SMEI Software Development

M.P.Cooke

and

C.J. Eyles

University of Birmingham

Issue 1

24 Feb 2000

Table of Contents

1. Introduction 3

2. Software Development and Management 3

2.1 Development Methods and Tools 3

2.2 Test and Verification 3

2.3 Configuration Control 3

2.4 Documentation 4

3. Status of Required Software 5

3.1 Onboard Control Software 5

3.2 Camera EGSE Software 5

3.2 Spacecraft Simulation Software 6

3.3 Telemetry Display Software 6

3.4 Image Display Software 6

3.5 Data Format Conversion Software 6

1. Introduction

There are a number of different areas for which software is required as part of the SMEI development programme. This document presents the methodology followed for SMEI software development, and the tools and utilities used. Finally, a breakdown of the major pieces of software presents their current status and the remaining work to be completed.

2. Software Development and Management

The following sections cover four main areas of interest for software development. The methodology, tools, test and verification procedures and configuration control used are all examined.

The software has been designed with testability and speed of development in mind, and is based on experience with previous missions.

2.1 Development Methods and Tools

The SMEI software can be divided into two major groupings: the onboard control software, and the ground support software.

The onboard software is built using the Texas Instruments development tools and it is written in assembler. Debugging is done using an in−circuit emulator, and PC based control software. Supporting data files, containing flat field tables, regions of interest, and so on are currently created using a text editor.

Once the 1553 interface code is complete and verified, development will switch to using the spacecraft simulator and patching. This will provide a useful indication of the amount of work required to support SMEI after launch.

The ground support software is built using Microsoft Visual Studio. It is written in a mixture of Visual Basic, Visual C++ and a small amount of assembler. Debugging is done using the integrated debugger. Most of these utilities are for supporting the development of the onboard software.

2.2 Test and Verification

The SMEI software has a modular design, with well defined interfaces between the modules. This permits individual testing at the module level, and a reduction in the scope for complex interactions between different modules.

Once the pace of development work declines, hardware and software verification scripts will be written to permit automated testing of the onboard software via the spacecraft simulation software.

Post−launch, the EM hardware will be used to verify the correct operation of new patches and operational procedures prior to their use on the FM.

2.3 Configuration Control

Due to the rapid development cycle, the exact versions of software in use at any time is important for tracing and eliminating software bugs. We have the following targets for the software:

- Well defined software releases
- Onboard software release number returned in the state of health
- All files required for every release are securely archived
- Software developed under a revision control system (CVS and SourceSafe)
- Well commented source code, with changelogs in the source files
- E^2Prom data tables have release numbers for data analysis purposes

With the onboard software development two release strategies are being followed. For EM development, software updates are written directly to the bootstrap E^2Prom as changes are made. This permits rapid updates and short turnaround of bug fixes and testing. It also builds confidence in the bootstrap code, which cannot be changed post−launch.

For the FM unit, where the number of write−cycles per E^2Prom chip needs to be controlled, updates are created as temporary software patches. These are transmitted to the DHU from the spacecraft simulator are part of the SMEI power−up configuration procedure. This permits the patching procedure to be tested, as well as reducing the E^2Prom write cycle count. FM software will be written to the bootstrap E^2Prom at well defined stags of the programme, eg before major tests, delivery, etc. This requires the test connector bootstrap write−enable link to be made.

The DHU E^2Prom contains data for flat field tables, regions of interest, instrument configurations, etc. A release identifier is provided, along with a checksum to track changes. The checksum is important for tracking changes to the flat field tables due to the extended time required to upload a complete table.

2.4 Documentation

There will be a number of other documents produced to support the software development.

- Commanding Specification (released as SMEI/BU/SPE/002)
- Telemetry Specification (released as SMEI/BU/SPE/003)
- Software Users Guide
- Operations Guide

The command and telemetry specification, along with the ICD, document the interface between the SMEI onboard software and the spacecraft.

The software users guide will cover procedures used to build a new release of the onboard software, patch command files and so forth.

The operations guide will provide instrument operation guidelines and limitations. It will also document the configuration tables in the DHU E^2Prom.

3. Status of Required Software

The following sections present the six major software tasks required to support SMEI. Function, current status and remaining work are presented for each module. The overall interactions between the support software is shown in figure 1.

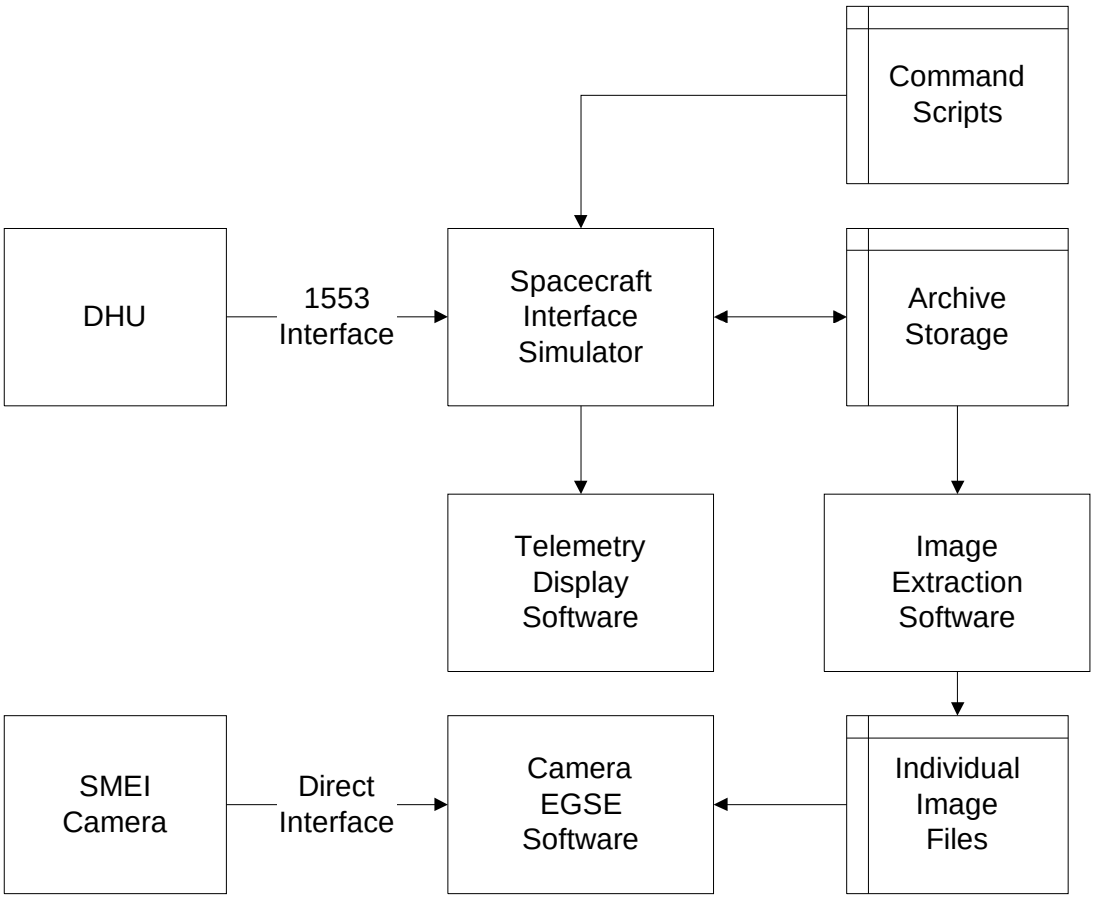

Figure 1. Software Interaction

3.1 Onboard Control Software

This is the software required onboard the SMEI instrument. It handles data acquisition, formatting and transmission to the spacecraft, along with instrument state of health generation.

Most individual software modules are complete. Further development of this software is on−hold until DHU hardware is available. Overall module integration and bug testing on real hardware to commence in early March 2000 when the EM DHU is available.

1553 functionality, command handling, and science & state of health data generation will be completed by the end of March 2000. This will enable SMEI to support spacecraft level 1553 compatibility tests in May 2000.

The expected completion date for the onboard software is August 2000. This will enable SMEI to support integration later in 2000.

3.2 Camera EGSE Software

This is test software used for camera checkout at the sub−system level. It provides basic control of the camera, permitting timed exposures to be made, and image storage to disk. Image presentation and basic analysis tools are also part of the package.

This software is complete.

3.2 Spacecraft Simulation Software

This GSE software permits testing of the SMEI instrument via a spacecraft interface simulator. The SMEI spacecraft simulator provides a simple mechanism to send commands to SMEI, receive science and SoH data, and to store the data received to a permanent archive.

This software is mostly complete and is currently under active development. Interface to the 1553 interface board in the EGSE−PC has been configured, and the software to support the science and state of health channels has been completed.

The software will require another man−week of development time to add command and attitude data channel support, and to add data archiving. Later in the programme, a simple macro language may be added if time permits.

3.3 Telemetry Display Software

This GSE software presents display pages to the user from real−time or stored state of health data.

This software has not been started. Creation of a framework for display page creation and data formatting will take approximately three man−weeks. Building a full suite of display pages can be done as the onboard software development progresses and the schedule requires.

3.4 Image Display Software

This GSE software provides image display capabilities for x−band data.

This software has not been started. A rudimentary image extraction tool can be written in a few days to produce images from the science data stream which can be loaded into the camera EGSE software. This will be enough to meet Birmingham test requirements. Writing a full display and analysis program would take many weeks.

3.5 Data Format Conversion Software

This GSE software provides an interface layer between RSC provided data and the data formats required for other GSE software modules.

This software will only be written once the data presentation formats between RSC and SMEI have been finalised. It should be reasonably trivial to write and test in a few days once the specification is decided.