

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

AIAA Infotech@Aerospace 2010 Extended Abstract Submission for the Integrated System Health Management (ISHM) Technical Area

Title: Lessons learned on implementing Fault Detection, Isolation, and Recovery (FDIR) in a Ground Launch Environment.

Bob Ferrell

NASA
NE-E9
Kennedy Space Center, FL 32899
Bob.A.Ferrell@nasa.gov

Mark Lewis

NASA
NE-E9
Kennedy Space Center, FL 32899
Mark.E.Lewis@nasa.gov

Jose Perotti

NASA
NE-E9
Kennedy Space Center, FL 32899
Jose.M.Perotti@nasa.gov

Barbara Brown

NASA
ARC-T
Kennedy Space Center, FL 32899
Barbara.L.Brown@nasa.gov

Rebecca Oostdyk

ASRC Aerospace
ASRC-25
Kennedy Space Center, FL 32899
Rebecca.L.Oostdyk@nasa.gov

Jesse Goerz

ASRC Aerospace
ASRC-39
Kennedy Space Center, FL 32899
Jesse.W.Goerz@nasa.gov

This paper's main purpose is to detail issues and lessons learned regarding designing, integrating, and implementing Fault Detection Isolation and Recovery (FDIR) for Constellation Exploration Program (CxP) Ground Operations at Kennedy Space Center (KSC).

Part of the overall implementation of National Aeronautics and Space Administration's (NASA's) CxP, FDIR is being implemented in three main components of the program (Ares, Orion, and Ground Operations/Processing). While not initially part of the design baseline for the CxP Ground Operations, NASA felt that FDIR is important enough to develop, that NASA's Exploration Systems Mission Directorate's (ESMD's) Exploration Technology Development Program (ETDP) initiated a task for it under their Integrated System Health Management (ISHM) research area. This task, referred to as the FDIR project, is a multi-year multi-center effort. The primary purpose of the FDIR project is to develop a prototype and pathway upon which Fault Detection and Isolation (FDI) may be transitioned into the Ground Operations baseline.

Currently, Qualtech Systems Inc (QSI) Commercial Off The Shelf (COTS) software products Testability Engineering and Maintenance System (TEAMS) Designer and TEAMS RDS/RT are being utilized in the implementation of FDI within the FDIR project. The TEAMS Designer COTS software product is being utilized to model the system with Functional Fault Models (FFMs). A limited set of systems in Ground Operations are being modeled by the FDIR project, and the entire Ares Launch Vehicle is being modeled under the Functional Fault Analysis (FFA) project at Marshall Space Flight Center (MSFC). Integration of the Ares FFMs and the Ground Processing FFMs is being done under the FDIR project also utilizing the TEAMS Designer COTS software product. One of the most significant challenges related to integration is to ensure that FFMs developed by different organizations can be integrated easily and without errors. Software Interface Control Documents (ICDs) for the FFMs and their usage will be addressed as the solution to this issue. In particular, the advantages and disadvantages of these ICDs across physically separate development groups will be delineated.

After creating the functional fault models in the TEAMS Designer COTS software, the modeler can utilize various reporting and analysis tools within the TEAMS Designer COTS software to verify the functionality of the model. The TEAMS Designer COTS software contains a TEAMS-RT Analysis Tool and Design for Testability (DFT) feedback which gives the modeler the opportunity to inject failure modes and failed test results into the model to test its functionality and do limited validation.

Ultimately, the TEAMS Designer COTS software outputs a Dependency-Matrix (D-Matrix) that maps failure effect propagation paths from failure modes identified in the FFMs to observable points, or test points, in the system. The TEAMS RDS/RT COTS software then uses the D-Matrix in real-time to determine which modules in the model are bad, suspect, or unknown, based on test results that are passed to it. In order for the TEAMS RDS/RT COTS software product to function correctly, custom software must be developed for interfacing to the CxP Ground Operations Launch Control System (LCS) and implementation of tests for resultant passage to the TEAMS RDS/RT COTS software.

This custom software is generically referred to as “wrapper code”. The wrapper code must be tailored to the telemetry, TEAMS model, and GUI with which it interfaces, but this results in code that is application specific and requires re-work each time the telemetry, model or GUI changes. Previous wrapper code applications required heavy involvement of the TEAMS modeler in the development process which necessitated a special skill mix to get the TEAMS model operating in real-time. A WrapperD application was developed with the intent of isolating the interface functions from one another so they can easily be tailored for other environments and mitigating the dependency of the WrapperD code on the intricacies of the TEAMS model to limit the modeler’s involvement in wrapper code development.

In particular the WrapperD application performs the tasks required for TEAMS RDS/RT COTS software to use the D-Matrix through the TEAMS APIs. These tasks include:

- 1) parsing the incoming telemetry stream,
- 2) performing logical tests based on the current telemetry data,
- 3) passing the results of the logical tests to TEAMS RDS through the TEAMS API,
- 4) requesting the bad, suspect, and unknown diagnoses from TEAMS RDS through the TEAMS API, and
- 5) communicating information that is required by a graphical user interface (GUI) to display the telemetry, test results, and diagnosis to a user.

Each of these five tasks is as independent as possible to facilitate re-use of the code in other real-time systems, and some of the code in the WrapperD application is auto-generated to make it re-usable regardless of the content of the TEAMS model.

All of this software is designed to operate in an open architecture that is utilized to integrate FDIR in the CxP Ground Operations Launch Control System (LCS). This architecture will be briefly discussed along with the overall Functional Fault Model (FFM) architecture that has been developed. Due to their complexity and failure history, cryogenic systems are being targeted for initial development in the FDIR prototype. In particular we have designated the initial subsystems for prototype integrations to be the Ground Liquid Hydrogen (LH2) subsystem and the Ares Upper Stage LH2 Main Propulsion Subsystem (MPS). In addition, performance issues in attempting to integrate FDIR into such large complex systems as the Ares Launch Vehicle and Ground Operations/Processing are also addressed.

Finally, further development of CxP Ground Operations FDIR will be discussed along with the future direction of FDIR at KSC.