Helioseismic and Magnetic Imager
for
Solar Dynamics Observatory

Concept Study Report

Appendix K

HMI SOC Ground System Plan

SU-HMI-S015
1 July 2003

Stanford University Hansen Experimental Physics Laboratory
and
Lockheed-Martin Solar and Astrophysics Laboratory
The principal data flows and products are summarized in Figure 5.1. Level 2 and 3 data are not normally archived, but generated on demand as they are needed. If the higher level data products are not archived, documentation of the algorithms, of the actual code, and of the calibration data used to create them from lower level data will accompany the higher-level data products as ancillary information. This will typically include version and configuration information for a pipeline analysis module in addition to references to the lower level data products and calibration data needed to recreate the higher level data product in question.

7. Data Archive

The approximate data volume of archiving is shown in Table 7.1 "HMI Data Archive". All the raw telemetry data are permanently archived, as are the lev0 filtergrams. Normally higher level number data products are reproduced as needed from the level 0 filtergrams although some higher data products can be archived as deemed appropriate. Whenever a data product is requested in a processing pipeline the system queries the database to determine if the dataset is on-line and where on the disk storage it is located. If the dataset is not on-line, it can optionally automatically be brought on-line and the requesting program is suspended until this occurs.

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
<th>Examples</th>
<th>Rate [GB/day]</th>
<th>Rate [TB/yr]</th>
<th>Cache [day]</th>
<th>Archived [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw</td>
<td>Telemetry</td>
<td>-</td>
<td>600</td>
<td>200</td>
<td>30</td>
<td>100</td>
</tr>
<tr>
<td>0</td>
<td>Filtergrams</td>
<td>-</td>
<td>1000</td>
<td>400</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>Observables</td>
<td>$V_{LOS}$, $B_{LOS}$, $I$, $EW$, Stokes Parameters</td>
<td>400</td>
<td>160</td>
<td>600</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>Reorganized data</td>
<td>Spatial/temporal Samples, Averages; Synoptic maps</td>
<td>10</td>
<td>3</td>
<td>3000</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>Inferences</td>
<td>Global modes, Analysis maps, Farside images, Coronal fields</td>
<td>&lt;1</td>
<td>&lt;1</td>
<td>2000</td>
<td>100</td>
</tr>
</tbody>
</table>

8. Data Distribution

Data distribution is handled through data export request web pages that will allow data selection by time(s) or type(s). Data products are retrieved from resident disk, archived media, or recomputed as required and are made available to the requestor in their indicated storage or via web ftp protocol. All HMI produced data will also be made available via the Virtual Solar Observatory (VSO) project.

9. Integration & Test

With the provision of injecting known telemetry data into the SOC front end, a complete test of data flow and integrity can be performed for each stage of the processing. Standard
regression test suites and validation procedures will be developed to verify processing at each stage through all the system additions and revisions. Throughput and load balancing tuning will be performed.

10. Software Development Environment

The software development environment will consist of Linux workstations with commonly mounted document and source code partitions. The development will be predominantly in C and perl, with possible additions of python, php, and java. The Intel icc will be the likely choice for the C compiler.

All development source code is under a Configuration Management (CM) system. The CM is based on BCS (A Baseline Configuration System) and RCS (Revision Control System). BCS is a set of utilities for maintaining a single baseline and multiple staging areas for a software development effort. BCS provides configuration management functionality as well as the means for multiple users to work concurrently on a common source tree with minimal conflict. RCS is a widely used version control system. The BCS based CM was used successfully throughout the MDI data system development. A trade study will be performed to investigate possible upgrade to a more sophisticated CM system.

11. System Hardware Architecture

Based on experience with MDI processing of similar data, the GDS hardware configuration is currently planned as a processor farm of 50 quad-CPU compute servers connected via a storage area network (SAN) to 400 TB online storage on RAID arrays and a small number of tape libraries providing another 500 TB of near-line storage, a telemetry handling subsystem, and a database subsystem. The GDS has a dedicated data connection to the DDS at White Sands on the one end, and is protected from the public network behind an internet firewall on the other end. A small number of designated access nodes allow remote monitoring of the GDS and provide data import and export functions. (see Figure 11.1) The telemetry subsystem consists of two identical hosts in an automatic failover configuration. Each host has local disk space large enough to receive 30 days' worth of raw telemetry although at any given time only one is active. After their integrity has been verified, the telemetry files are permanently archived and copied to disks on the SAN to which the processor farm has direct, read-only access. The entire subsystem is protected by a UPS and will be able to continue operation for up to 60 days without help from the rest of the GDS.

The HMI pipeline software schedules all data production and analysis tasks to be run on the processor farm, which sees the entire online storage as local disk space through a cluster file system. By using identically configured servers, tape units and SAN switches, and RAID protection for disk arrays, an adequate level of redundancy and data protection will be achieved.
Figure 1.1. Overview of HMI Data Flow
Figure 1.2. HMI Dataflow Concept
Figure 2.1  MDI Pipeline Heritage

- **pe**: pipeline Execution - Command line program that runs analysis modules in a pipe and supplies them DSDS services
- **dsds_svc**: Data Storage and Distribution System server - Provides Oracle DB services to pe and ampex_svc
- **ampex_svc**: Ampex server - Provides tape archive/retrieve services to dsds_svc
- **analysis modules**: Computation modules run by pe (e.g. v2helio_svc)
- **pvm**: Parallel Virtual Machine - Provides for process control and communications in independent environments for each user
- **Oracle**: RDBMS containing data storage tables and information on all datasets produced by analysis modules

Figure 2.2. MDI Software Components
Figure 3.1. SDO DDS – HMI SOC Data Exchange
Figure 3.2a. DDS-SOC Protocol

solar2:/web/hmi/htdocs/development/jim/dds_soc.straw 08May2003

DDS - HMI SOC Strawman Data Flow
--------------------------------

1. The DDS will put and get files on the HMI SOC via an ftp or similar
type program (e.g. FASTCopy).

2. The normal SOC directories for file interchange are:
   
   /dds2soc
   /soc2dds

3. Every minute the DDS writes to /dds2soc a telemetry and quality and
   accounting file named like so and in this order:

   HMI_2002.11.14_10:31:00.tlm
   HMI_2002.11.14_10:31:00.txt

4. The .tlm file will be ~400MB and contain HMI VCDUs in time order and
   without duplication. Real-time delivery will be attempted only once by
   the DDS as files become available.

5. The .txt file will be an ascii file with TBD quality and accounting
   information. Two item of information will the name and size in bytes of
   the corresponding tlm file. The .txt contains an end of file indicator
   field.

6. The SOC will initiate processing the .tlm file when the proper .txt
   is received and verified (txt size, tlm name, tlm size, TBD...).

7. Once an hour on the hour the DDS writes to /dds2soc a delivery
   status file (dsf) named like so:

   HMI_2002.11.14_10:00:00.dsf

   This .dsf contains the file names of all the telemetry files that the
   DDS has that have not yet been acknowledged by the SOC.

   The dsf file will contain ASCII text entries as follow:
   <filename>, <size>, <status>
   <filename> will be formatted as shown above.
   <size> will be the size of the file in bytes.
   <status> will be a numerical filed that can have one of three
   values:
   0 - Delivery Pending; File has not been delivered for whatever
   reason
   1 - Delivered; DDS thinks file has been delivered, waiting for
   SOC ack
   2 - Retransmit; SOC requested a retransmit, retransmit qued
   3 - Retransmit Failed
   4 - Retransmit Successful
8. Upon receipt of the .dsf the SOC will check all the Delivery Pending and Delivered files against what it has successfully received and write an acknowledgement status file (asf) in /soc2dds. The asf file will contain ASCII text entries identical in format to that of the dsf file. The <status> field will have one of 4 values:

- 0 - Delivery Pending; File has not been delivered for whatever reason
- 1 - N/A
- 2 - Retransmit; SOC requested a retransmit
- 3 - SOC Acknowledged, SOC acknowledging receipt of file

Files marked for retransmission may have 2 optional fields:

- <Machine name> <directory>

  - <Machine name> Machine to which retransmissions are to be delivered, must be know by the DDS
  - <Directory> Must be writeable by the DDS

If optional fields are absent, data will be delivered to default location.

Files marked for retransmission will be delivered as bandwidth allows. In general, any file name can be added to the asf for the SOC to request a retransmission. The only way that the SOC makes a tlm file retransmission request is via this /soc2dds.asf file. The DDS removes the /soc2dds.asf file after it retrieves it.

9. At the end of each day the SOC will write a .ack file to /soc2dds containing the tlm file names that have been archived successfully on that day. The DDS will not delete a file that has not been so acknowledged but will notify the SOC manager by e-mail that an acknowledgement is still pending. This is done at say 20 days, well before the normal 30day limit to delete the file. (The 30d delete should not be hardwired. If there is a real problem and disk space and lots of phone calls, the data should not be dropped due to rigidity in cron scripts or the like.)
In summary here are the file types that are exchanged:

<table>
<thead>
<tr>
<th>Extension</th>
<th>Originator</th>
<th>Frequency</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>.tlm</td>
<td>DDS</td>
<td>1 minute</td>
<td>HMI VCDUs</td>
</tr>
<tr>
<td>.txt</td>
<td>DDS</td>
<td>1 minute</td>
<td>Ascii quality and accounting information</td>
</tr>
<tr>
<td>.dsf</td>
<td>DDS</td>
<td>1 hour</td>
<td>DDS delivery status file</td>
</tr>
<tr>
<td>.asf</td>
<td>SOC</td>
<td>1 hour</td>
<td>SOC acknowledgement status file</td>
</tr>
<tr>
<td>.ack</td>
<td>SOC</td>
<td>1 day</td>
<td>tlm file names successfully archived</td>
</tr>
</tbody>
</table>

Footnote:

We will definitively know all the HMI telemetry to expect from interpretation of the hk data. Eventually this will be defined to be the definitive means to know what files to expect from the DDS.

Figure 3.2c. DDS-SOC Protocol
Figure 4.1. Dataset Sequence – Level 0.
Figure 5.1
Figure 11.1. GDS Layout.