National Aeronautics and Space Administration

NASA Safety Center Special Study

APRIL 2009 *SPECIAL STUDY: NASA CLOSE CALL*

Shuttle Software Anomaly

A few minutes after the Shuttle Endeavour reached orbit for STS-126 on November 14, 2008, mission control noticed that the shuttle did not automatically transfer two communications processes from launch to orbit configuration. Primary communications continued to use **S-band** frequencies after they should have transferred to the more powerful **Ku-band**. The link between the shuttle and its payload—the **Payload Signal Processor (PSP)**—remained configured for a radio link rather than switching automatically to the hardwired umbilical connection.

Fortunately, mission control was able to manually command both the S-band/Ku-band switch and the PSP port shift. While mission control was not able to re-instate automatic transfers during flight, they continued to monitor communications and manually operated necessary transfers for the remainder of the mission. STS-126 completed its mission successfully and returned to earth without further software problems.

While the software problems did not endanger the mission, they caught management's attention because "in-flight" software anomalies on the shuttle are rare. Software goes through rigorous reviews during development and testing to prevent this sort of problem, and most software anomalies are detected and fixed long before the shuttle leaves the ground.

BACKGROUND

SHUTTLE FLIGHT SOFTWARE

The Shuttle's Primary Avionics Software System (PASS) contains approximately 500,000 lines of flight source code. To maintain careful configuration control, every major functional update is identified as an **Operational Increment (OI)**. Each OI triggers a rigorous software development process that includes requirements definition, software design code development, system build verification, system performance verification, and mission preparation. A single OI takes about 18 months to complete.

SOFTWARE COMMANDS

Shuttle software typically stores data and output commands in blocks of code. These are known as **common data pools (compools)**. An addressing restriction requires output commands to be stored at even-numbered addresses within the compools. The code conventionally uses **fullword** alignment to name and store commands. Fullword alignment uses four bytes of information to specify the data address. When used consistently, fullword alignment forces all outputs to even addresses.

To accomplish automatic communication handovers, the Systems Management computer retrieves commands from compool locations and sends them to the Ground Command Interface Logic (GCIL), which controls configuration of the shuttle communications and tracking system. Each GCIL command requires two consecutive commands: a reset command followed by the new configuration command.

ISS018E008539

Figure 1: Space Shuttle Endeavour preparing to dock with the International Space Station during STS-126. The round Ku-band antenna is visible to the right of the cockpit. The antenna is stowed in the payload bay until the shuttle reaches orbit.

Shuttle operations that rely on radio frequencies for direct shuttle to ground communications during missions use S-band frequencies (1,700-2,300MHz) during launch, then switch to Ku-band communications (15,250-17250MHz) while in orbit. Once Ku-band communications are operational, the S-band remains as a fail-safe backup should the Ku-band lose signal.

The shuttle Payload Signal Processor (PSP) can be configured via RF link or hardwired umbilical, with the RF link typically used for deployed payloads and the umbilical for payloads that remain in the payload bay. The PSP can be switched back and forth between the RF and umbilical links as necessary throughout the mission to command different payloads.

WHAT HAPPENED?

In 2007, OI-33 inserted a **halfword** (two bytes rather than the fullword's four) of new data in a compool. The halfword addition shifted three GCIL output commands within the compool from even addresses to odd addresses. When OI-33 debuted on STS-126, the GCIL no longer received the appropriate commands from the Systems Management computer, and crucial communications handovers had to be manually initiated by mission control.

Conditions leading to this anomaly were introduced years before STS-126 launched. One of the primary contractors reporting on this anomaly to the Shuttle Avionics Software Control Board described the software error as a trap that evolved over several code modifications:

SHUTTLE SOFTWARE DEVELOPMENT

There are multiple levels of verification in the flight software development process. Relevant to this study, Level 6 is devoted to software verification for new or altered code and any code that was directly impacted by code changes. Verification and validation continue during Level 7 testing, which includes both nominal and off-nominal regression tests. Level 8 focuses on reconfiguration testing in preparation for the mission, using flight-specific data in tests similar to Level 7 testing.

After development is completed, testing and simulation continue at the **Shuttle Avionics Integration Laboratory (SAIL)**, a facility that is designed to test actual shuttle hardware and flight software in a simulated flight environment. The Integrated Avionics Verification (IAV) team conducts full integration testing at SAIL with the ability to configure a full complement of avionics.

SETTING THE TRAP In July 1989, OI-20 introduced new code that used fullword alignment for the three GCIL commands but did not use additional techniques to "lock down" outputs to even addresses. Instead, the OI-20 programmers left warning comments in the code immediately after the change history to notify future programmers of the need to monitor these addresses over subsequent changes (Figure 2a). This violated the intent of a programming standard that was later determined to be unclear; while the standard required programmers to use additional techniques to force outputs to even addresses, the original wording could be interpreted as merely a requirement for fullword alignment.

CAMOUFLAGING THE TRAP During OI-29 in August 2000, space restrictions in the compool forced the compool's change history to the bottom of the code. The warning added during OI-20 remained in its original location at the top of the compool code. Although the warning was still in the code, the new arrangement effectively buried the warning (Figure 2b). Originally, the warning was in a place that would always be noticed by teams adding to the change history, but in the new arrangement, the warning would not be noticed unless that specific section of code was being reviewed.

FALLING INTO THE TRAP OI-33 (January 2007) introduced the error that compromised automatic communication shifts on STS-126. The OI was not intended to change any code related to the affected functions on STS-126, but it added one halfword of data in the common data pool. The halfword shifted commands from the required even addresses for output data to odd addresses (Figure 2c). No one noticed the warnings that had been buried by OI-29. Reviews, inspection, and development testing missed the induced alignment problem, and software verification focused on the modified code. When the shuttle reached orbit, the GCIL received unrelated data instead of commands to initiate PSP port moding and the S-band/Ku-band handover. This triggered the anomaly.

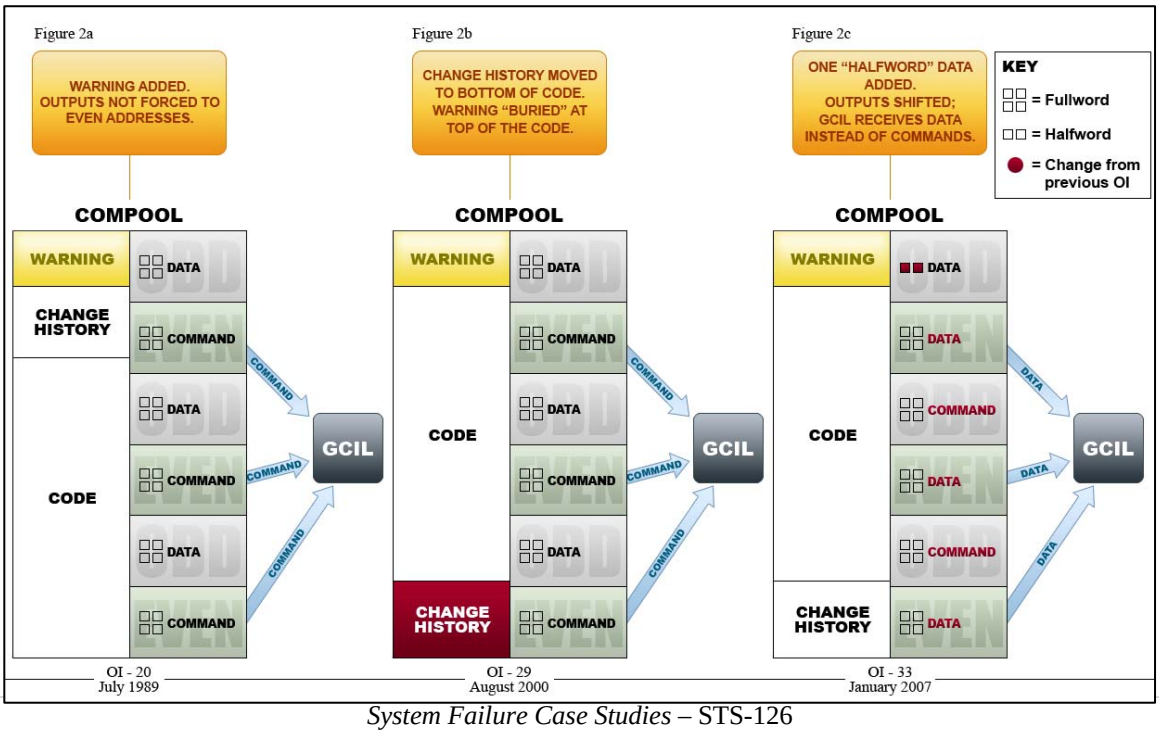

Figure 3: the STS-126 Crew

Proximate Cause

The S-band/Ku-band handover and PSP port shift did not happen automatically because the GCIL received incorrect commands from the Systems Management computer. The SM computer sent the wrong commands because software changes implemented during OI-33 shifted output command data within the compool from even addresses to odd addresses. Due to software changes introduced during OI-20 in 1989, the output commands relied on proper fullword alignment.

Underlying Issues

Imprecise Standards - Traps Set by Historical Changes to the Flight Software

A programming standard required programmers to lock compool data to fullword addresses, but the standard was unclear and programmers were not held accountable to the standard. While they recognized the danger in their technique enough to include warnings in their code, the OI-20 development team did not lock down command outputs. Avoiding data shifts was a recognized "good practice," but it was not formally documented. The programmers did not have sufficient training to fully appreciate the potential effects of inserting or deleting data in the compool. Even if formal training had been provided to address these issues, there were no items in the inspection checklist to remind programmers of specific impacts to consider while developing or inspecting the code. The inspection checklist was used to confirm that the code had been reviewed for compliance with the programming standards. The software standard has since been revised to clarify the necessity of using fullword alignment and locking down data outputs.

Scope of Flight Software Monitoring in the Software Production Facility

The Software Production Facility did not support closed-loop modeling to test the pertinent switching commands sent to the GCIL during OI-33. Level 6 testing at the Software Production Facility confirmed that the GCIL received SM computer outputs, but tests did not verify that these outputs were correct. Output commands were not verified because the OI's code modifications did not directly impact the commands. Testers would have needed to take extra steps to detect the command problems on STS-126 during the development process. Levels 7 and 8 did not catch the command problems, either, because the end-to-end testing had not been updated to include several new or modified functions.

SAIL Testing: PSP Port Moding

With some additional set-up, the Shuttle Avionics Integration Laboratory (SAIL) could have tested the switch from RF to umbilical, but this test was not identified as an essential test for OI-33. The STS-126 payload only used the umbilical. During STS-126, the PSP initialized with the RF link selected and the different switch configuration did not force an override to the umbilical. Fortunately, the switchover was only required once during flight, and the ground crew was able to manually direct the shift.

SAIL Testing: S-band/Ku-band Handover

Testing at the SAIL uncovered what hindsight seemed to indicate were clear indications of the S-band/Ku-band handover problem, but the nature of the tests led the original test team to misinterpret these issues, and no discrepancy reports were filed.

During a setup step for the only formally verified handover, the IAV team manually initiated a handover when it did not occur automatically. They did not report this anomaly because this handover was not a verification requirement and there was a history of problems with this handover due to lab setup issues. This test was later suspended after the handover that was to be formally verified failed, but the handover was successful after the test was restarted. This handover success was a false verification; the test restart reset the GCIL so the command was transmitted successfully. A third automatic handover failure occurred later in this test, but this failure was not noted because it was not in the section of the test verifying the handover.

None of the S-band/Ku-band handover failures were documented in SAIL anomaly reports. Three factors may have influenced this oversight: the misleading success of the one S-band/Ku-band handover that was required for verification, the belief that OI-33 did not impact handover logic, and a history of handover problems unrelated to the flight software.

Aftermath

The flight software team isolated and confirmed the software problems within a few hours of recognizing the anomalies. The STS-126 mission was not affected, and the team implemented a patch to correctly align the affected output commands for future missions. Corrective and preventive actions have been identified, and while some actions were still in progress as this study was published, those that could impact STS-119 have been implemented.

For Future NASA Missions

STS-126 illustrates the need to ensure critical elements are embedded in design and procedures, provide sufficient training, complete rigorous end-to-end testing and verification, follow the oft-quoted mantra, "Test as you fly," and find the real causes of all test anomalies.

Specific Procedures and Training

Programmers unknowingly violated the intent of flight software programming standards during OI-20. The standards specified that outputs should be forced to fullword alignment, but standards did not forbid data changes within the common data pool that could affect fullword alignment. Avoiding such data shifts was recognized as a good practice, but the practice was undocumented. "Good practices" that are critical to mission success must be documented in procedures and formalized in training.

End-to-End Verification

Incomplete end-to-end verification may mask important errors that are not identified during function-specific tests. Although this type of simulation was anticipated later during integrated avionics verification at SAIL, a closed-loop test during development and verification may have identified the error earlier on in the process when specific affected functions were tested. Even in situations where individual parts may function as required, it is vital to ensure that transitions between functions occur properly. Although functions may seem completely unrelated, lack of end-to-end verification can have significant consequences when dealing with tightly connected, complex systems.

Test—and Practice—as You Fly

Although getting as close as possible to actual mission conditions may be difficult, there are measures that can be taken to close the gap between simulation and reality. IAV testing at SAIL did not have a documented requirement to test all the functions affected during STS-126. Some subtle differences between testing and flight conditions masked the problems that became apparent during the mission. The SAIL test configurations masked the PSP port moding problem. And by not requiring verification of the S-band/Ku-band handover in both directions, in addition to a GCIL reset that would not happen in flight, tests also missed the communication handover problem.

These gaps in testing show that even slight differences between testing and actual flight configuration and sequences can conceal important errors. Test procedures should make every effort to mimic flight conditions.

The PSP port moding function and the S-band/Ku-band handover was not exercised during STS-126 training simulations or vehicle processing. Although the Shuttle Mission Simulator is a training facility and not a formal verification/test facility, it provides an additional opportunity to uncover flight software problems during extra hours of software run time. Had the affected functions been exercised during simulator training, the problems with the flight software may have been found.

Anomaly Documentation

This close call indicates that test anomaly reporting was less rigorous than expected. Personnel did not document test deviations because they falsely assumed the test failures were familiar lab issues and were not related to the functions being tested at the time. Had the Ku-band/S-band handover failures been documented in discrepancy reports, the reports would have led to further analysis and evaluation. In mission critical testing, all test failures should require some level of follow-up. Although a failed portion of the test may seem unrelated, it's important to always take investigation of any anomaly at least one step further.

Similar Cases

WIRE System Failure Case Study ("Cover Blown," SFCS 3.2) discusses another test anomaly that misled the team. Both WIRE and SOHO ("Million Mile Rescue," SFCS 3.2) experienced hardware failures tied to latent conditions similar to those that contributed to the STS-126 anomalies.

References

This case study is based on communications with personnel involved in STS-126 and on the documents below:

"Flight Software Readiness." *STS-119 Joint Shuttle/Station Flight Readiness Review*. United Space Alliance Presentation, 02/03/09.

"Space Shuttle Orbiter Systems." *HSF-The Shuttle*. 04/07/03. Accessed 03/10/09. < http://spaceflight.nasa.gov/shuttle/reference/shutref/orbiter/>

Fishman, Charles. "They Write the Right Stuff." *FastCompany.com*. 1996. Accessed 03/10/2009. <http://www.fastcompany.com/magazine/06/>

ACKNOWLEDGEMENTS: Special thanks to John Magley, John Marinaro and Karen Meinert for their insightful peer reviews.

NASA SAFETY CENTER

Executive Editor: Steve Lilley steve.k.lilley@nasa.gov
Developed by ARES Corporation