

**21st Digital Avionics and Systems Conference
27-31 October, 2002**

Abstract Submittal

**Authors: Paula Pingree, JPL, Pasadena, CA
Dr. Erich Mikk, self-represented, Estonia
Dr. Gerard Holzmann, Bell Labs, Murray Hill, NJ**

**Validation of Mission Critical Software Design and Implementation
using Model Checking**

Over the years, the complexity of space missions has dramatically increased with more of the critical aspects of a spacecraft's design being implemented in software. With the added functionality and performance required by the software to meet system requirements, the robustness of the software must be upheld. Traditional software validation methods of simulation and testing are being stretched to adequately cover the needs of software development in this growing environment. It is becoming increasingly difficult to establish traditional software validation practices that confidently confirm the robustness of the design in balance with cost and schedule needs of the project. As a result model checking is emerging as a powerful validation technique for mission critical software. Model Checking conducts an *exhaustive exploration* of all possible behaviors of a software system design and as such can be used to detect defects in designs that are typically difficult to discover with conventional testing approaches.

StateFlow[®] by The Mathworks was used to develop the mission critical Fault Protection (FP) flight software (FSW) for NASA's Deep Space 1 mission. Demonstrating the trend toward statechart modeling and auto-code generation, StateFlow[®] has also been adopted for the FP FSW development on NASA's Deep Impact project, scheduled to launch in 2004. Both missions share a core component of FSW for which the design has been validated using SPIN. Our aim is to validate mission-specific components of FSW that are specified using statecharts and are being translated automatically to the final flight code for the mission. We established an automatic translation procedure from statecharts to SPIN for the validation of the mission-specific components. To guarantee the compliance with the generated code our translation tool set preserves the StateFlow[®] semantics. We are now able to specify and validate portions of mission critical software design and implementation using exhaustive exploration techniques.