

Wideband GapFiller Activity-Based Commanding

Andrew B. Rowland
WGS Project Engineer
The Aerospace Corp.



Overview

- **The Old Days: Time-Based Commanding**
- **The New Days: Activity-Based Commanding**
- **Design Example**
- **Design Considerations**
- **Challenges**
- **Summary**

Time-Based Commanding

- **Use Case:** Placing a satellite processor variable in the telemetry wavetrain
- **Read a Step:** Operator would receive a contact support plan and see that processor variables would need to be placed in the telemetry wavetrain.
- **Write a Step:** Operator would look up addresses of variables, convert to commands, write the commands on a passplan. Second operator would check the commands.
- **Do a Step:** If all is good, third operator would send the commands and verify the results.
- **Get a Pat on the Back**

Activity-Based Commanding

- ***Shift in Thinking*** – Want to accomplish the activity of placing variables in the telemetry wavetrain
 - Let the code take care of the details
 - Variable Lookup
 - Command Generation
 - Fault recovery
- The operator is happy when the activity completes, not each individual command
- Allows the operator to focus on bigger picture

Design Example

- **With WGS, there are many valid solutions to this problem of modifying the telemetry wavetrain**
 - **Most efficient use of modification space**
 - **Most expedient way to command the changes**
 - **Most flexible use for future activities**
 - **Permanent or temporary configurations**
 - **Etc.**
- **The circumstances at the time of the activity determines the solution method**
- **Our code is implemented on the ground segment, but could certainly be implemented in the space segment**

Design Example Cont'd

- **1 Day: Design TIM – Defined requirements**
- **1 Day: Pruned Requirements based on time/money**
- **3 Weeks: Code and unit test ~1850 lines of code**
- **1 Week: Beta test and provide feedback**
- **2 Days: Fix Beta Issues**
- **1 Week: Formal test, including Beta issues**
- **Deliver and use**
- **Tweak as mission needs dictate – repeat many of the steps above**

Design Considerations

- **Team approach critical: Operators, Coders, Satellite Engineers, Program Office**
- **Time and money determined the level of sophistication of our code**
 - **Operator specifies the desired solution method rather than the code looking at circumstances and picking a method**
 - **Fault recovery is done by the operator. Fault recovery could certainly be done by the code but budget did not allow for this level capability.**
 - **If a fault is detected, processing stops and asks operator for direction**

Challenges

- **Operators and Commanders are used to time-based commanding and are resistant to change methods**
 - **Solution is to test, test, test and demonstrate**
 - **Confidence in this method comes with use**
- **Requirements were pruned that should have been kept resulting in tweaks and retesting later**
 - **Learn from past experience and apply to future efforts**
 - **Keep design open to allow future changes**
 - **Choose design wisely!**
- **Operator knowledge atrophy and fault recovery**
 - **Initial and recurring operator training**
 - **Specific tools built to aid in fault isolation and recovery from faults**

Summary

- **Activity-Based commanding requires a shift in thinking**
- **Design, engineering and testing can be extensive and expensive**
- **More efficient operations, allows operator to focus on big picture**
- **Time and money determine level of sophistication**
- **Operators and engineers are still capable and can solve many problems during the contact**