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
## Top Level Functionality

<table>
<thead>
<tr>
<th>SVCS</th>
<th>GNC</th>
<th>ISE</th>
<th>C&amp;T</th>
<th>RJ</th>
<th>EPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMS P</td>
<td>Process Control G</td>
<td>ISE Cont C</td>
<td>ACS Gnd Comm C</td>
<td>Exec Cmds P</td>
<td>Exec &amp; Control P</td>
</tr>
<tr>
<td>UIL P</td>
<td>Attitude Control C</td>
<td>SYS Cont C</td>
<td>End-to-End Gnd Comm C</td>
<td>Monitor MDM P</td>
<td>Monitor EPS S</td>
</tr>
<tr>
<td></td>
<td>Nav &amp; Guidance C</td>
<td>S. Pwr Cont C</td>
<td>FDI C</td>
<td>Auto Track S</td>
<td>Ctrl Primary Pwr S</td>
</tr>
<tr>
<td></td>
<td>Attitude Determ P</td>
<td>FR P</td>
<td>ACS St’able Ant P</td>
<td>FDI S</td>
<td>FDI S</td>
</tr>
<tr>
<td></td>
<td>Pointing &amp; Support P</td>
<td>Station Modes S</td>
<td>NonACS C/O S</td>
<td></td>
<td>EATCS and OMCS are not included</td>
</tr>
<tr>
<td></td>
<td>FDI S</td>
<td></td>
<td></td>
<td></td>
<td>Complete</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Partial</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Stub Only</td>
</tr>
</tbody>
</table>
• 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.
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
— 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
ISS 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