

## **State Analysis: Subduing Complex Systems and Software**

Robert Rasmussen and Daniel Dvorak  
Jet Propulsion Laboratory  
California Institute of Technology

The complexity of modern space systems has reached a threshold of viability where customary methods of control are no longer affordable or sufficiently reliable. Moreover, a straightforward extrapolation of past methods has neither the conceptual reach nor the analytical depth to address the problems ahead of us. Part of the problem is system design — the interdisciplinary approach to myriad issues involved in specifying, constructing, integrating, testing, and operating a system. Divide-and-conquer strategies fail to scale in the tangled web of interactions typically encountered in complex systems. Another part of the problem is automation technology — the software and algorithms distributed across space that constitute an automated system. These systems work only as well as knowledge of system design conveyed to them allows. In both cases, constructing large, complicated systems that are reliable and affordable demands principled design discipline that goes well beyond traditional methods. This dilemma has motivated the Mission Data System (MDS), and in particular, a design methodology known as "state analysis". MDS is unified software architecture for distributed, automated flight, ground, and test systems. It is based on the notion of state, and it is organized in a manner conducive to application by systems engineers. This is accomplished by incorporating within the structure of the software a model of the system under control, expressed in terms of system state. The principle value of this approach is a synergism between systems engineering and software that directly addresses both sides of the complexity dilemma. Systems engineers benefit from a formal, rigorous approach to modeling that is embodied in the set of abstractions imposed by the architecture. Software engineers benefit from an easier, inspectable translation of system knowledge into a software form amenable to automation and complex behavior.

### *Note to Document Review:*

*This abstract, and the eventual paper, will cover material similar to a previously cleared talk, Clearance #02-1568, "Bridging the Gap Between Systems and Software Engineering".*