Satellite Operations on $15 a Day

Ground System Architectures Workshop 2003

March 4-6, 2003

Jeff Jacobsen - Boeing
Mark Lafferty - Boeing
Cesar Lindo - Boeing
Steve Sponaugle - Boeing
Hank Tseu - Boeing
Agenda

Background/Problem Statement

Solution Path

Architecture

Results

Future Work

Lessons Learned
Background

- 80 Satellite Vehicle (SV) LEO constellation controlled from a distributed OS/COMET™ based ground system
- Average 8.5 activities / SV / day => 680 daily activities
- 8 operators / shift x 4-12hr shifts => 32 operators
- Average 15 operator errors/month
  - No breaks
  - Fatigue
  - Repetitive work
- Shrinking ops budget
- The state of the delivered ground system software was not optimized for automated operations
Solution Path

• Mechanization precursors:
  – Pass plans / perform files
  – Automated testing tool

• Operations staff built automation system
  – Hybrid prototype / incremental build lifecycle
  – Relied on system hooks built into ground system
  – Trial & error to mitigate ground system deficiencies
  – Open source platform (Perl) combined with Comet Command Language (CCL)
Architecture

Mission Planning → Read Planned Activities → Create Automated Procedures

Satellite Control

SVX → SVY → SVZ → OS/COMET™ → Iridium SV
Architecture (continued)

• Properties:
  – ~12 KSLOC
  – Perl / Tk / CCL based

• Features:
  – Activity feedback
  – Retry ability
  – Activity flow control
  – Failure reporting / logging /metrics
  – 99.9% success rate
• At FOC there were 8 operators per shift on a 7 by 24 schedule executing over 900 objectives per day. (32)
• Through process improvements the number of operators was reduced to 5 per shift on a 7 by 24 schedule. (20)
• The workload was further reduced to 2 per shift on a 7 by 24 schedule by implementing automation. (8)
• In the initial stages of the program there was an average of 15 operator errors per month.
• With process improvements, training, and automation the operations now average less than 10 operator errors per year.
Future Work

• Clean up code (in progress)
  – Further modularization.
  – Add more comments/documentation
  – Facilitate maintenance
  – Promote code re-use
  – Remove unutilized functionality

• Implemented markup language to define activities
  – Simplify the addition of new activities
  – Allow addition of new features
  – Facilitate file maintenance

• Implemented dynamic activity re-schedule (in progress)

• Implement full feedback loop (in design)
  – Auto-respon to SV anomalies
  – Heuristics based on metrics to further improve performance
Lessons Learned

• Automation work would not have been possible without having ground system hooks in the software.
• Changing requirements due to a dynamic operations environment makes it impossible to completely determine requirements for automation software.
• Ground system developer and vendor support are an important key to success.
• It is critical to plan for automation from the beginning of the lifecycle and lay the foundation for automation along the way.
• Plan to perform the automation work when the operations has matured, or at least plan to dedicate resources to tuning the automation to suit the realities of operations.
• Plan for real-time operations process improvement as early as possible to free up resources to work on automation tasks.
End of Presentation

QUESTIONS?