Graphical Model-based Software Development at the Space Vehicle Integration Laboratory

Michael J Phillips
Lockheed Martin Space Systems Company
November 5th, 2009

3rd Annual Workshop on Flight Software (FSW-09) @ JPL

Space Vehicle Integration Lab Overview

Space Vehicle Integration Laboratory (SVIL)

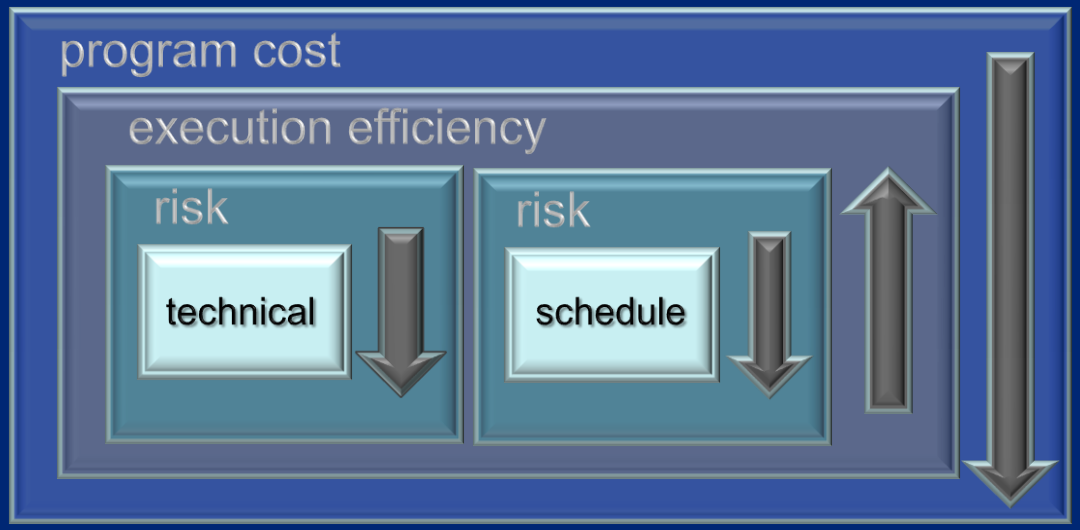
"A powerful new engineering environment for spacecraft development"

The Power of Simulation: Reducing Schedule and Cost While Improving Mission Success

- The SVIL is a space vehicle simulation and modeling environment that is primarily useful in supporting programs (study, research, development) and IR&Ds with early risk mitigation tools and strategies that benefit vehicle and subsystem component architecture, design, analysis, and trade activities.
- The SVIL also provides an environment in which early flight software architectures, designs, and prototype code can be tested against robust vehicle simulations (both non-real-time and real-time)
- The SVIL is **NOT** a substitute for program-specific test beds that satisfy unique vehicle simulation requirements (including HWIL)

SVIL Business Impacts

- Four foundational SVIL capabilities promise to increase execution efficiency and lower program costs:
  1. Model-Based Graphical Software with Autocode capability for Embedded Systems Software
  2. Low-Cost Test Beds introduced early in the lifecycle that fill a crucial gap in a suite of simulation capabilities currently on typical programs
  3. Risk Reduction testing using Rapid Prototyping with Models and Test Beds in critical risk areas
  4. Early Lifecycle Testing on heritage and new software products to gain customer confidence and mature the end software product



SVIL Support of the Program Lifecycle

Existing Program

Phase	SVIL Benefit
Requirements	• Requirements validation through early simulation • Early feedback re: requirements feasibility and specificity • Provides early look at requirements coverage
Vehicle HW/SW Architecture	• Provides early understanding of vehicle component architecture and interfaces • Early simulation of SW architecture matched to C&DH architecture provides initial eval of throughput and capacity
Vehicle HW/SW Design	• Early understanding of HW and SW design using non-real-time and real-time simulation • Thorough understanding of SW design and how it works with HW using high fidelity real-time simulation
FSW/HW Integration	• Early understanding of actual FSW integration with non-flight qualified SBC • Provides FSW credibility in areas of nominal and anomalous operational states
Support	• Support for looking at software upgrades for modular spacecraft • Support for Block programs

IR&D/Pre-proposal

Activity	SVIL Benefit
Technology Evaluation	• Provides services to support vehicle HW and SW technology studies • Simulation capabilities validate technology choices to make bids credible
Requirements Validation	• Requirements validation through early simulation • Early feedback re: requirements feasibility and specificity • Provides early look at requirements coverage
Vehicle Arch/Design Concepts	• Early understanding of HW and SW design concepts using non-real-time and real-time simulation • Provides capabilities to perform what-if analyses related to HW, SW, and interface architecture and design trades
FSW/C&DH Integration Concepts	• Provides early look at potential performance issues associated with SW/HW integration • Allows for trades on potential FSW implementation and related issues with existing/new C&DH HW

SVIL – Primary Products & Services

PRODUCTS

- **Integrated Space Vehicle Simulation Development Environment:** Enables programs to efficiently develop spacecraft flight and simulation software products
- **Space Vehicle Non Real-time Simulators:** Workstation simulation (used for development, debug, and unit test) and faster-than-real-time simulator (PC-based)
- **Space Vehicle Real-time Simulators:** Low-cost COTS-based real-time simulators capable of executing flight software (on flight-representative single board computers)

SERVICES

- **Prototyping existing and planned spacecraft configurations:**
  - Static & dynamic component & interface modeling
- **Non- & Real-time Architecture Execution:**
  - Spacecraft architecture evaluation & trades
  - Data flow, I/O bandwidth & margin characterization
  - Physical component interfaces over standard data busses
- **Spacecraft component configuration verification:**
  - Requirements, architecture, design, implementation traceability
  - Early FSW and hardware integration validation
  - Complete test data reduction and analysis

Graphical Modeling in the SVIL

Graphical models describe things like:

- Structure (interfaces)
- Behavior (functionality)
- Concurrency (timing/interactions)

Graphical representations consist of:

- Block diagrams
- Connecting wires
- Mathematical and logical constructs

Some common tools

- LabVIEW
  - National Instruments
- MATRIXx
  - National Instruments
- Simulink
  - The MathWorks
- Rhapsody
  - IBM

Some Graphical Model Capabilities

- Critical algorithm development
- Model integration on the desktop
- Development of FSW
- Definition of interfaces

Graphical Software for Simulation

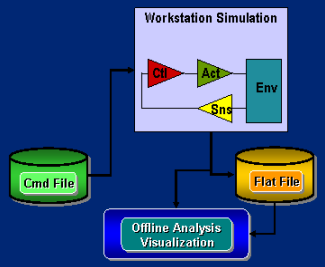
- A foundational pillar of SVIL is simulation software
  - Decision made early in SVIL concepts to include graphical model-based software
  - Models are architected specifically to develop real-time embedded software
- SVIL is primarily intended for early program simulation
  - Key benefit of simulations is an integrated look at spacecraft performance at any point in lifecycle
  - Simulation balancing act:
    - Risk: Early simulation can be challenging for programs with tight budgets
    - Opportunity: Past history has shown early simulation is critical to successful execution
- Company-wide assets are available for all programs to use, but…
  - Fact: programs still require particular test beds for qualification on actual flight software targets
  - Result: SVIL generic simulators decrease in utility as program moves through life cycle

Early simulation methodologies are mature and provide proven efficiencies

The SVIL Provides Multiple Simulation & Development Environments

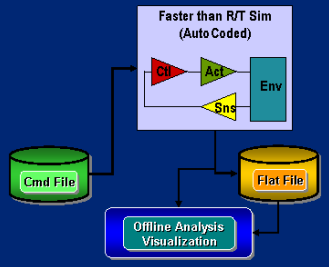
Workstation (non-RT) Simulation

- Model-based software development environment
  - Native environment for model-development and reviews
- Full spacecraft (subsystems) simulation
  - Environment can be tailored



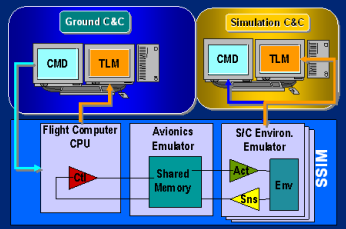
All Software Real-time Simulation

- Compiled version of Workstation Simulation
- Compile for commonly available target
- Includes the capability to execute entire FSW against the truth models in a non-real-time environment



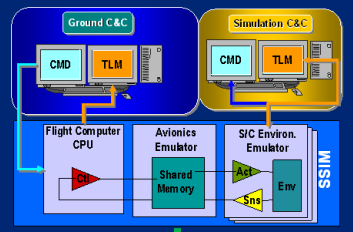
Spacecraft Simulator Test Beds

- Medium fidelity flight computer avionics test bed
- FSW executes on commercial variant of processor
- Useful for mission rehearsals, ground system integration, development of vehicle test procedures



Processor-in-the-Loop (PIL) Hardware-in-the-Loop (HIL) Simulators

- High fidelity flight computer avionics test bed
- VFS executes on flight-like processor
- HIL allows models to be replaced with actual sensors and actuators
- Used to test system component I/Fs and qualify VFS



Subsystem EDU Hardware

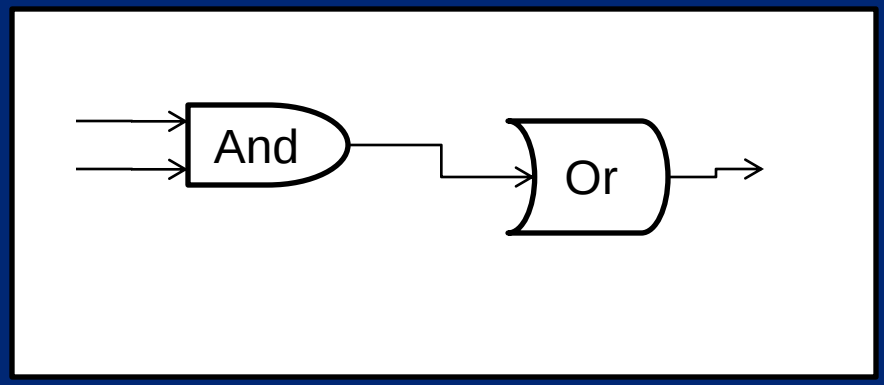
GN&C | EPS | TCS | C&DH | Payload | Comm

Variety of platforms provides many opportunities for simulation

Graphical Software as Communications Medium: Example 1

- Unambiguous requirement interpretation
  - Quick and easy to implement a model representing a requirement
  - Executable for demonstrating in meetings
  - Demonstration always results in the response: “Ahh, I didn’t think of that interpretation!”

Example: CSCI A shall inhibit/enable computation of the individually selected or all evaluations or responses on command.



Communication among engineers a continuing problem

Graphical Software as Communications Medium: Example 2

- Documenting Interfaces
  - Embed details about signals in model for developer convenience
  - Use a report generator to create ICDs between software packages that are not models
  - Use scripts to generate CM-able files to rebuild interfaces or compare against CM copies

Graphical tools enable capability for embedding important data in diagrams

Graphical Software as Communications Medium: Example 3

- Engage code-phobic engineers
  - Problem: It is challenging to get Subject Matter Expert (SME) to review and sign-off on implemented simulation software when development completes late in spacecraft lifecycle
  - Solution: Build graphical simulation software earlier in the lifecycle
    - Non-software engineers are uncomfortable with understanding software design and code products
    - Graphical Model constructs draw in the audience
      - Naturally focuses on functions and interfaces
      - Naturally opens up communication between SMEs in the room

A picture is worth a thousand lines of an ICD

Summary

- Improving communication through Graphical SW
  - Increased product quality by reducing translation and interpretation errors
  - Increased customer confidence since the design and implementation are captured in a readable, graphical format
- Early defect detection / correction through rapid prototyping
  - Increased product quality by reducing the total number of latent defects in delivered products
  - Reduction in overall costs by finding and correcting defects earlier in the program life cycle
- Risk reduction and opportunity creation
  - Decrease a program's cost and schedule variability
  - Ensure our customers that our products and solutions are sound
  - Directly lower program cost through the application of new technologies
  - Enable design evolution to create a better product

Questions?