VML Sequencing: Growing Capabilities over Multiple Missions

Dr. Christopher A. Grasso[*]
Jet Propulsion Laboratory / California Institute of Technology / Blue Sun Enterprises, Boulder, Colorado, 80302

and

Patricia d. Lock[†]
Jet Propulsion Laboratory / California Institute of Technology, Pasadena, California, 91109

I. Introduction

Virtual Machine Language (VML) is an award-winning advanced procedural sequencing language in use on NASA deep-space missions since 1997. Missions featuring VML include Spitzer Space Telescope, Mars Odyssey, Stardust, Genesis, Mars Reconnaissance Orbiter, Phoenix, Dawn and Juno. The latest deployed version, VML 2.0, features parameterized functions, conditionals, polymorphism, a rich set of control directives and data types, event detection and response, and on-the-fly creation of spacecraft commands. This feature set is used to simplify spacecraft operations and science gathering activities. A new 2.1 version is being prototyped for use as an executive within flight instruments, and may be deployed on Juno.

VML is used for a diverse set of mission functions on its various host spacecraft, including launch sequencing, daily activity loads, orbit insertion, aerobraking, entry-descent-landing, science observation, and fault responses. On Dawn, VML is used to autonomously control thrust output of the Ion Propulsion System. Generic implementations of several major uses are presented. Functional problem factoring and resource utilization are also considered.

VML is divided into three major components. The flight component exists onboard the spacecraft, allowing VML sequences to run within the flight context. The VML compiler translates human readable sequences into binary executables placed onboard and loaded by the flight component. Offline Virtual Machine is a workstation program that marries the flight component to a user interface, can run sequences at several hundred thousand times real-time, and provides a runtime behavior with 100% fidelity to the flight context. Each of these components is used in the development and deployment of sequences for flight. This paper discusses the use of these components in typical operations development processes on missions like Mars Odyssey, Phoenix, and Dawn.

Blocks are reusable relative time-tagged sequences that parameterize routine operations, and are typically packaged together into single uplinkable files called libraries. Sequences are single-use sets of instructions that run in absolute or relative time. The relationship between reusable blocks and one-use sequences is discussed. Reduced development effort due to iterative block development is outlined, along with typical development procedures. The lower cost and reduced complexity involved with creating blocks rather than flight software is noted, as is the reduction in uplink size. The ability to migrate to the spacecraft functionality that is more traditionally implemented on the ground is examined. The implications for implementing spacecraft autonomy without the need for expensive flight software agent development are discussed.

Increasingly more capable versions of VML have flown on a series of missions. The arc of VML 0, VML 1.0, VML 1.1, VML 2, and VML 2.1 is examined. Given VML's long lineage of missions and increasing capability, further simplification of operations using features in VML 2.1 is discussed. Finally, the application of lessons learned on each of the VML missions, and the incorporation of new features based on these lessons, is provided. VML is available for distribution free of charge by the Jet Propulsion Laboratory under NASA Technical Report 40365.

[*] Principle Flight Software Engineer, Blue Sun Enterprises, 1942 Broadway Street Suite 314, Boulder, CO, 80302, Professional Member.

[†] Senior System Engineer, End-to-End Information System Engineering, Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA, 91109, Member.