SOFTWARE SAFETY ANALYSIS ACTIVITIES DURING SOFTWARE DEVELOPMENT PHASES OF THE MICROWAVE LIMB SOUNDER (MLS)

Hui-Yin Shaw, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109. 818 354-9540, hui-yin.shaw@jpl.nasa.gov
And
Joseph S. Sherif , California State University, Fullerton, CA 92834.
714 278-2740, jsherif@fullerton.edu

ABSTRACT

Safety analysis is a systematic and orderly process for the acquisition and evaluation of specific information pertaining to the safety of a system. The purpose of the Microwave Limb Sounder (MLS) Software Safety Analysis (SSA) is to identify potential hazards to MLS, the Earth Orbiting System Satellite (EOS) and related launch vehicle facilities and personnel. The results of the SSA will be used to: 1. Affect the requirement and design of the software system whenever practical to assure control and mitigation of possible system hazards, and 2. Identify those potential hazards introduced or impacted by the software systems. This paper describes the MLS software safety analysis activities and documents the SSA results.

SOFTWARE SAFETY ACTIVITIES (SSA)

The MLS software safety analysis is tailored from the methodology provided in the NASA Technical Standard for Software Safety [4]. The tailored SSA objectives and activities are described in this section. Guidelines provided in the MLS System Safety are given below.
1. The safety criteria and methodology used to classify and rank the potential hazards are taken from MIL-STD-882C Table 1 for Catastrophic (Category I) and Critical (Category II) Hazards. [3]
2. Software is classified as safety-critical when it is a potential cause of a hazard or will be used to support the control of a hazard.
3. Hazardous software commands that are only executed during unmanned flight operations are not regarded as safety risks, but rather as reliability risks with potential of damage to the instrument/system or loss of scientific data.
4. All hazard reports will have traceability by providing specific source references for each control and verification approach.

INTRODUCTION

The Microwave Limb Sounder (MLS) is an instrument to be carried on board a NASA Earth Observing System (EOS) satellite. Its objective is to measure naturally occurring microwave thermal emission from the limb of Earth's atmosphere to remotely sense vertical profiles of selected atmospheric gases, temperature and pressure. Previous and on-going MLS experiments include spacecraft, aircraft and balloon versions. The space MLS experiment is designed to address a broad range of global change issues. A series of spectrometers and radiometers covering a range of frequencies will be employed in this MLS experiment. The instrument software is defined to include all flight software developed for execution in the MLS instrument flight computer.

FLIGHT SOFTWARE DEVELOPMENT

The software development for the EOS MLS Flight Software represents an approximate 3-year effort at a staffing of three software developers on the average, for that duration. There is a total of approximately 10,000 Lines of Code (LOC). The MLS software development follows the Waterfall Lifecycle model: Requirements Analysis, Design, Implementation and Test.

SSA OBJECTIVES AND ACTIVITIES

System Requirements and Design Phase

SSA Objective: Review input from system safety analyses and identify any software that has the potential to cause a hazard or is required to support control of a hazard.
During this phase, the System Safety Engineer examines the MLS flight and ground support equipment design, interfaces, test and operations for potential hazards at the system and subsystem levels. The Preliminary Hazard Analysis (PHA) and the Phase I Safety Assessment Report are produced as result of this activity. These reports identify catastrophic and critical hazard causes pertaining to pre-launch, launch, and post-launch periods. The safety criteria and methodology used to classify and rank the potential hazards of MLS instrument are taken from MIL-STD-882C Table 1 for Catastrophic (Category 1) and Critical (Category II) Hazards. [2]
The software hazard analysis is an extension of the system hazard analysis.
The SSA activities in this phase include:
- Review the available system safety reports [1, 2]
- Identify the reported hazards that may be attributed to software faults
- Identify the software components that take part in the detection or control of system/component hazards

Software Requirements Phase

SSA Objectives: Ensure that the development of the software requirements includes the software safety requirements, which addresses software hazard issues identified in the previous phase. Also Ensure that appropriate instrument safety requirements flow down to the software safety requirements and that they are adequate.
The SSA activities in this phase include:
- Follow-up on the concerns identified in System Safety Analyses phase.

- Identify critical commands using inputs from the SSA work of previous phase and the system safety requirements. Critical commands are those commands that are hazardous to the operation or safety of the instrument if used improperly or untimely
- Recommend software safety requirements as appropriate.
- Review the Software Requirements document to make sure that Instrument (system) safety requirements are adequately addressed in the software safety requirements.

Software Design & Implementation Phase

SSA Objectives: Ensure that the software design and implementation properly incorporate software safety requirements. And ensure that the appropriate test cases, procedures and success criteria are defined to ensure proper implementation of the software safety requirements and design.
The SSA activities in this phase include:
- Review (sub) system and component Failure Mode Effect Analyses (FMEAs) and Fault Tree Analyses (FTAs) for hazards that may potentially be attributed to software.
- Identify safety-related deficiencies in design and recommend for correction
- Ensure that test plan and procedure contain adequate test cases and success criteria for verifying software safety requirements and design
- Analyze software requirements and design changes for safety impact.

Software Acceptance Test Phase

SSA Objective: Ensure that the results of the software safety verification are satisfactory.
The SSA activities in this phase include:
- Ensure that test cases for software safety/fault-protection requirements have been conducted and that the success criteria are met
- Review software change requests for safety impact
- Ensure that test cases for safety/fault-protection requirements are appropriately revised as needed when changes are made to the software safety requirements/design.
- Ensure that safety-related information is included in the User Guide or other appropriate documentation.

Instrument Integration Phase and Beyond

SSA Objective: Ensure that the results of the software safety-related verifications are satisfactory.
The SSA activities in this phase include:
- Assess proper closure of safety-related software anomalies. Software problem reports having safety impact are directed to the Systems Safety Office for review.[3]
- Review software change requests for safety impact
- Ensure that software changes with safety impact are adequately verified in software regression test prior to submission for system-level test.

Results and Findings

The SSA results from each development phase are provided to cognizant engineers in a concurrent engineering fashion to facilitate timely evaluation of safety issues. The results and findings are reported to the System Safety Engineer for inclusion in the System Safety Data Package and are summarized in Table 1.

Acknowledgments

Table 1. Summary of SSA Activities and Results by Development Phase

Development Phase	Activities	Results
System Requirements and Design Phase	Review the two available system safety reports [1, 2] Identify the reported hazards in [1, 2] that may be attributed to software faults .	Analysis of the MLS Flight Equipment and Ground Operations has identified eight potential hazards. Out of these eight hazards, two were identified for further investigation for possible software involvement. Of these two potential software hazards, one was determined to be a non-issue and the other was followed-up in the software Requirements Phase (see first item in S/W Requirements Phase).
	Identify software components that take part in the detection or control of system/component hazards, when information is available.	No information on required software components to detect/control system hazard was available during this phase. However this information became available in the Software Design phase.
S/W Requirements Phase	Follow-up on the concerns identified in System Safety Analyses phase. Identify critical commands Recommend software safety requirements as appropriate	One command was identified as critical command. Recommendations were made, and they were incorporated in the revised Software Requirements Document.

	Ensure that instrument safety requirements are adequately addressed in the SRD.	Instrument safety/fault protection requirements were traced to software safety/fault protection requirements. Various recommendations were made to software requirements and changes were incorporated in the subsequent SRD update. Command-related requirements are in compliance with system-level requirements.
S/W Design Phase	Review system and component FMEA & FTA analyses.	Reviewed System-Level FMECA final version and found no s/w related issues, except for those previously identified in the Software Fault Tree Analysis study. These are software reliability issues (or mission critical) and are not safety-critical within the context of system safety Reviewed IGSE FMEA [MLS IGSE-EM Interface FMEA (Rack #1) for potential software safety issues. No safety issues relevant to software were reported.
	Identify safety-related deficiencies in design and recommend for correction	Reviewed Software Design Document, Software Requirements Document, and Command and Telemetry Handbook. Discrepancies, issues and recommendations were noted. These include mission-critical issues (inconsistencies in the engineering and science channels for the downlink telemetry's and command formats). None of these issues identified are safety-hazardous.
	Ensure that test plan and procedure contain adequate test cases and success criteria for verifying software safety requirements and design	Safety-related test cases are added. These new test cases are traced to safety requirements / design [5-7].
	Analyze software changes for safety impact	Reviewed revised SRD and found no negative safety impact from changed requirements.

S/W Acceptance Test Phase	Ensure software safety test cases are successful	(This portion of the analyses is to be reported at the completion of system integration testing.)
	Review software change requests for safety impact	
	Ensure appropriate revision of test cases as needed	
	Ensure safety-related information is included in the User Guide or other appropriate documentation [8].	
IT&V Phase	Assess proper closure of safety-related software anomalies	Reviewed IGSE User's Guide [9] No safety-related operational constraints were identified.
	Review software change requests for safety impact	
	Ensure adequate software regression test for software safety-related changes	

REFERENCES

[1] MLS Preliminary Hazard Analysis IOM 516-DSR-97-059, August, 1997.

[2] MLS Phase I Safety Assessment Report, JPL D-14871, Jan. 1998.

[3] EOS MLS System Safety Plan, JPL D-12980, Sept. 1997.

[4] NASA Technical Standard, Software Safety, NASA-STD-8719.13A. 1999.

[5] NASA Software Assurance Standard, NASA-STD-2201-1993.

[6] JPL Standard for System Safety, JPL D-560, 1999.

[7] MLS Instrument Functional Requirements and Design Constraints, JPL D-13362, 1998.

[8] General Interface Requirements Document (GIRD), GSFC 422-11-12-01, Jan. 1994.

[9] EOS MLS Instrument Ground Support Equipment User's Guide, JPL D-17011, 1999.