NASA-CR-197760

NASA-IVV-94-009
WVU-SRL-94-009
WVU-SCS-TR-95-9
CERC-TR-TM-94-009

NASA/WVU Software IV & V Facility
Software Research Laboratory
Technical Report Series

IN-61-CR
49558
p. 5

Using SCR Methods to Analyze Requirements Documentation

by John Callahan and Jeffery Morrison

(NASA-CR-197760) USING SCR METHODS TO ANALYZE REQUIREMENTS DOCUMENTATION (West Virginia Univ.) 5 p

N95-26668

Unclas

G3/61 0048508

National Aeronautics and Space Administration

West Virginia University

According to the terms of Cooperative Agreement #NCCW-0040, the following approval is granted for distribution of this technical report outside the NASA/WVU Software Research Laboratory

George J. Sabolish 4-19-95
George J. Sabolish Date
Manager, Software Engineering

4-19-95
John R. Callahan Date
WVU Principal Investigator

Published in the Proceedings of the 4th International Naval Research Laboratory Software Cost Reduction Workshop, Washington, DC, November 29-30, 1994

Using SCR Methods to Analyze Requirements Documentation

John Callahan *Jeffery Morrison*
NASA/West Virginia University IV&V Facility

1 Introduction

It is well-known that prose-based requirements documents are often inconsistent and incomplete. Many projects, however, still communicate system requirements primarily through the use of prose. As an experiment, we are using SCR methods to verify selected portions of NASA's EOSDIS Core System (ECS) requirements documentation. We are using the SCR approach as a spot-inspection tool on ECS requirements documents even though the requirements are not described in the SCR format. We believe that the formal and systematic approach of the SCR requirements methods will provide insight as to whether the requirements are internally inconsistent or incomplete. To date, we have successfully found several problems with the ECS documentation such as inconsistencies within operational scenario descriptions. Such inconsistencies are often due to ambiguous organization or confusing language. We outline how the SCR approach is used to identify poor descriptions within project documentation. Our goal is to help verification and validation (V&V) teams to identify such problems early in the software development lifecycle even in projects that do not employ full SCR requirement specifications.

2 Background

As part of its Mission-To-Plant Earth project, the National Aeronautics and Space Administration is building the Earth Observing System (EOS) to monitor various aspects of the Earth's ecology. Eventually, EOS will be comprised of over 10 satellites in Earth orbit that each serve as platforms for several oceanographic and geological science instruments. The spacecraft operations, data archiving, and requests handling subsystems of EOS form the EOS Data and Information System (EOSDIS). EOSDIS will serve as the clearinghouse for data requests from end-user scientists around the world.

The EOS Core System (ECS) is the central component of the EOSDIS, providing the coordinating functions for the EOS operational ground system. It constitutes the largest portion of the EOSDIS. The ECS will provide the planning and scheduling, command and control, data processing, data archiving, system management, communications management, networking, data distribution functions required to support EOS operations and data access.

3 Description of Problem

The requirements for ECS are contained in several large documents with a great majority of them written in prose form. Of these documents, there are two that essentially describe the behavior of ECS. They are the *Operations Concept Document (OC)* and the *Functional and Performance Requirements Specification Document (FPRS)*. The OC Document describes scenarios for normal ECS functionality. The document also contains a small number of diagrams that help clarify the prose of the scenarios. The FPRS Document includes brief scenarios, context diagrams, conceptual data flow tables, and functional requirements (i.e., shall causes).

NASA does use an electronic database to manage relationships between these requirements and other development artifacts (e.g., design components, code). Various database tools allow for analysis of relationships between requirement levels[1], within requirements levels by keyword association, and artifact tracability.

Given this requirements framework, it is difficult to determine if the requirements are complete and consistent with respect to specific operational scenarios. A scenario, as described in the OC document, may involve many requirements at different levels. Changes to scenarios in the OC document, however, cannot be easily traced to the requirements database. Thus, we believe that documents will become both internally and externally inconsistent and incomplete as the scenarios of intended system usage evolve.

4 Applying the SCR methods

By modeling the scenarios and requirements as mode charts using the SCR methods, we have been able to identify problems within and between the documents. Our analysis using the SCR requirements method has been applied in two areas: the first being the scheduling scenarios for data acquisitions; and the second being the requirements for Flight Operations Segment (FOS). For the scheduling scenarios, which are from the Operations Concept Document, we modeled the Data Acquisition Request (DAR) and initial scheduling scenario using mode transition tables. The DAR scenario describes the process for scientist to request environmental data that is not already being acquired through normal EOS instrument observations. The process of developing the mode transition table for DAR scenario was very straight forward. We made all the transitions from one component to another the modes and the steps necessary to arrive at a transition the conditions for a change in modes.

The initial scheduling scenario depicts the steps necessary for scheduling environment data collection for a particular week in the future. For this scenario we took a different approach. We

[1] Requirement levels are defined as broad functions in a hierarchical fashion. For example, a Level 0 requirement might be "Place a man on the moon" and descendant Level 1 requirements of the Level 0 requirement would be *Build a rocket, Launch it*, ...) Level 2, 3 and 4 requirements address system, subsystem, and component level requirements respectively

developed a mode transition table for each component involved in the scenario. After developing the individual tables, we merged them into a single table representing the entire scenario. We felt this approach would highlight any inconsistencies among component interfaces.

The FOS is responsible for EOS mission operations, including the planning, scheduling, commanding, and monitoring of U.S. spacecraft and U.S. EOS instruments on-board the U.S. and International Partner (IP) series of spacecraft. In our analysis, we have only modeled the input and output data items for the FOS requirements that are derived from Performance Requirements Specification Document. We derived the input and output data items by first examining the data flow of each individual component in the Context Diagrams. The diagrams consist of pictorial representations of component interactions. From the information gathered from the Context Diagrams, we started the development of the input and output data items. We then compared this information with the Conceptual Data Flow Tables. The tables consist of prose descriptions of component interactions. The exercise of comparing the information gathered from the diagrams and the tables highlight any inconsistencies between them.

5 Discussion

We believe the experiment has been successful. For the scenarios, we found several inconsistencies. Most are caused by confusion on the authors interpretation of scenarios in writing requirements and vice-versa. In addition, we have used the SCR methods to find incomplete scenario and requirements descriptions. They are often incomplete because the scenarios only address the nominal cases. Many scenarios do not specify behavior for off-nominal cases. While the Operations Concepts document is intended to describe desire behavior, the lack of off-nominal cases allows for ambiguity in the interpretation of functionality in such cases. If this is left until design or coding phases, we believe that serious and costly problems will occur.

The requirements for the FOS input and output data items have 7 inconsistencies. Six of these come from disagreements between the FOS Context Diagram and the Conceptual Data Flow Tables of individual components. The FOS Context Diagram seems not to have been proofed very well. The remaining inconsistency deals with a misaligned data item name one of the Conceptual Data Flow Tables. Incompleteness is also evident for the FOS requirements. The procedure of creating the input and output data items clearly shows the authors have yet to consider the content or format of any information passed between components.

In the future, we plan to use the mode transition tables as formal scenarios for examining the electronic databases used to manage the requirements. We are currently constructing additional *views* of the requirements database based on the mode transition tables. We believe that this view combined with the requirements levels view will facilitate better tracking of management and technical changes to the requirements and their impact on the project.