Early Flight Control System Work Statement

Roger Racine

11 Sept. 2002

- **EFCS Task Statement**
- **Roles**

EFCS Task Statement

- The SSFPO, in Feb. 1992, asked Draper to develop an Early Flight Control System (EFCS) as a feasibility demonstration of flight critical functions essential for controlling the Space Station Freedom for Mission Builds 2-4.
  - Control System
  - Commanded from ground

ISS

Top Level Functionality

SVCS	GNC	ISE	C&T	RJ	EPS
DMS P	Process Control C	ISE Cont C	ACS Gnd Comm C	Exec Cmds P	Exec & Control P
UIL P	Attitude Control C	SYS Cont C	End-to-End Gnd Comm C	Monitor MDM P	Monitor EPS S
	Nav & Guidance C	S. Pwr Cont C	FDI C	Auto Track S	Ctrl Primary Pwr S
	Attitude Determ P	FR P	ACS St’able Ant P	FDI S	FDI S
	Pointing & Support P	Station Modes S	NonACS C/O S		
	FDI S				

EATCS and OMCS are not included

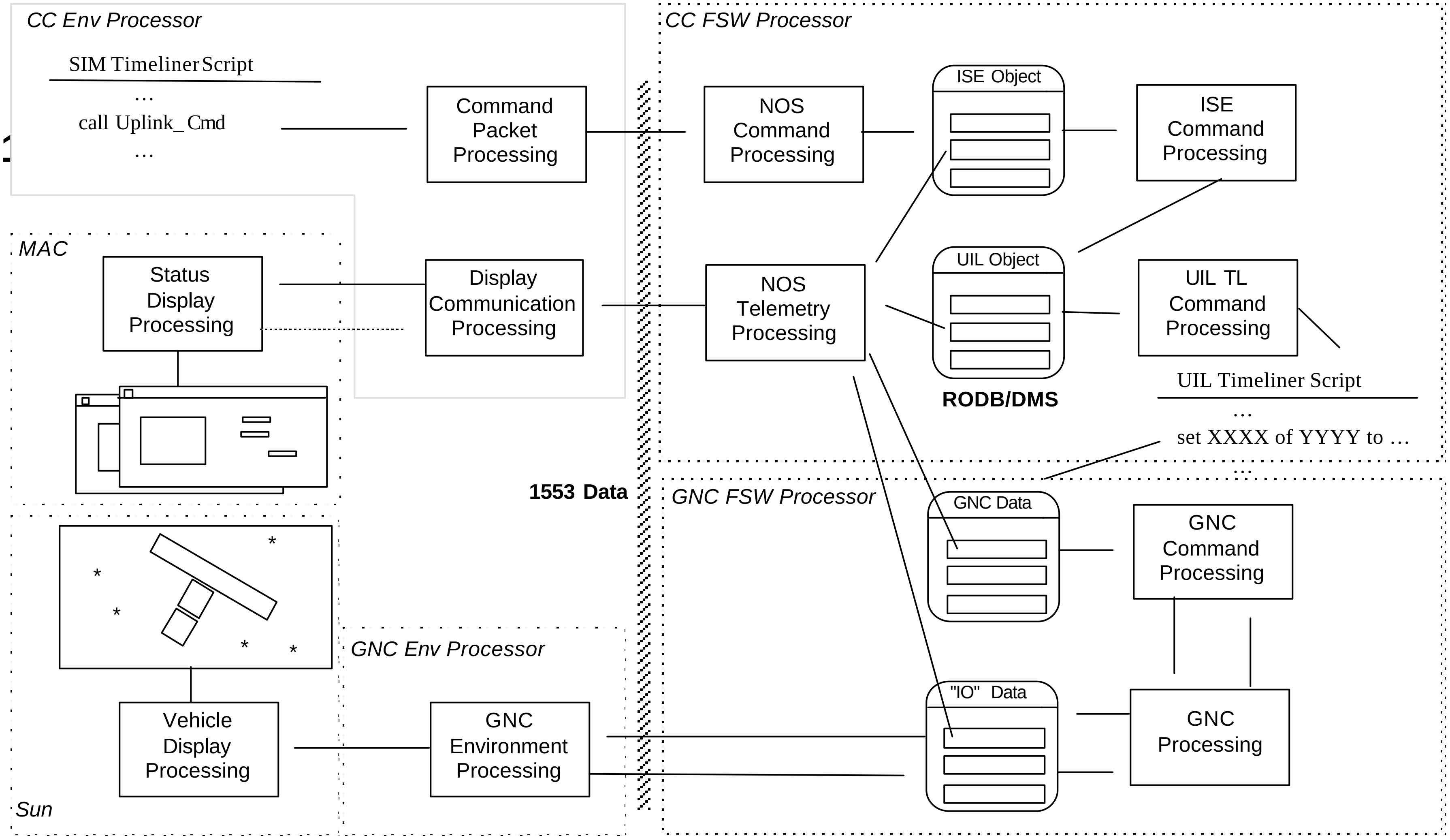
Complete **P**artial **S**tub Only

Host-Based Configuration

- Initial integration is performed, non-real-time, on host computer.
- Host and real-time testbed are running identical software except for machine-dependent routines.
- All systems and all environment modules, are linked together into one Ada program (real-time environment uses multiple Ada programs).
  - Application interfaces remain the same.
- Benefits
  - Instrumenting software for debugging does not affect timing.
  - It is possible to stop a simulation, look at data, and then continue.
  - Many simulations can run at the same time.

Demo Data Flow

ISS



Roles Needed

- For the Control System software, the following roles need to be partitioned among the available personnel:
  - — Overall leader
    - » Responsible for creating the Software Development Plan, maintaining the schedule, creating status reports, etc.
  - — Requirements Analyst
    - » Responsible for writing the Software Requirements Specification (SRS)
  - — Control algorithm developer
    - » Responsible for the design of the control systems
      - Generates at least the Top-Level Design documentation for the Control system
  - — Software architect
    - » Responsible for the high-level software design
      - Creates at least the Top-Level Design documentation laying out the structure of the software

Roles (Continued)

- Control software coder
  - Writes the Control software
- Design documenter
  - Writes the Detailed Design document
- Test Lead
  - Writes the Software Test Plan
- Test SW algorithm developer
- Test SW coder
- Version Control person
  - Responsible for dealing with the version control system
- Integration lead
  - The problem solver. Responsible for integrating the Control software with the other software in the ISS, and getting it to work
- Display developer
  - Takes telemetry data and displays it

Guidelines

- Expect requirements changes
  - Trying to stay ahead of the main developers means NASA or the contractors might change something
- The customer wants demonstrations. Part of the job is making sure the demonstrations are professional
  - Look good
  - Provide enough information to show the system working well
- All the software was developed quickly. There is no guarantee that problems are all due to new software