STEREO
IMPACT

GSE Software Development Plan

GSE_SDP_A.doc
Version A – 2001-Sep-06

_____
David Curtis, UCB IMPACT Project Manager

Document Revision Record

Rev.	Date	Description of Change	Approved By
A	2001-Sep-6	Preliminary Draft	-

Distribution List

Dave Curtis, UCB
Mike Hashii, UCB
Peter Scroeder, UCB
Andrew Davis, Caltech
Tycho von Rosenvinge, GSFC
Mario Acuna, GSFC
Harry Culver, GSFC

Table of Contents

1. Overview

1.1. *Introduction*

IMPACT consists of a number of instruments connected to the STEREO spacecraft via the IMPACT IDPU.   This document describes the software development plan for GSE software used in the development and test of the IMPACT suite. A number of such GSE shall be developed at UCB, Caltech, and GSFC. In addition, this document covers the software used in the post-launch data production tasks.

UCB has had experience with similar GSE and data processing software tasks on HESSI, Lunar Prospector, FAST, Cluster, Mars Global Surveyor, Wind, Polar, etc.   Caltech and GSFC have had experience on ACE, MGS, Wind, Mars Global Surveyor, etc.

1.2. *Document Conventions*

In this document, TBD (To Be Determined) means that no data currently exists. A value followed by TBR (To Be Resolved) means that this value is preliminary. In either case, the value is typically followed by a code such as UCB indicating who is responsible for providing the data, and a unique reference number.

1.3. *Applicable Documents*

The following documents include drawings and STEREO Project policies, and are part of the Interface Requirements. In the event of a conflict between this SDP and the following documents, this SDP takes precedence. Most documents can be found on the Berkeley STEREO/IMPACT FTP site:

http://sprg.ssl.berkeley.edu/impact/dwc/

1.   Specifications/IDPUSimSpec – specification for the IDPU Simulator GSE hardware

1.4. *Acronyms*

CDR	Critical Design Review
CSCI	Computer Software Configuration Item
DCB	Data Controller Board
GSE	Ground Support Equipment
GSFC	Goddard Space Flight Center
ICD	Interface Control Document
IDPU	Instrument Data Processing Unit
IMPACT	In-situ Measurements of Particles and CME Transients
ISG	IDPU Simulator GSE
MAG	Magnetometer instrument
MOC	Mission Operations Center
PDR	Preliminary Design Review
PLASTIC	PLasma and SupraThermal Ion Composition instrument
PER	Pre-Environmental Review

POC   Payload Operations Center
SDP   Software Development Plan
SEP   Solar Energetic Particle instrument suite
SSC   Stereo Science Center
SSL   Space Sciences Laboratory
STE   SupraThermal Electron instrument
SWAVES   STEREO Waves instrument
SWEA   Solar Wind Electron Analyzer
TBD   To Be Determined
TBR   To Be Resolved
UCB   University of California at Berkeley
UNH   University of New Hampshire

2. Instrument Ground System Description

2.1. *IDPU Simulator GSE*

The IDPU Simulator GSE is used to test instruments prior to integration with the IDPU.
The GSE consists of a hardware box (the ISG) that performs the low-level interface
functions, and a PC on which runs the user interface software. This software consists of
two modules that may or may not run on the same PC: The ISG Core Software, which
performs the low-level interface with the ISG hardware, and can formulate instrument
commands and display instrument data at a very low level (Hex), and a Science Module
Module, optimized for each instrument to display data and send commands. The link
between the PC and the ISG is through the PC parallel port.

UCB is responsible for the ISG hardware, the ISG Core Software, and Science Modules
for SWEA, STE, and MAG.   Caltech will develop Science Modules for SEP, which will
be basically the same as the Science Display software used with the Command &
Telemetry GSE at the IMPACT suite level (since the IDPU is mostly a bent pipe for SEP
commands and telemetry). UNH is responsible for Science Module for PLASTIC.
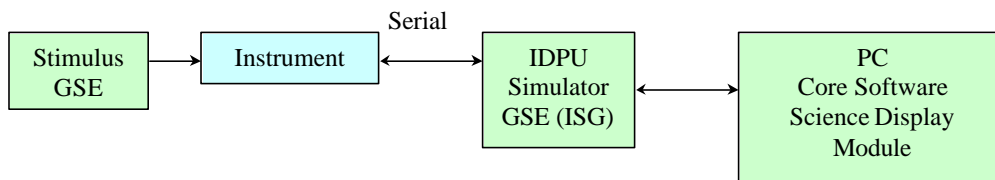

Figure 2.1-1 IDPU Simulator GSE Setup

2.2. *Instrument Simulator GSE*

The ISG hardware can also be used as an instrument simulator input to the IDPU. In this
application, the ISG will be connected to a PC running Instrument Simulator software to
provide a continuous stream of data similar to what is made by the instrument, and
displaying any commands received from the IDPU. This software will be developed by
UCB. It will only be used in bench-level tests of the IDPU prior to integration with the
instruments.

2.3. *Command & Telemetry GSE*

This GSE is used at the IMPACT Suite level, and interfaces to either the APL-supplied Spacecraft Emulator (prior to IMPACT integration with the spacecraft), or the MOC (after integration with the spacecraft). It performs the command scripting and housekeeping display / limit checking functions. It will be used starting at IDPU bench testing, through Suite and Spacecraft I&T, and will form the heart of the POC during post-launch Operations. Note that since the connection to the MOC or Emulator is via Ethernet, the Command & Telemetry GSE may be remote.
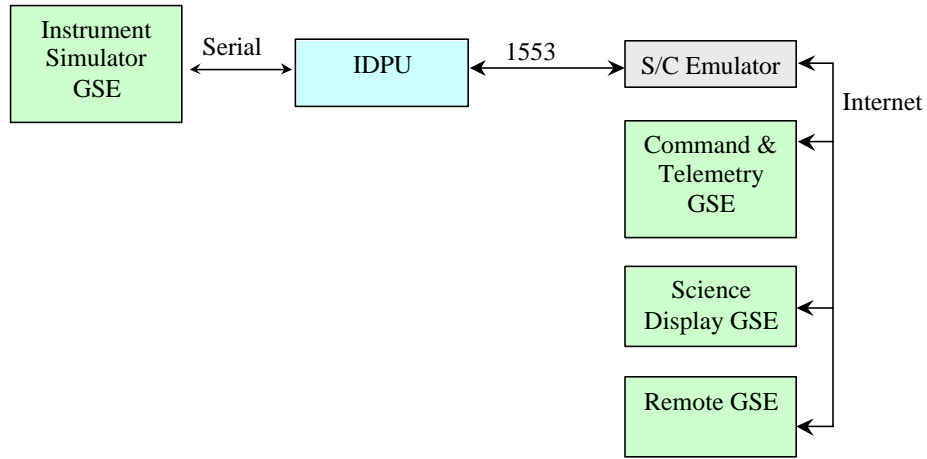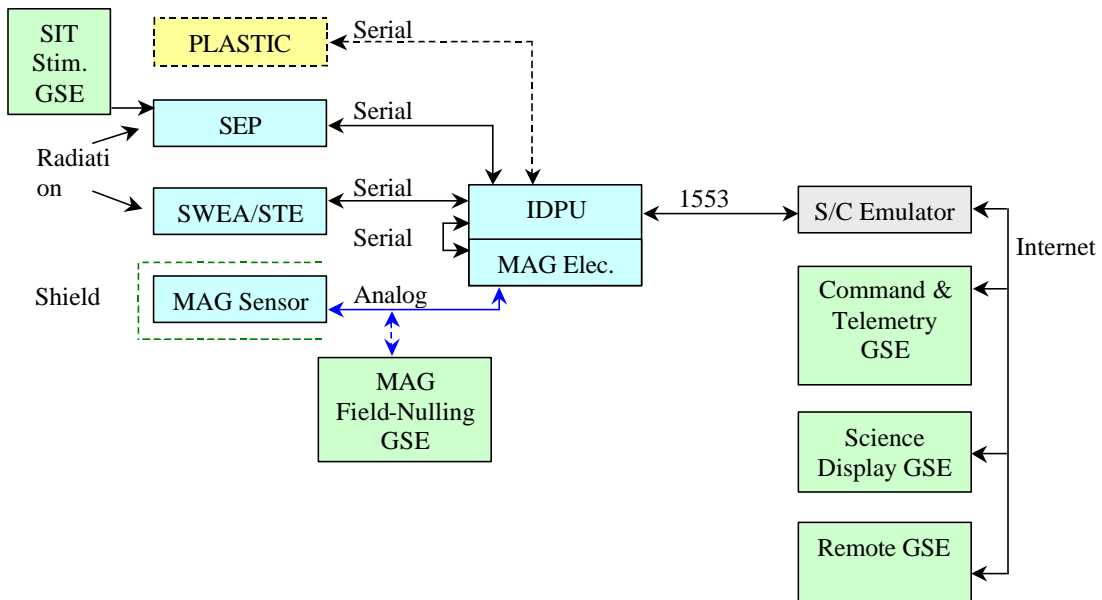
Figure 2.3-1 IDPU GSE Setup
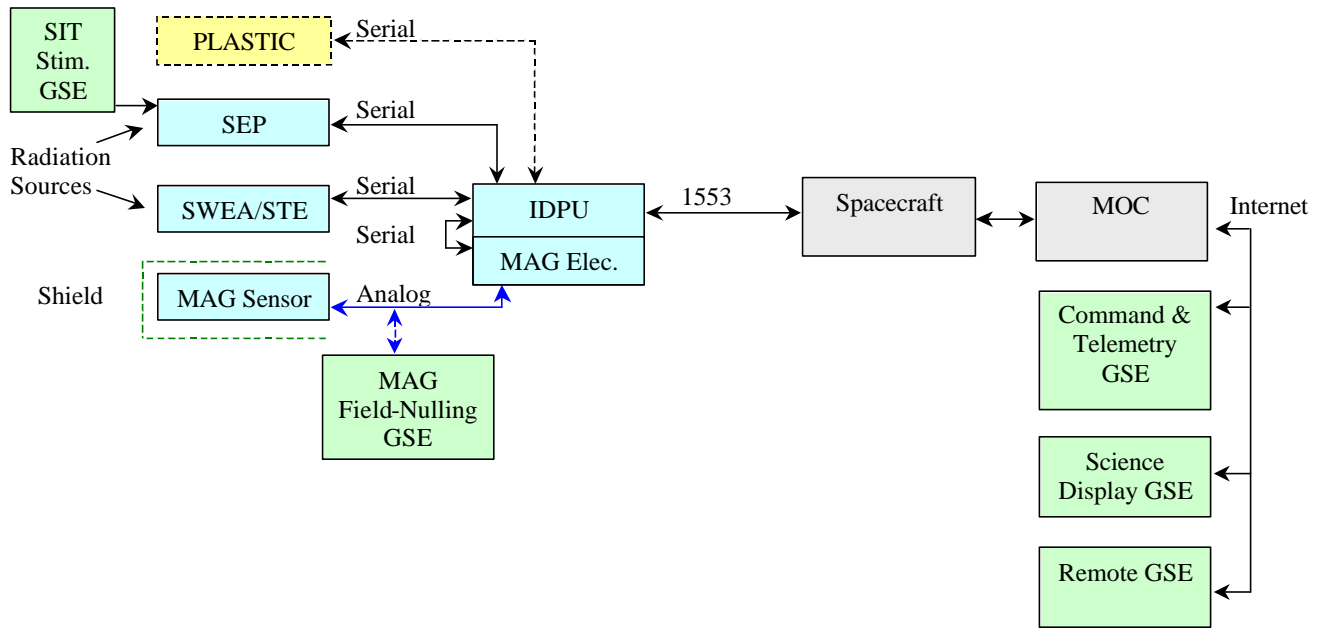
Figure 2.3-2 IMPACT Suite GSE Setup
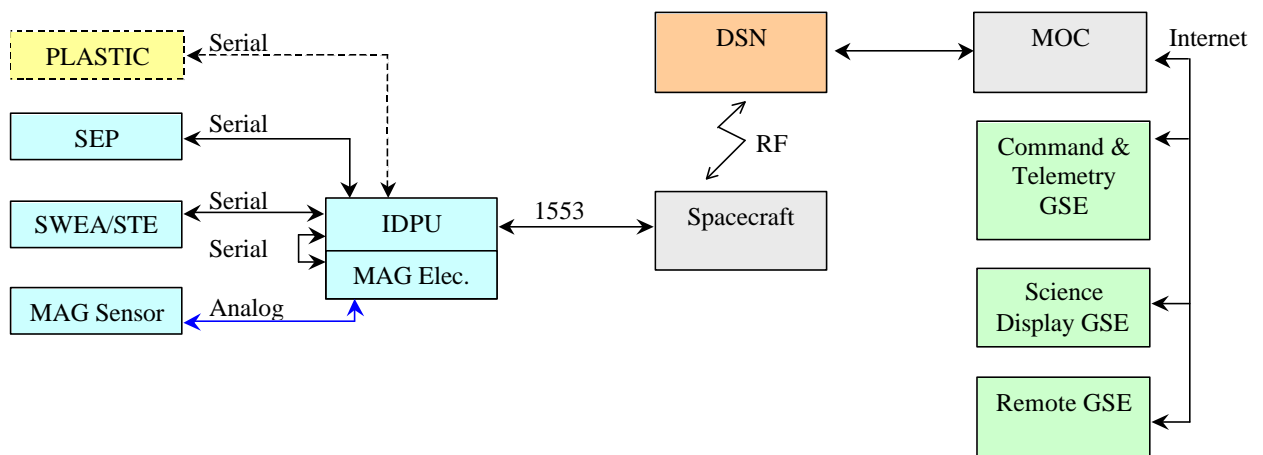
Figure 2.3-3 Spacecraft I&T GSE Setup

Figure 2.3-4 On-Orbit POC Setup

2.4. *Science Display GSE*

The Science Display GSE is an augmentation to the Command & Display GSE designed
to display instrument science data. It will connect to the Emulator or MOC just like the
Command & Telemetry GSE, but only for telemetry, not command. A different Science
Display GSE shall be developed for each instrument to optimally display the telemetry
from that instrument in real time. It normally runs on a separate PC to decouple the
operator from the scientist and provide more space for the user interface on the screen. It
also decouples the development effort, and makes changes to the Science Display GSE
less cumbersome since it is not part of the Critical GSE, and so has less configuration
control.

The Science Display GSE for SWEA, STE, and MAG will be written by UCB. Caltech will write the Science Display program for the SEP instruments. (UNH is responsible for the Science Display software for PLASTIC)

Note that for most instruments the Science Module that connects to the IDPU Simulator will be significantly different from the Science Display that works with the Command & Telemetry GSE because the data format and bandwidth is much different. The exception is SEP, for which the data looks the same (SEP provides packetized telemetry to the IDPU, which it just passes on).

2.5. *Beacon Data Decommutator*

This is a simple program to decomutate the beacon telemetry stream and convert the data to engineering units. It will be developed by UCB to run in the SSC.

2.6. *Level 1 Decomutators*

These will be developed and run at the instrument home institutions. They convert Level zero data to Level 1 data, which is synoptic time series data in engineering units for the science community. The SWEA and STE versions will be developed at UCB. The MAG version will be developed at GSFC by the MAG team. The SEP version will be developed by Caltech.

3. Development Plan

3.1. *Ground Software Identification*

1. ISG Core
2. ISG Science Modules:
    2.1. MAG Science Module
    2.2. SWEA Science Module
    2.3. STE Science Module
    2.4. SEP Science Module
3. Instrument Simulator GSE
4. Command & Telemetry GSE
5. Science Displays:
    5.1. MAG Science Display
    5.2. SWEA Science Display
    5.3. STE Science Display
    5.4. SEP Science Display
6. Beacon Data Decommutator
7. Level 1 Generators
    7.1. MAG
    7.2. SWEA
    7.3. STE
    7.4. SEP

Item 1 will be based on a stripped-down version of the HESSI Spacecraft Simulator GSE, with significant changes in the data interface, and a new command & telemetry database

Item 4 will also be based on the HESSI Spacecraft Simulator GSE, with significant changes in the data interface, and a new command & telemetry database

Items 6 and 7 will be loosely based on existing software from previous programs such as ACE, MGS, and Lunar Prospector.

The remaining items are mostly new code, though made simple by the use of existing user interface code libraries such as Labview CVI.

3.2. *Development Environment and Equipment Needed*

The GSE fall into two categories: the real-time PC-based GSE (items 1-5 above), and Workstation-based data analysis tools (items 6 and 7 above).   The PC-based GSE shall be developed using Labview CVI, which is a C language development environment with a graphical user interface library. The Workstation-based tools will be developed using C and IDL. GSE that can control the instruments shall be tested on prototype hardware prior to its first use on flight hardware.

3.3. *Mission Critical Ground Software*

Mission Critical software is that which can significantly negatively impact the IMPACT project if it fails. In that category are the Command and Telemetry GSE and the IDPU Simulator GSE Core module, since they command the instruments and are responsible for monitoring state of health.

3.3.1. Testing

Mission Critical software will receive extra attention in testing to those features that make it mission critical. Commanding features shall be fully rung-out against ETU hardware prior to use on Flight Hardware.   Safety monitors shall be verified by simulating out-of-range conditions in the ETU hardware.

3.3.2. Software configuration management / backup plan

Mission Critical GSE software shall come under configuration control when it has passed acceptance tests and is ready for use on Flight hardware. Any subsequent changes shall be subject to acceptance testing, and shall be approved by the Project Manager prior to being incorporated into the GSE.

An archive of all released versions of GSE software shall be maintained and backed up on a regular basis.

3.4. *Reviews*

No formal reviews are planned for GSE software beyond what is covered in the PDR, CDR, and similar reviews.

3.5. *Documentation*

GSE software development is documented by this plan. In addition, a brief set of requirements shall be written and agreed to by the intended users prior to the start of coding, and a Users guide shall be written following coding. Other than this plan, these are internal, non-deliverable documents for the use of the IMPACT team.

4. Management Plan

4.1. *Heritage and Reuse*

Estimate the size of the completed software in terms of non-comment lines of source code and the number of anticipated modules. Describe the percentage reuse anticipated from previous projects. Define the heritage of the re-usable code.

4.2. *Staffing Plan*

4.2.1. Level of effort

For each ground software element identified in section 3.1, provide the anticipated level of effort required. (Man-months/quarter).

Gives names of people who will be working on the software. If not known, then explain how the organization will provide experienced personnel when needed for this development effort.

4.2.2. Manpower

Provide quarterly estimates of manpower for the life of the project.

4.3. *Ground Software Development Schedule*

Include SDP, User Guide, Ground Software Releases, Reviews.