Prototype Implementation of a Goal-Based Flight Software Health Management Service

Matthew Barry
Kestrel Technology, LLC
4984 El Camino Real, Suite 230
Los Altos, California, 94022
Email: mrbarry@kestreltechnology.com

Gregory Horvath and David Wagner
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91109
Email: gregory.a.horvath@jpl.nasa.gov
Email: david.a.wagner@jpl.nasa.gov

Extended Abstract

The FAILSAFE project is developing concepts and prototype implementations for software health management in mission-critical real-time embedded systems. The project unites features of the industry standard ARINC-653 Avionics Application Software Standard Interface and JPL's Mission Data System (MDS) technology. The ARINC-653 standard establishes requirements for the services provided by partitioned real-time operating systems. The MDS technology provides a state analysis method, canonical architecture, and software framework that facilitates the design and implementation of software-intensive complex systems. We use the MDS technology to provide the health management function for an ARINC-653 application implementation. In particular, we focus on showing how this combination enables reasoning about and recovering from application software problems. In this sense one might consider this to be a concept of continuous software verification using dynamic analysis.

In order to make it a compelling demonstration for current aerospace initiatives, we imposed on our prototype a number of requirements derived from NASA's Constellation Program. In particular, we adopted both the computer-based control system safety requirements and the safety-related requirements completeness checks from the Constellation Program's computing system requirements. The control system safety requirements address issues associated with loss of function and with inadvertent activation. The completeness checks address a number of issues associated with the specification and implementation of features for software safety and for the interaction of hardware and software. For each relevant element in both sets of adopted requirements, we identified one or more demonstration features that might show how the requirement might be met using a software health management approach. Our prototype application software mimics the Space Shuttle orbiter's abort control sequencer software task, which provides safety-related functions to manage vehicle performance during launch aborts. We turned this task into a goal-based function that, when working in concert with the software health manager, aims to work around software and hardware problems in order to maximize abort performance results.

In addition to our adopted application requirements, the ARINC-653 standard imposes a number of requirements on the system integrator for developing the requisite error handler process. Under ARINC-653, the health monitoring (HM) service is invoked by an application calling the application error service or by the operating system or hardware detecting a fault. The recovery action is dependent on the error level. The system integration specifies in the module HM and partition HM tables the recovery actions for each module and partition level error. The application programmer defines in a special error handler process the recovery actions for process-level errors. The error handler can stop and restart failed processes, restart the entire partition, or shut down a partition. It is these HM and error process details that we implement with the MDS technology, showing how a state-analytic approach is appropriate for identifying fault determination details, and showing how the framework supports acting upon state estimation and control features in order to achieve safety-related goals.

Our paper will describe the requirements, design, and implementation of our software health manager (the error handler process) and the software under control (the application processes). We will provide some test results of this Phase II prototype, and will describe future directions for the remainder of Phase II and the new topics we plan to address in Phase III.

Acknowledgment

The FAILSAFE project is sponsored by a NASA Small Business Technology Transfer (STTR) Phase II award under contract NNA07AA24C.