Investigating Model-Based Autonomy for Solar Probe Plus.


Bill Van Besien
Flight Software Engineer

(Notice: This presentation does not contain export controlled information)
Contents

- Mission Profile
- Scope of Autonomy; Challenges in the Solar Environment
- Overview of Rule-Based (RB) System
- Motivations for a Model-Based (MB) Approach
  - Software technology demonstrations
- Comparative Analysis of MB and RB for SPP
- Influence and Future Prospects
Executive Summary

- Decision to *proceed* with Rule-Based Autonomy system
  - Extensive and successful operational history.
  - Substantial re-use from previous missions.
  - Other program-specific constraints.

- Model-Based technology to other applications
  - Model-based system planned to be used on other upcoming spacecraft and UAV projects.
  - Benefits of this model-based approach to be used in continued development of SPP autonomy engine.
Mission Profile / Science Objectives

- Significantly contribute to “our ability to characterize and forecast the [solar] radiation environment”
  - *Structure and dynamics* of the solar magnetic fields.
  - Tracing the *flow of energy* that heats the corona.
  - Understanding the mechanisms and flows of energetic particles.
  - The “dusty plasma” phenomenon.
Mission Profile / Spacecraft

- Trajectory/Mission Plan
  - Two dozen orbits, gradually brought in

- Propulsion
  - 3-Axis Stabilized, no deep-space maneuvers

- Thermal Protection System (TPS)
  - Heat difference of up to 2000 C

- Solar Arrays
  - Slew deployment, carefully prevent overheating

- Instruments
  - Magnetometers, particle detectors, imagers, etc…
Mission Profile / Autonomy Challenges

- Communication eclipses
  - Up to 34 days, cumulatively, throughout orbit
  - Very low emergency downlink rate

- Primary drivers for on-board Fault Protection
  - Maintaining TPS pointing
  - Avoiding solar array overheating

- Autonomy must recover into operational state during thermal-critical regions

- Essential Requirements
  - Manage design complexity
  - Execute predictably and robustly
  - Provide high levels of verifiability
Autonomy System Evolution

- **Generation Zero**
  - Early science missions; No notional separation of Autonomy.

- **Generation 1**
  - (ACE) Monitoring of single telemetry points.

- **Generation 2**
  - (NEAR, TIMED) More expressive conditions.

- **Generation 3+**
  - STEREO, MESSENGER, RBSP/VAP
  - More expressiveness; Notions of CT, Storage Vars, etc…
Rule-Based Autonomy

- Single-fault tolerant
- Allocations for dozens of...
  - RPN Expressions
  - Computed Telemetry
  - Storage Variables
- Fine-grained control and manipulation
# Rule-Based HTML X-Reference Doc

<table>
<thead>
<tr>
<th>Rule Number</th>
<th>Title</th>
<th>Rule Premise</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (Rule DB # = 003)</td>
<td>sLVS in EA</td>
<td>( \text{OBS MODE} == \text{MODE EA} ) &amp; &amp; ( \text{MAIN BUS VOLT EU} &lt; 27.0 &amp; &amp; \text{PS MAIN BUS VOLT ST} == \text{MUX AD OK} )</td>
</tr>
<tr>
<td>3 (Rule DB # = 023)</td>
<td>sCLT (Not EA)</td>
<td>( \text{OBS MODE} != \text{MODE EA} &amp; &amp; \text{SCLT TIME OUT HRS} &gt; 60.0 )</td>
</tr>
<tr>
<td>25 (Rule DB # = 032)</td>
<td>Monitor HGA Gimbal Invalid</td>
<td>( \text{OBS MODE} != \text{MODE EA} &amp; &amp; \text{GC HGA_CNTL ST} != \text{HGA_CNTL DISABLED} &amp; &amp; \text{GC HGA PNTG AT EARTH} == 0 )</td>
</tr>
<tr>
<td>28 (Rule DB # = 039)</td>
<td>Persistent ST Fault in OBS or STBY</td>
<td>( \text{OBS MODE} == \text{MODE STANDBY}</td>
</tr>
<tr>
<td>37 (Rule DB # = 033)</td>
<td>TWTA to Transmit when Battery OK (Not Ascent)</td>
<td>( \text{OBS MODE} != \text{MODE ASCENT} &amp; &amp; \text{TWT VOL EU} &gt; 5.0 &amp; &amp; \text{RF EPC ST} &gt; \text{0x1A} &amp; &amp; \text{RF TWT ST} &gt; \text{0x1A} &amp; &amp; ( \text{BATT PRES1 EU} &gt; 650.0 &amp; &amp; \text{BATT PRES1 ST} == \text{MUX AD OK} )</td>
</tr>
<tr>
<td>38 (Rule DB # =)</td>
<td>TWTA in Idle Mode (for Launch + Not Ascent)</td>
<td>( \text{OBS MODE} != \text{MODE ASCENT} &amp; &amp; \text{TWT VOL EU} &gt; 5.0 &amp; &amp; \text{RF EPC ST} &lt; \text{0x1A}</td>
</tr>
<tr>
<td>39 (Rule DB # =)</td>
<td>TWTA in Off Mode (for Launch + Not Ascent)</td>
<td>( \text{OBS MODE} != \text{MODE ASCENT} &amp; &amp; \text{TWT VOL EU} &lt; 5.0</td>
</tr>
<tr>
<td>41 (Rule DB # = 040)</td>
<td>High Temp Batt Discharge (Not EA)</td>
<td>( \text{OBS MODE} != \text{MODE EA} &amp; &amp; \text{BATT CURR1 EU} &lt; \text{0.0} &amp; &amp; \text{BATT CURR1 ST} == \text{MUX AD OK} &amp; &amp; ( \text{BATT TEMP1 EU} &gt; 18.0</td>
</tr>
<tr>
<td>118 (Rule DB # = 201)</td>
<td>Monitor HGARA Temperature</td>
<td>( \text{OBS MODE} != \text{MODE EA} &amp; &amp; \text{HGARA TNP EU} &gt; 55.0 &amp; &amp; \text{TRIO BS ACK ST} == 0 )</td>
</tr>
</tbody>
</table>
Roughly Equivalent Model-Based View
Principles of Autonomy System Design

- **Understandability**
  - Necessary for reviews.
  - Essential for future modifications.

- **Flexibility**
  - Speeds development and testing.
  - Eases the burden on ops staff.

- **Verifiability**
  - Prevent crunch in I&T testing.
  - Ensure risk level.
Model-Based Motivations – Testing

- 2008 NASA Fault Management Workshop findings

- Finding #1 – The “Downstream” Testing Crunch
  - Late testbed availability led to rapid spending growth during I&T.

- Finding #4 – FM Representation and Design Guidelines
  - Lack of sufficient formalization in FM design and documentation.
  - Recommendation: Identify representation techniques to improve FM system design and review.
Motivation: Design-as-Implementation

- **Understandability**
  - *The design is the implementation.*
  - Design not subject to interpretation.
  - Intuitive understanding of the autonomy system behavior.

- **Flexibility**
  - Ability to manipulate, alter autonomy model behavior on-the-fly.

- **Verifiability**
  - Ability to test early *without* testbed integration.
  - Graph-based foundation amenable to formal verification methods.
Core concepts developed FY 06 – FY 09 PI George Cancro

Based upon Bell Labs Virtual Finite State Machine (VFSM)

ExecSpec Software Suite

• Design Tool (ESD)
  Intuitive visual programming for state model logic through diagrams

• Interpreter (ESI)
  Interprets and executes diagrams

• Visualizer (ESV)
  Monitoring tool to provide situational awareness
Why not Model-Based Auto-coders?

- Similar COTS offerings exist, why not use them?
  
  “[COTS alternatives] do not provide the end-to-end flexibility and operations monitoring capability necessary for next generation autonomy development systems”

- Desire to separate “interpreter”/engine from autonomy model
  - Simpler to alter or upload new models.

- Surveyed alternatives can not support CONOPS
  - MOPS, command and telemetry infrastructure.
  - Run-time manipulation of models, inputs.
  - No separate designer, visualizer, interpreter.
Investigating Suitability for SPP

- Early Technology Development
  - Concept-development IRADs on STEREO

- SPP Suitability Investigation
  - SPP FSW integration prototype
  - Tech readiness demo on UAV
  - Formal verification IRAD
  - Trade study
1 – ExecSpec Concept of Operations

Visual Development & Test Environment (ESD)

Diagrams

Telemetry to animate Functionality during Operations

Real-Time Embedded Interpreter (ESI)

Data from Vehicle

Decisions (Domain-Specific Commands)

Embedded System
2 – ExecSpec Designer Overview
Modeling STEREO
3 – Early Work on STEREO Testbed

- STEREO autonomy system *translated* into 43 state machines.
- ExecSpec interpreter inserted into STEREO flight software.
- ExecSpec visualizer/designer inserted into ground system.
- Demonstrated in hardware testbed our model-based software handing *most* STEREO fault management requirements.
**Requirement:**

**Safety:** “Never radiate while swapping antennas”

AG !(twta=radiating & ant=swapping)
Formal Verification Study

- “Translated” STEREO autonomy system into a model-based conception
- Developed *ExecSpec*-to-*NuSMV* compiler
  - Assumptions → Plant Models
  - Significant interactions across system
- Proved critical safety constraints
  - Introduced faults, confirmed detection
  - Order of seconds
Objective: Demonstrate performing critical in-flight fault management in challenging environment for UAV platform.

Approach
- Develop FM design
- Integrate in UAV FSW environment
- Establish and demo CONOPS

Flight tests
- Override in-flight autopilot under anomalous conditions.

Successful Demonstrations
- First - “Unicorn” UAV in MD.
- Second – Commercial UAV on West Coast.
5 – Trade Study

- Formal, comparative analysis of both approaches

- “Null Hypothesis” – \( H_0 \): Rule-Based autonomy suitable for SPP.

- Question – Does model-based scheme provide a substantially compelling case to overrule \( H_0 \)?

- Concerns – Risk Management
  - Limit additional new technology on SPP
  - Leverage software re-use

- Approach
  - Several dozen metrics to compare schemes
  - Each with “suitability score”
Comparative Metrics (1)

- Re-initialization speed
  - Frequent processor rotation is expected

- Managing subsystem interdependencies
  - Complex system with many interactions

- Intuitive, visual system for reviewing designs
  - Model-Based on top for “design”
  - No clear benefit for either for mission ops

- Designing sequences of several decision points
  - Complex responses requiring checks of telemetry while in progress
Comparative Metrics (2)

- Path dependent responses
  - When path to fault is as important as fault itself.
  - Finer “granularity” to responses.

- Similarity of design to implementation
  - Reduces cost and risk; Review of design becomes review of implementation

- Facilitation of formal verification
  - Important, but not expected to replace testing, so will add additional setup cost
  - Model-Based, but potential for rule based

- Efficiency of implementation
  - Speed of rule evaluation, time budget, non-volatile footprint
Comparative Metrics (3)

- Parallel development
  - Well established process in RB approach
  - Merging process not clear in MB approach

- Starting point for testing and implementing logic
  - Can start preliminary testing and implementation sooner

- Past Mission Experience
  - Previous successful expertise valuable during all mission phases
  - FM Working Group finding – switching autonomy paradigms can be problem-prone
Evaluative Conclusions

- Model-based, rule-based systems *equivalent expressiveness*

- By formal metrics, marginal benefit for MB approach for SPP
  - However, not sufficiently compelling to overcome motivations to remain with RB approach

- Post-investigation plans
  - Proceed with Rule-Based autonomy system
  - Elements of MB software to be leveraged for rule-based system
Reusable FSW Architecture in Context

**Diagram:**
- **OS Abstraction, CFE Services, CFE Messaging plus APL Libraries**
- **VxWorks OS Layer**
- **Processor Hardware & SSR Memory**
- **cPCI Interface**
- **Spacecraft Interface Card**
- **SSR Card**

**Software Legend:**
- **New for SPP**
- **C&DH Applications Reused from RBSP**
- **CFE Middleware (GSFC), Reused from RBSP**
Future Developments

- UAV Systems
- Cyber security
- Future small-sat space missions

- Rule-Based System
- Maximize tech and s/w reuse
- Leverage some model-based benefits
Sources, References, and Further Reading

- This presentation is an overview of the contributions made by:
  
  Justin Thomas, George Cancro, Russell Turner, Paul Rosendall, Eli Kahn, Eric Melin, Mike Pekala, among others. Thank you!