Cassini Remote Terminal Interface Units as Part of Instrument Test Bed Operations During the Tour of Saturn

M.A. Sarrel, S.C. Clancy,
A.J. Runkle, L.E. Tatge, R.E. Lin

Jet Propulsion Laboratory,
California Institute of Technology,
Pasadena, California, 91109, USA.

Abstract

This paper will describe the ways in which the Cassini Remote Terminal Interface Units (RTIUs) were upgraded and enhanced to support flight operations during the Cassini tour of Saturn. The Cassini RTIUs send commands from the instrument's ground support equipment computer to the instrument's engineering model and sends telemetry back. They are very simplified simulators of the Cassini spacecraft's Command and Data System (CDS). The original RTIUs were built before launch and used to test the instruments during Assembly, Test and Launch Operations (ATLO). Across the project, there were three different implementations of RTIU.

The original intent of the RTIU upgrade was to get the instruments on to a common RTIU implementation and to update the hardware. The original hardware would become more difficult to maintain over the course of the lengthy Cassini mission. And, because of its slower processors, it could not support some of the higher spacecraft telemetry rates. But, it became clear that the usefulness during the tour of Saturn of the instrument test beds could be enhanced if new features were added to the RTIUs.

The new RTIU implementation can verify sequences of commands from the Cassini uplink process. The command file is translated using the Cassini uplink software into a format that the RTIU can understand and replay to the instrument engineering model at a remote site.

Through a similar translation script, the new RTIU can replay a test performed the Cassini Integration and Test Laboratory (ITL). The ITL contains the best ground model of the Cassini CDS and Attitude and Articulation Control System (AACS) flight hardware and software. This feature reduces the need for instruments to bring their engineering models to JPL for testing. To speed the testing of long sequences, the new RTIU can both skip and repeat sections of a command sequence. Also, the Cassini CDS recently created a library region on the spacecraft to store Instrument Expanded Block (IEBs) on the spacecraft so that they did not need to be up linked from the ground as frequently. We emulated the same feature in the RTIU.

The requirements on the RTIU during the tour phase of the mission are different from those during the pre-launch ATLO phase. During ATLO the phase, the focus is on building and integrating instrument hardware and software. During the tour phase, the focus changes to designing, building and testing sequences of commands for scientific purposes. The RTIU must also support activities such as instrument anomaly resolution, new flight software testing and verification and validation. The features we have added to the new RTIU are designed to better integrate the RTIU into tour processes and with tour requirements.

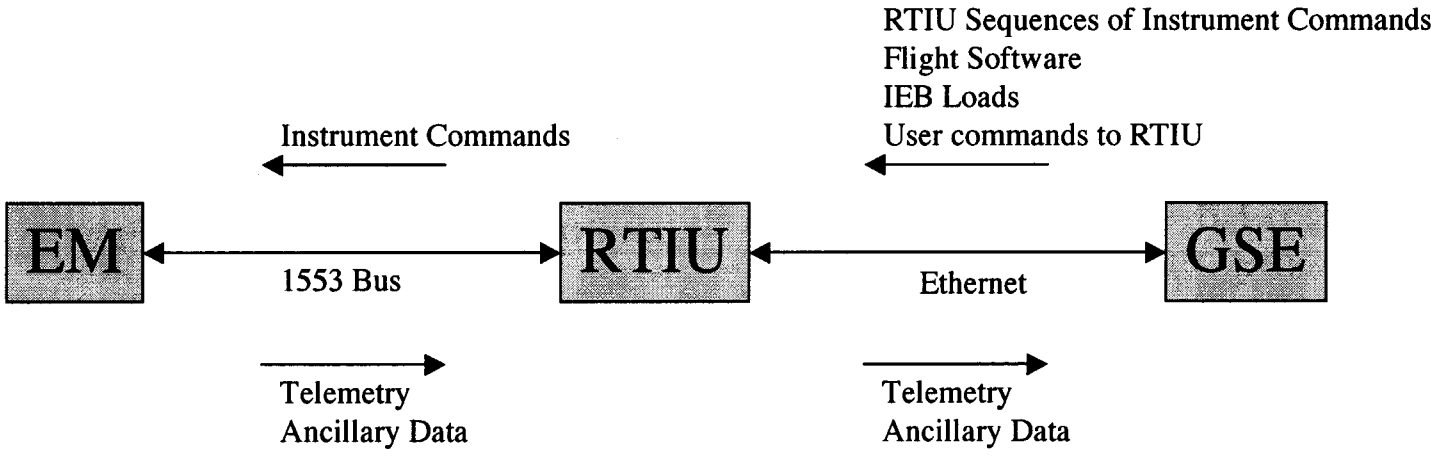
Reasons for Upgrade

- Before upgrade, we had three different implementations of RTIU with different capabilities.
- No plan for hardware replenishment
- No plan for software maintenance.
- Other benefits, like improved performance and capability came later.
- Cassini has very long mission (3-4 yr pre-launch, 7 yr cruise, 4 yr tour, TBD yr extended mission).

Pre-Launch vs Post-Launch

- Pre-Launch
  - Build/test instrument hardware.
  - Integrate onto spacecraft.
  - Build/test instrument flight software.
  - Build/test instrument ground software.
- Post-Launch
  - Design/test sequences for Science Planning Virtual Team.
  - Design/test sequence for Sequences for Sequence Virtual Team.
  - Design/test Instrument Expanded Blocks.
  - Resolve Instrument Anomalies.
  - Test changes to command database or telemetry modes.

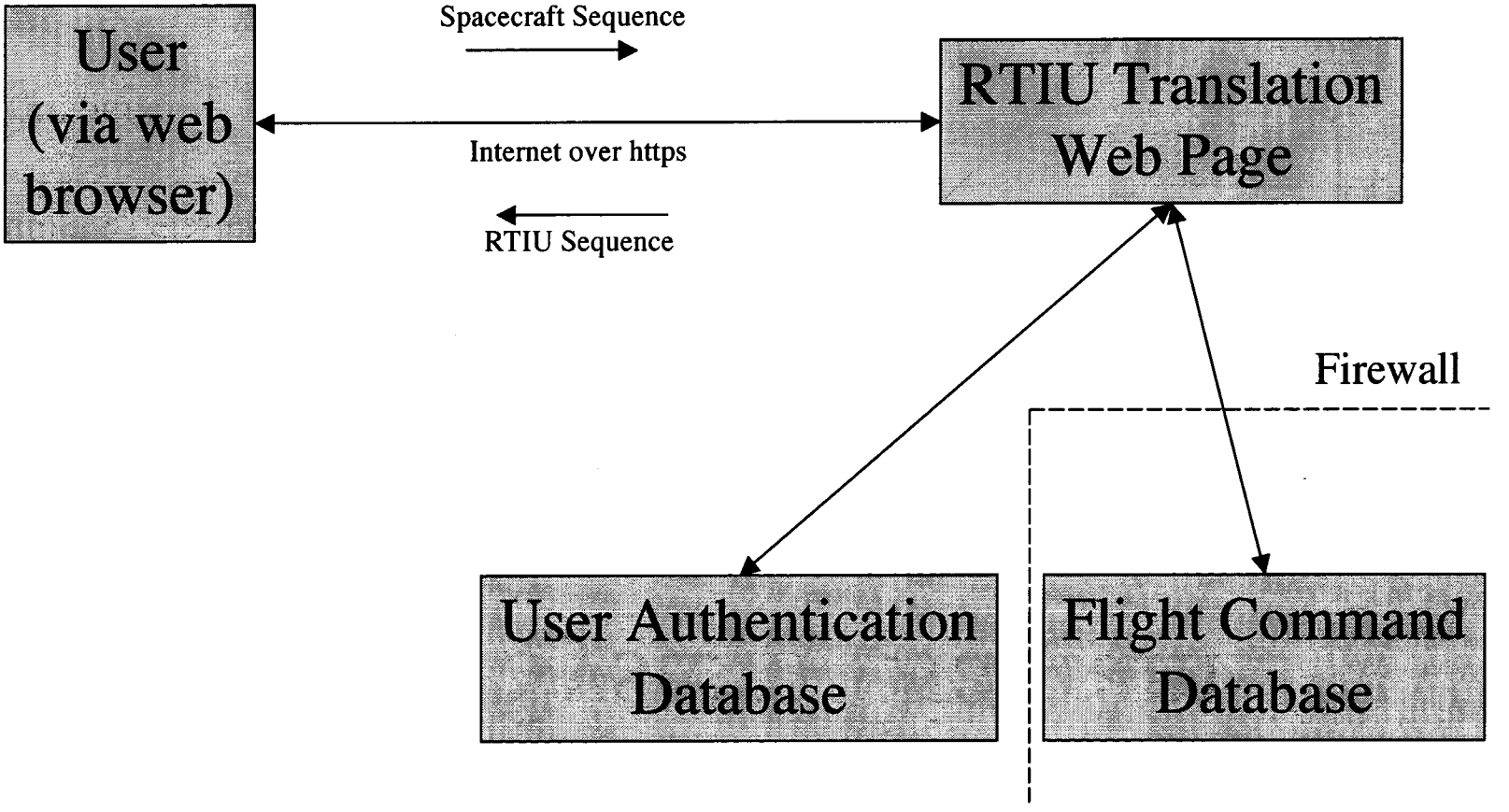
RTIU System Diagram



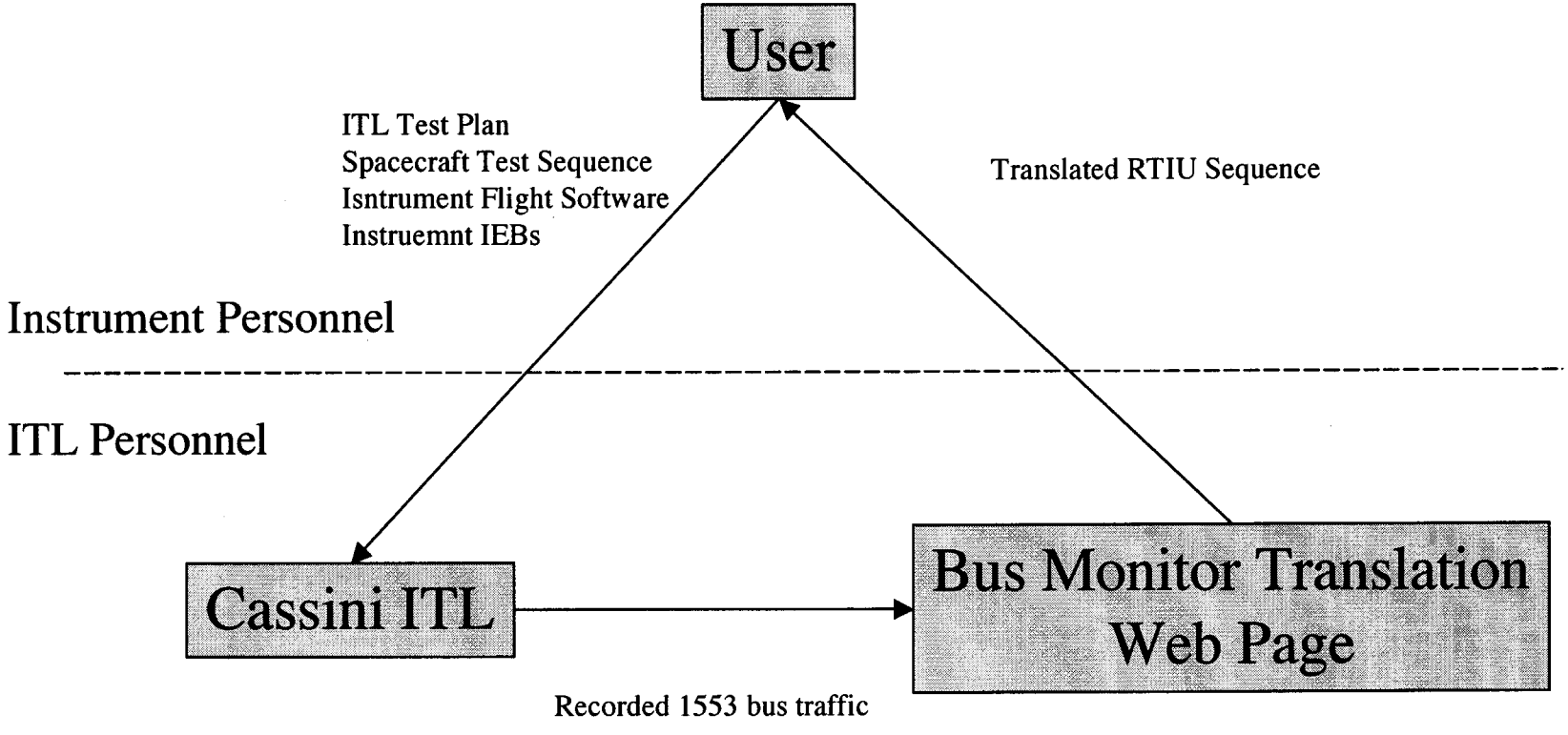
Components of the System

- Hardware
  - Motorola VME Chassis
  - Motorola PPC Motherboard
  - Systran 1553 Interface Card
- Software
  - Custom RTIU software that runs on VME hardware.
- Web pages
  - Translate spacecraft sequences into hex-based sequences in RTIU format.
  - Translate spacecraft sequence into hex-based RTIU Instrument Expanded Blocks in RTIU format.
  - Translate recorded bus monitor data from Cassini Integration and Test Lab into hex-based RTIU sequence.

RTIU System Diagram



RTIU System Diagram



New Features for Tour Operations

- Verify actual command sequences translated by the same software used for flight.
- Replay sequences of commands as run in the Cassini Instrument Test Laboratory with exactly the same content and timing.
- Simulated Instrument Expanded Block library region as on spacecraft.
- Load instrument flight software with same (or accelerated) rate.

New Features for Tour Operations

- Collect and report ancillary data as broadcast by instrument engineering model.
- Allow multiple instruments simultaneously (useful for ISS which appears on the spacecraft bus as two instruments).
- Emulation modes for backward compatibility with instrument ground support software designed to work with one of the three previous implementations of RTIU.

New Features for Tour Operations

- Optionally add ground headers to facilitate compatibility with instrument ground software.
- Windowing to allow parts of a sequence to either be repeated or skipped (good for long sequences).
- Allow simulation of timing jitter on 1553 bus.

System Tradeoffs

- The Cassini RTIU is a special-purpose piece of hardware and software.
- It is Cassini-specific.
- Does a limited set of things:
  - Send commands
  - Collect telemetry
  - Execute and manage sequences of commands
  - Simulate a few Cassini-unique CDS features

System Tradeoffs

- Designed as drop-in replacement for existing RTIUs.
- Does not attempt to do any data display.
- Requires instruments to have separate software for telemetry display and analysis.

Benefits

- Reduced need for instruments to request testing time in Cassini Instrument Test Laboratory.
- More efficient use of that ITL time.
- Better fidelity of translated sequences.
- Increased ability to diagnose or rule out instrument problems.
- Easier regression testing of ITL tests.
- Ability to simulate all spacecraft data rates.

Conclusions

- Don't build and forget. This applies to any software-based system that must last many years.
  - Changing operating systems
  - Changing security environments
  - Hardware replenishments
  - Bug fixes
  - New feature requirements
- Several instruments took the opportunity to upgrade their ground support equipment software.

Conclusions

- Allowed one instrument to get rid of a custom-built timing board along with their old RTIU.
- System includes more than just RTIU software and hardware, but various web pages and integration with project processes.
- On-going hardware and software maintenance is important
  - Even when primary development ends, must have enough funding to keep software development viable.
  - Need to stock spare hardware and have maintanence plan.