A SOFTWARE ARCHITECTURE
FOR AUTOMATING OPERATIONS PROCESSES

KI: VIN MILLER
Contact: SUSAN MURPHY
Operation Engineering Lab
Jet Propulsion Laboratory, MS 301-345
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91109-8099
(818) 354-1948
FAX: (818) 393-4100
susan@natasha.jpl.nasa.gov

The Operations Engineering Laboratory (OEL) at JPL has developed a software architecture based on an integrated toolkit approach for simplifying and automating mission operations tasks. The toolkit approach is based on building adaptable, reusable graphical tools that are integrated through a combination of libraries, scripts, and system-level user interface shells. The graphical interface shells are designed to integrate and visually guide a user through the complex steps in an operations process. They provide a user with an integrated system-level picture of an overall process, defining the required inputs and possible outputs through interactive on-screen graphics.

The OEL has developed the software for building these process-oriented graphical user interface (GUI) shells. The OEL Shell development system (OEL: SHELL) is an extension of JPL's Widget Creation Library (WCL). The OEL: SHELL system can be used to easily build user interfaces for running complex processes, applications with extensive command-line interfaces, and tool-integration tasks. The interface shells display a logical process flow using arrows and box graphics. They also allow a user to select which output products are desired and which input sources are needed, eliminating the need to know which program and its associated command-line parameters must be executed in each case. The shells have also proved valuable for use as operations training tools because of the OEL: SHELL hypertext help environment.

The OEL toolkit approach is guided by several principles, including the use of ASCII text file interfaces with a multimission format, Perl scripts for mission-specific adaptation code, and programs that include a simple command-line interface for batch mode processing. Projects can adapt the interface shells by simple changes to the resource configuration file. This approach has allowed the development of sophisticated, automated software systems that are easy, cheap, and fast to build.

This paper will discuss our toolkit approach and the OEL: SHELL interface builder in the context of a real operations process example. The paper will discuss the design and implementation of a Ulysses toolkit for generating the mission sequence of events. The Sequence of Events Generation (SEG) system provides an adaptable multimission toolkit for producing a time-ordered listing and timeline display of spacecraft commands, state changes, and required ground activities. The multimission SEG software is easily adapted and OEL: SHELL templates are built to meet different mission requirements. The SEG system was adapted in a unique way for the Ulysses mission since the spacecraft does all commanding in real time. The Ulysses SEG toolkit allows a user to interactively build commands on a timeline display in spacecraft event time and then the system automatically derives required ground events, builds a mission sequence of events listing, and outputs a space flight operations schedule.