Space Computing Systems Validation Challenges

ITI Workshop
Coordinated Science Laboratory
University of Illinois

Raphael R. Some
New Millennium Program Technologist
Jet Propulsion Laboratory
California Institute of Technology
Introduction

• Challenges of Spaceborne Computing Systems
  – The Good, The Bad, The Ugly

• Validation Approaches
  – Past, Present, Future

• Some Thoughts
1958
First U.S. satellite

Explorer 1

The Good: Small, Simple, Robust – It Worked!
Over 50 NASA Missions Currently Flying

- **Spitzer** studying stars and galaxies in the infrared
- **GALEX** surveying galaxies in the ultraviolet
- **Ulysses** studying the sun
- **Aqua** studying Earth’s oceans
- **Aura** studying Earth’s atmosphere
- **Cassini** studying Saturn
- **CALIPSO** studying Earth’s climate
- **Two Voyagers** on an interstellar mission
- **Chandra** studying the x-ray universe
- **Hubble** studying the universe
- **Mars Odyssey**, rovers “Spirit” and “Opportunity” studying Mars
- **MESSENGER** on its way to Mercury
- **New Horizons** on its way to Pluto
- **QuikScat**, Jason-1, CloudSat, and **GRACE** (plus ASTER, MISR, AIRS, MLS and TES instruments) monitoring Earth.

The Bad: Complex Expensive Systems, Severe Environments, Remote Locations, No Second Chances – Sometimes They Work,… Sometimes Not So Good
The Ugly
(Significantly more severe than Earth orbit)

• High Radiation
  – Mrads and GeV

• Extreme Temperatures
  – -270 deg F on Europa to >900 deg F on Venus,
  – >1000 cycles of 100 deg on MER (Mars)

• Vibration
  – Launch, Planetary Entry, Descent, Landing, Roving, Quakes, Impacts, Turbulence

• Power
  – <100W (typically <50W) available for computing

• Mass
  – < 10kg available for computing

• Low Error Tolerance
  – Navigation, Automated Operations, Communication, Deployments
More Ugliness

• One-Off Systems in a Cost/Schedule Constrained Environment
  • Hardware:
    – Theory: Legacy, Rad Hard, Fully Qualified, Thoroughly Characterized, Tested and Validated
    – Reality: Complex COTS and Custom Parts, Minimal Characterization and Test Possible (Current DRAMs have upwards of 60 modes of operation)
  • Software:
    – Theory: Software Fixes All Ills
• Often Can’t Test Final System Until It’s Flown
  – Realistic Space/Mission Environment Unavailable On The Ground
  – Software Not Available Until After Launch
• Next Gen Systems Need COTS Multicore Machines, Low Power, High Performance Parallel Processing: Science Data Processing (not just compression) and Autonomy (not just automation)
Flight and Ground Software Anomalies
(It’s Not Getting Any Better!)

Chart Courtesy of: Martin Feather, Al Nikora
Some Examples

• **Software:**
  – Mars Climate Orbiter (Mars ’98) – km vs miles
  – MER – buffer overflow
  – Arianne V – 64b->16b conversion register overflow
  – Cassini – command sequencer buffer size and command concatenation/reconstitution

• **Hardware:**
  – Galileo Antenna Deployment
  – Cassini Memory
  – ST5 Memory
  – MER FPGAs
The REALY Ugly

- Ground Based COTS Systems Are Not Immune
  - Neutron Induced SEU’s reported at 250nm node
  - Alpha Induced SEUs reported at 65nm node
  - COTS Supercomputers in benign lab environments require fault tolerance due to MTTF of SOTA COTS components
  - Hardware Companies are Incorporating Fault Tolerance Into Their Processors and Support Chips To Reduce But Not Eliminate The Problem
  - Some Hardware Companies are Starting To Look At Hardness By Design Techniques (radiation, noise, thermal, mfg defects,…)
  - The Issue Is No Longer “will it upset?”, But “what upset rate won’t be noticed”
  - COTS Software – Unreliable and Opaque
  - Current Software Schedules/Budgets/Failure Rates are Unacceptable
  - System Failures are Endemic
    - Accepted As Normal and Unavoidable
Validation Approaches
Past

- Gross level radiation testing of critical components
- Standard Shake & Bake of Subsystems & Systems
- Unit and Build Testing of Software
  - Simple RTOS used as a ground commanded sequencer
  - Extensive testing on ground based simulator
  - Success oriented testing of normal ops
- Extensive code walk through, and testing on simulators of operational sequences
  - Success oriented testing (does it work in expected scenarios)
- Extensive operator and engineer participation in every aspect of operation, close monitoring of sequence execution, quick human reaction to problems
- Bottom line:
  - Simplify system, test spec’d scenarios, count on human ingenuity and hope for the best
Validation Approaches Present

• Similar to Past With Some Additions:
  – Occasional Board Level Hardware Rad Test Using Custom Test Software
  – Occasional Software/System Model Based Validation (eg. Spin)
  – Occasional Software/System Formal Methods Based Validation

BUT

  – Model and Formal Methods Based Validation Difficult With Large Complex Systems
  – Still Require Significant Engineer Involvement in Operations
  – Still Find Errors in System and Application Codes, and Unanticipated Hardware Faults during mission ops
Thoughts On Validation Approaches For Future Systems

• **Assertion:** The Validation Problem Can Not Be Solved in the V&V Domain
  – The Fundamental Issue is Minimization, Knowledge and Control of State Space
  – To Achieve System Validation, The State Space Must Be Constrained
  – Once Constraints Are Placed On State Space, Automated Methods Can Be Applied

• **Need a New Design/Test/V&V Paradigm (here’s one possibility)**
  – Understand The Problem:
    • Extensive characterization of detailed component fault set/rates
    • Fault/Error propagation model
  – Fault Tolerance built into all systems/hardware/software
    • Supported by models, tools and automation at the design level
  – Automated formal methods and model based validation of code segments and system operational modes to the extent possible.

  – **Sequencer Based Software Design/Implementation**
    • Standardized constructs and implementation rules
    • Standardized representations and abstractions
  – Software JTAG Bus
  – Automated Exhaustive Test Vector Generation and Test Execution
    • Fault Injection Testing Using Fault/Error Models

  – Board level system radiation (and other environmental stresses) testing with operational software and realistic worst case system operational scenarios
  – Random Unstructured System Test in realistic (simulated) system environment