher levels of abstraction, functions tend to be **concurrent**
- At lower levels of detail, functions eventually become **serial**
- To determine concurrency, consider if functions can run independently (given necessary inputs)
- **Serial relationships** tend to impose more constraints on implementation
- Understand **why** functions are concurrent or serial and be conscious of the implications



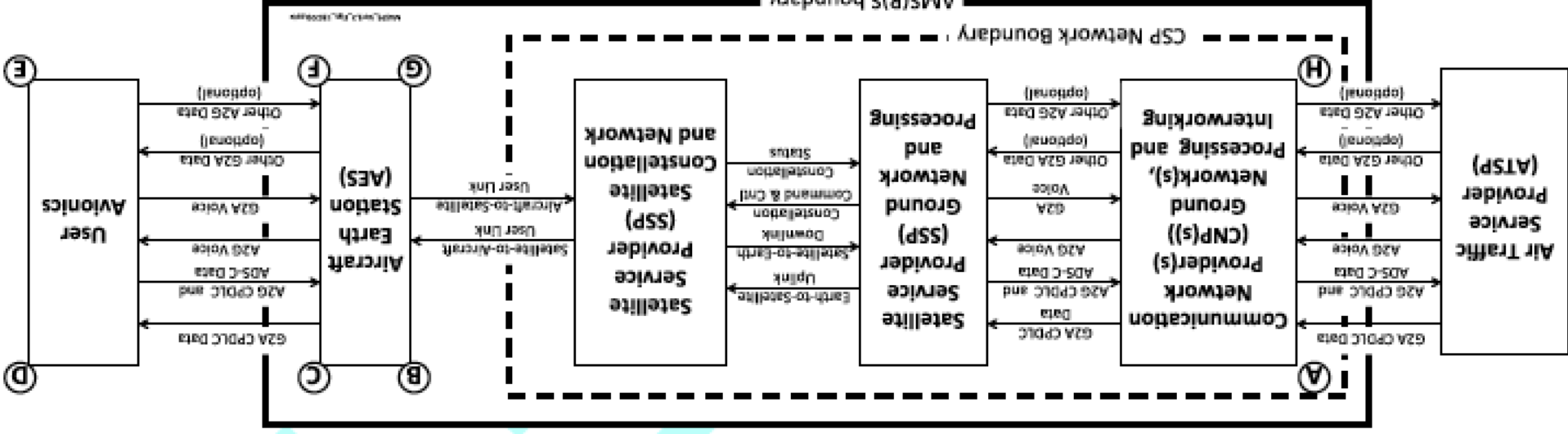


Figure 1-2: Notional AMS(R)S Partitioning

"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.

A few words about user interfaces

- In the “up front” functional analysis phase
 - Consider what information must be presented
 - Consider what information must be retained
 - Consider what information must be input
 - Consider users’ environments
 - Don’t forget the **caretakers!**
- In the “detailed” functional analysis phase
 - Develop user interface command structure
 - This can be viewed as a functional hierarchy, taking the commands as functions
 - Apply advanced techniques (not covered in 450/451)
 - “Leave details to the experts!”



Functional Trade Studies

- Trade studies are **hard to do** at the functional level
- **Heuristic**: Do the hard parts first
- What do we trade?

- Function **sets** defining different mission scenarios
- Different functional decompositions of the same higher-level function
- Alternatives should span the range of potential solutions
- What are our metrics?

- Number
- Complexity
- Reuse
- Scalability
- Integrity/reliability
- The trade need not be explicit or detailed, but does need to be captured.

- **Heuristic**: Simplify, Simplify, Simplify!
- **Heuristic**: It doesn't exist if it isn't written down



Functional/architectural tradeoff example

Functional Product Requirements - Operating Parameters		Maximize, minimize, or target	
Target Range	Relative Importance	Importance Rating	Customer Need
257			ADS-B functionality
236			Advanced Surveillance Functionality
221			DSB-AM Voice
223			ATS Data Comm.
294			Good availability
270			Good continuity
260			Minimize interference to existing Sys.
			Minimize susceptibility to existing Sys.
			Upgrade of existing radio
			Minimize Number of Antennas
			Comp. w/current antenna placement
			AOC Data Comm.
			Minimize acq. cost.
			Minimize installation complexity

0, 1, 3, 9

1,2,3,4,5

Things to consider

- **Heuristic: Act on fact!**
- Often the simple trade study will only show what you don't know.
- **Heuristic: Know what you don't know.**
- Be careful with your weighting, as it can change the results of table-based trades.
- **Heuristic: The last time your solution is perfect is before you show it to someone else.**

