CORONAL DIAGNOSTIC SPECTROMETER
SoHO

CDS SOFTWARE NOTE No.1

Version 1.0 28 March 1993

CDS GROUNDBASED SOFTWARE REQUIREMENTS

C. D. Pike
Rutherford Appleton Laboratory
Chilton
Didcot, Oxon OX11 0QX

cdp@astro1.bnsc.rl.ac.uk

CONTENTS

3.3.3 Display the RT telemetry. (Spacetec)

3.3.4 Store the RT telemetry. (Spacetec)

3.3.5 Collect and store the PB telemetry. (Spacetec)

3.3.6 Collect and store the FD telemetry. (TBD)

3.3.7 Produce FITS files from telemetry files. (GSFC)

3.3.8 Produce CDS summary data for planning. (GSFC)

3.3.9 Creation of quicklook (QL) data. (GSFC)

3.3.10 Archiving and distribution of data. (GSFC)

3.3.11 Distribution of software. (RAL)

3.3.12 Access to SOHO (non-CDS) archive data. (GSFC)

3.4 Database and catalogue management. (GSFC)

3.4.1 RDBMS specifications.

3.4.2 Interface specifications with the CDS catalogue.

3.4.3 Catalogue software requirements.

3.4.3.1 Populate the catalogue.

3.4.3.2 Maintain the catalogue.

3.4.3.3 Browse the catalogue.

3.4.3.4 Extract data from the catalogue.

3.4.3.5 Extract CDS contribution to the EOF core catalogue.

3.4.3.6 Extract CDS summary catalogue data.

3.5 Instrument and detector status monitoring. (RAL/MSSL)

3.5.1 Intensity calibration.

3.5.1.1 Monitoring of standard quiet sun areas.

3.5.1.2 Line pair intensity monitoring.

3.5.1.3 Common wavelength regions for NI & GI spectra.

3.5.1.4 Comparison with external data.

3.5.1.5 Comparison with SUMER data.

3.5.1.6 Use SUMER and UVCS data for line pair ratios.

3.5.1.7 Flat-field corrections.

3.5.1.8 CCD dark count.

3.5.1.9 Maintenance of intensity calibration records.

3.5.1.10 GIS zero-order detector.

3.5.2 Wavelength calibration.

3.5.2.1 Pre-launch calibration.

3.5.2.2 In situ self-calibration.

3.5.2.3 Buildup of an 'average' wavelength calibration.

3.5.2.4 Cross calibration with SUMER.

3.5.2.5 Storage of wavelength calibration.

3.5.3 CDS point spread function definition.

3.5.3.1 Source location prediction.

3.5.3.2 Special observation planning.

3.5.3.3 Analysis of the observed profile.

3.5.4 Pointing and alignment calibration.

3.5.4.1 Access to the spacecraft attitude information.

3.5.4.2 Coordinate conversion routines.

3.5.4.3 CDS offset relative to spacecraft.

3.5.4.4 CDS offset relative to other SOHO instruments.

3.5.5 Other detector characteristics.

3.5.6 Stray and scattered light.

3.6 Calibration of raw data. (RAL/MSSL)

3.6.1 Flat-fielding.

3.6.2 Relative intensity correction.

3.6.3 Application of wavelength calibration.

1. Scope of document.

In the recently distributed Software Definition Document (SDD), Harrison et al. gave a broad outline of software policy issues and presented an initial overview of the groundbased software requirements. In this document we specify in more detail the requirements for the groundbased software needed at the EOF, and at other data analysis sites. The software to plan observations with CDS, to process and analyse the resultant data, to maintain databases and catalogues of associated information, to analyse the instrument and detector performance and to maintain and distribute archive data are all considered.

Some of the software requirements may well change in detail, as well as in philosophy, as launch approaches, but it is hoped that the requirements specified here will form a sufficiently broad and accurate foundation upon we can base the design specification and subsequent coding. No account is taken in this document of software that already exists within the solar community and that is already in a usable form or which could be easily adapted to CDS needs. That software can be taken on board at a subsequent design stage and its availability will simply be taken as a welcome bonus from then on. One exception to this is the software written by Spacetec for the EGSE FM. That software has been written specifically for the SOHO/CDS mission and will fulfill our requirements under the headings noted.

In the contents listing, the names of the institutes that have taken on general responsibility for the various software areas have been noted. This is not meant to indicate a rigid allocation, but should reflect currently expressed interests and the likely availability of resources.

2. Requirements overview.

With such a broad remit, it is useful to separate the software requirements into a number of distinct groups or packages. It is of course realized, and indeed intended, that there should be a great deal of interplay among the programs listed under different groupings. For completeness, some task descriptions are listed under more than one category, but where this happens an attempt (probably incomplete) has been made to cross reference them. The categories under which a need for software is envisaged are as follows:

- Observation technical planning.

- Observation science planning.

- Receipt and distribution of data.

- Database and catalogue management.

- Instrument and detector status monitoring.

- Calibration of raw data.

- Data manipulation.

- Science analysis.

3. Detailed functional requirements.

This section contains the functional requirements for each of the software 'packages' listed above.

3.1 Observation technical planning.

This is the so-called mimic CDS software model wherein a potential user may experiment with various CDS setups and gain some insight into the likely data products. There are two possible requirements. Firstly we require a software model of the CDS running on a workstation that will check the feasibility and correctness of the requested setup according to, and limited by, the accuracy of stored parameters of the CDS design. Secondly we may wish to test the effect (eg. timings) of telemetered command sequences. This would require a mockup of the CDHS software/hardware and could be implemented by having spare microprocessors running the onboard software and being addressable by a workstation. Timing loops could replace the actions of the detectors and mechanisms and it might be possible to input synthetic data arrays to simulate data retrieved from the detectors.

3.1.1 Specify the observation.

The user will request a specific instrument setup and data taking mode (ie specify spectrograph, line list, dwell times, scan pattern etc.). All previously uploaded sequences will have been stored so that they can be recalled by name or other specified reference.

There is to be a hierarchy of possible commands that can be sent to the spacecraft from simple mechanism or mode changes to single commands that define complete series. We will need software which will accept commands at any level and check their logic as they propogate to lower levels and then give an indication of the data produced by such commands.

Note that the term **raster** is used to denote a two-dimensional, rectangular scan pattern of observations. This definition is not affected by the existence of degenerate cases eg. where the second dimension has size one, both dimensions have size one or the step size in both dimensions is zero. It is intended that more complex scan patterns will be formed from this basic building block.

3.1.2 Validate the requested setup and report any problems.

The sophistication of the error/inconsistency checking will evolve with time, but the first iteration will involve trapping impossible setups, out of range parameters, inadvisable dwell times, excessive telemetry demands etc..

3.1.3 Suggest remedies for problems.

As experience accumulates, it should be possible for the software to evolve into an expert system where it will identify problems and recommend changes of parameters to resolve them. Initially, recommendations are likely to be limited to reminders of sensible slit choices, limits on slit/mirror positions and integration times, compression schemes etc..

3.1.4 Calculate the telemetry requirements and any data compression or selection factor needed.

If the telemetry requirements exceed the CDS allocation, then data compression/selection will be required. In that case, the software should present the user with the available solutions in terms of line/window selection or further data compression possibilities. The choice of compression technique will be made as a parameter to be uplinked.

3.1.5 Calculate the appropriate dummy data.

For a number of purposes, including observation planning, we require the ability to calculate dummy or synthetic datasets. These will be arrays containing physically realistic spectral data. We require the ability to write the dummy data to FITS files whence they can be manipulated in the guise of real data. It should also be possible to write the data to simulated telemetry files which can subsequently be processed into standard FITS data files. This will provide the basis for an end-to-end test of the processing.

Realistic noise and/or various other special effects (eg spatial/velocity stucture) should optionally be added to the synthetic data.

3.1.6 Moving feature tracking.

The means by which feature tracking will be implemented is still TBD (ie on board versus ground-based). If the software and calculations are to be ground-based, we will need tools to help design observational sequences for moving targets (see section 3.5.3.2 for details in the context of calibration). Mechanism movements can be applied asynchronously to exposure commands and so periodic adjustment can be made during an exposure. Exactly what positional offset information would be associated with such data is as yet TBD.

3.2 Observation science planning.

Software is required to give the user as much relevant information on the state of solar activity as possible. CDS is also required to provide planning information for users of other SOHO instruments. We will require the ability to peruse recent CDS data, to relate it to data from other sources and to make CDS summary data available to non-CDS users. In more detail we require:

3.2.1 Access to the CDS database and catalogues.

Details of all previous CDS observations need to be accessible so that users can compare different instrument setups and their resultant data (see section 3.4).

3.2.2 Access to all SOHO summary data.

We will need to be able to read or receive the summary data (this includes images from some instruments) provided by CDS and all other SOHO experiments.

3.2.3 Access to non-SOHO data.

We will require access to as many external observation datasets as possible or relevant. Table 3.1.6 of the Science Operations Plan (version 3.1), hereafter called the SOP, gives a selection of data which will be available at the EOF.

Also implied is the ability to display and interact appropriately with these various kinds of data. For instance, we will need routines to overlay proposed CDS target areas on solar disk images and conversely to specify interactively a region of interest and for the corresponding CDS raster area to be defined automatically. Note the coordinate system to be used SOHO-wide is defined in section 2.3.2 of SOP.

3.2.4 Access to SOHO ancillary data.

The data which the project will provide as ancillary data is defined in Table 3.1.1 of the SOP. This includes orbital elements, spacecraft attitude and timing data. These will be needed for planning purposes, so programs to retrieve and interpret these data are required.

3.2.5 Packaging of the proposed observation.

We require software to gather other associated data that go with a proposed observation eg. campaign description, target identification, observer reference. A form-filling program would be an appropriate interface.

3.2.6 Communication with ECS.

The CDS plans, both on the short and long term, have to be sent to the ECS for communication to the spacecraft. The Spacetec software requirements document (version 1/1, section 3.4.1) describes code that will be written to perform this. Any other ground-based planning software must be interfaced to that. Two specific procedures required are:

3.2.6.1 Final observation details.

Once a CDS observation is defined scientifically and thereafter in terms of the commands to control the instrument, various parameters will have to be further associated with the observation (eg sequence, study, observer identifications) and the whole will then be entered into the planning files from where it will be communicated to the ECS.

It must be possible to modify the plan after transmission to the ECS up until such time as the observation plan is uploaded to the spacecraft.

3.2.6.2 Activity plan.

Section 2.1.4.1 of the Interface Control Document gives details of the forward planning information that must be supplied to ECS. We will need tools to allow the CDS team to compose and transmit an activity plan at a level of detail appropriate to the planning timescale involved.

3.3 Receipt, management and distribution of data.

There will be three forms of telemetry to be dealt with, the realtime (RT), playback (PB) and final distribution (FD) telemetry. Each has slightly different requirements, but some of the software will be common to all three. Each type of telemetry will be stored in a 'telemetry' file prior to being processed into FITS files.

3.3.1 Capture the RT telemetry.

During the periods of realtime data reception, software will be running to capture the RT telemetry as supplied by the ECS.

3.3.2 Calibrate the housekeeping telemetry.

The housekeeping telemetry values must be calibrated, ie converted to engineering units, according to the latest telemetry database values. This function will apply to all forms of telemetry.

3.3.3 Display the RT telemetry.

The housekeeping information should be extracted and displayed on the realtime pages. Representative science data (spectra and/or selectable image buildup) should be extracted and displayed in realtime.

3.3.4 Store the RT telemetry.

The RT telemetry will be stored on disk for immediate use.

3.3.5 Collect and store the PB telemetry.

The PB telemetry must be copied from the ECS as soon as it becomes available and stored in the PB telemetry files. After a period of time (TBD) the telemetry will be archived to a suitable medium (TBD).

The PB telemetry will be treated in the same way as RT, so that the same software applies.

The PB science data becomes available on two timescales, immediate and few hours later, the latter is assumed to supersede the first and the RT telemetry completely.

3.3.6 Collect and store the FD telemetry.

The medium upon which the FD telemetry will be exchanged is still to be determined though it is likely to be ISO 9660 format CD-ROM. This will be discussed at the next SOHO Splinter meeting.

3.3.7 Produce FITS files from any of the telemetry files.

Creating FITS files from the telemetry will involve extracting all the details associated with the observation (spacecraft attitude, calibration file identification, observing logs etc.) from the planning files. The science data must be extracted from the telemetry file and the associated details written to the FITS headers. At the same time, characteristics (eg basic statistics) of the actual data, together with appropriate auxiliary information, must be supplied to the CDS catalogue and fed back to the planning files.

In order to avoid confusion, we note that data written to FITS files will have been subjected to some calibration (decompression, standard wavelength calibration, time adjustment to geocentric) and are, therefore, no longer level-0 data. These FITS files will be referred to as level-1 FITS files.

Each FITS file will contain data from one and only one raster. A raster is defined in section 3.1.1. Non-rectangular scans will therefore have to be composed of simpler (possibly one-dimensionally degenerate) rasters. A long series of monitoring observations with no position offset can also be broken into manageable pieces by defining it as a series of (N,1) rasters with zero spatial offsets in each coordinate giving a series of N observations of the same area (neglecting solar rotation).

The FITS files will be named by assigning a single character followed by seven digits to form a sequential numbering system. The filename extension will in general signify the type of file (eg. .FITS or .FTS if limited to three characters).

Observations that have, for whatever reason, been interrupted (eg. by inter-instrument flags or ground command) will have their data area padded with invalid data values. Keywords in the FITS file header will signal that the data are incomplete. There will also be a 'data invalid' keyword which will be used to identify data that, although present and readable, are intended never to be used.

3.3.8 Produce CDS summary data for CDS/SOHO planning.

The summary data to be provided to EOF by each instrument are defined in section 3.1.4 of the SOP. According to this, all CDS has to provide is a list of observations obtained (modes, times and locations).

We will also provide summary *images* ie. intensity or intensity ratio maps at given wavelength(s) whenever they are available.

We need software to extract the CDS summary data and to make it available to EOF within 24 hours of receipt. The non-image summary data will be a product of the catalogue software (section 3.4) but the images will be related, if not identical, to the quick look images (see section 3.3.9).

3.3.9 Creation of quicklook (QL) data.

The Spacetec software will be capable of displaying the science data directly from the telemetry in a variety of formats. To avoid confusion, this should perhaps be called simply the real-time image display . In the current context, the term *Quick Look* refers to special images and FITS files created by IDL routines from the telemetry files. These images can then be displayed, as with the data from the standard level-1 FITS files, in order to assess, in near-realtime, the incoming data. The files can subsequently be used for planning and also passed to other groups as part of the CDS summary data product.

The QL data need not be fully calibrated, on the other hand they clearly should not be so 'raw' as to be misleading since the QL data will be an important planning tool both for CDS observers and for other SOHO users.

3.3.10 Archiving and distribution of data.

The CDS science archive will consist of copies of the level-1 FITS data files and the associated catalogues. The medium for the longterm storage is TBD. If a tape medium is used, the recommendation is to organize the archive using detached SFDU headers. A similar system could be used to archive the raw FD telemetry files.

We will need software to create the archive on the export medium, software to read data from it and software tools to aid in user-generated requests for extraction of data from it. Although, in a sense, it comes at the end of the list of products discussed here, that last item should in practice be given a high priority as it will be the first contact many users have with the CDS data system. First impressions are often very long lasting!

3.3.11 Distribution of software.

The master copy of the IDL software will be kept and maintained at the EOF. Regular distribution of the software to cooperating institutes will be required and subsets of it will be passed to other SOHO experiment groups.

We will need tools to write the software to a suitable medium and to allow a variety of platforms to read and install it. Utilities exist for reading UNIX TAR files on VMS machines for instance.

3.3.12 Access to SOHO (non-CDS) archive data.

We need to negotiate with other instrument teams about means of access to their online and archive data.

3.4 Database and catalogue management.

The CDS catalogue will be maintained using a commercial relational database management system (RDBMS). The actual software package to be used has not yet been determined. There are requirements that can be placed on the RDBMS software package itself, as well as on the software written using that package. These two sets of requirements will be considered separately below.

The Spacetec document lists software to create ASCII files from the telemetry and command databases and to convert them to EXCEL readable files. However, my (W.T.T.) understanding is that these files will be used during AIV, and will not have relevance to operations, or to the CDS on-line catalogue.

3.4.1 RDBMS specifications.

It has been agreed by the SOHO Science Operations Working Group (SOWG) that each team will use a commercial RDBMS package to maintain their individual catalogues, and that this package will be based around, or support, the ANSI standard Structured Query Language (SQL). This implies compliance with a number of technical specifications, such as being able to join tables, which will not be gone into here. It is assumed that all commercial SQL products meet some minimum level of functionality which are common to them all. However, there are a few additional technical specifications which must be met by this software that it is valuable to enumerate. These are:

◦ Must support embedded SQL (ESQL) commands within the C language.

◦ Must provide a standalone fourth generation language (4GL) interface which interacts with the user through a forms-driven interface.

◦ The interactive interface must allow for spawning OS shell commands.

◦ Must be capable of importing and exporting flat ASCII files, both inside and out of the interactive interface.

◦ Must provide security so that unauthorized users cannot modify the database information.

◦ Must support a client-server model, with one workstation acting as the server, and the other workstations acting as clients to that server, through a TCP/IP interface.

◦ Must be supported on a wide variety of Unix platforms.

◦ Software must not have to be rewritten to work on different platforms.

◦ Client software must be supported under DOS.

◦ It is possible that various sites would want to replicate the CDS catalogue software used at the EOF. This may have an impact on which platforms would be required to be supported.

It is very important that the CDS on-line catalogue software be easy to use, since this will most likely be the first piece of software that a novice user of the CDS data will see. This, of course, is ultimately the responsibility of the programmer, but choosing an RDBMS package that facilitates this task is very important. Since terminal interfaces are highly variable, the best way to achieve this end is through an X-windows interface. Thus, the software must:

◦ Provide a completely X-windows based display, and not work through an intermediate environment such as xterm. It would be best if the software was built on the Motif toolkit.

◦ Provide complete support for the mouse, so that the user does not need to use the keyboard to perform any action other than data entry.

◦ The software must work under a number of different windowing systems. In particular, it must support both X-windows on the Unix workstations, and Microsoft Windows on DOS-based PCs. It should take on the native look-and-feel of whatever windowing system it is being run on.

◦ The same software should also be capable of working with an ordinary ANSI terminal.

3.4.2 Interface specifications with the CDS catalogue.

All catalogue data will be passed to the RDBMS package as flat ASCII files. The exact format of these files will depend on the database package selected, but will most likely consist of a series of rows of catalogue entries, with the data stored in fields of precisely defined size, so that they form aligned columns in the file (e.g. as written by FORTRAN FORMAT statements). Another, less desirable, approach may be to allow the size of the data fields to vary, and separate them with commas.

Data extracted from the RDBMS system for use by other software will also be in the form of flat ASCII files.

3.4.3 Catalogue software requirements.

Once the appropriate RDBMS software package has been selected, the specific software needed to populate, browse, and extract data from the catalogue will need to be written. This software must be able to:

3.4.3.1 Populate the catalogue.

The software must be capable of reading in flat ASCII files containing catalogue information. This process could possibly be performed by a shell script, but it may be better to write this software with ESQL and C. Error checking must be performed to ensure that database reliability is maintained. The program must be capable of proceeding without human interaction, reading in fresh catalogue information as soon as it is available.

3.4.3.2 Maintain the catalogue.

The software must provide for the capability of a human operator to interactively edit the catalogue when necessary. Security safeguards must be implemented so that only authorized users may write into the catalogue. Logs must be maintained of all editing operations.

3.4.3.3 Browse the catalogue.

The software must provide a user-friendly interface that will allow for scientists to browse through the database. The software must be capable of allowing the user to flexibly design queries to the database, through a menu or forms-driven interface. The software must also be able to extract data from other instrument teams' catalogues, through either gateways or importing ASCII files, and to correlate that data with the CDS catalogue. Results of queries must be viewable on the screen, with the options of writing the data to a flat ASCII file, and of sending the data to a printer.

3.4.3.4 Extract data from the catalogue.

As well as the interactive interface to the catalogue, there must also be support for automatic extraction of catalogue data. There must be a mechanism for parameterizing the selection criteria for the data to be extracted.

3.4.3.5 Extract the CDS contribution to the EOF core catalogue.

The data to be incorporated in the catalogue maintained on the EOF core system must be extracted from the CDS catalogue by an automated program that can proceed without human interaction.

3.4.3.6 Extract the CDS summary catalogue data.

The catalogue data to be incorporated into the monthly SOHO summary data must be extracted by an automated program that can proceed without human interaction. The data must be written out in the proper format, as yet undefined.

3.5 Instrument and detector status monitoring.

Observations will be taken specifically to monitor the instrument and detector performances. The software in this section should be able to extract the relevant data and to analyse them for particular calibration purposes. It must also be easy to associate a particular dataset with the calibrations appropriate to it. The details of all the calibrations must be readily available to all analysis software and in some cases to the planning software (eg. current sensitivity calibration for exposure time calculations). It should be borne in mind that the instrument/detector engineers will want access to calibration data and in some circumstances this may mean access to the telemetry data rather than just the science data.

3.5.1 Intensity calibration.

There will exist an absolute calibration from pre-flight laboratory work and more observations will have to be taken *in situ*. There is the possibility that some external calibration will be provided during the mission by rocket-borne instruments. Various schemes for the internal calibration are discussed in the BB and in many cases the software ought to be able to perform the analysis with little or no user interaction. The intensity calibration consists of two parts, an overall throughput calibration and a detailed detector pixel-to-pixel calibration (the flat-field).

Methods which have been suggested for achieving a reliable intensity calibration are:

3.5.1.1 Monitoring of standard quiet sun areas.

It may be possible to monitor the intensity calibration using regular observations of similar quiet-sun areas. We need interaction here with the planning/summary data. A list must be maintained of relevant observations or the relevant observations must be easily identifiable from a catalogue search. The ability to select a new area for observation must be aided by the provision of the necessary data, for example, the latest CDS and ground-based images. We need to define what parameters of the data are to be deduced and recorded (total counts, line counts etc.).

3.5.1.2 Line pair intensity monitoring.

The BB lists several pairs of lines whose relative intensity is fixed and accurately known. The actual observed relative intensities will form part of the intensity calibration. We must maintain line lists and have the ability to extract the line ratios automatically from any appropriate data (even if they were not obtained specifically for calibration purposes).

3.5.1.3 Common wavelength regions for NI & GI spectra.

The wavelength overlap between NI & GI spectra will allow an internal CDS comparison. Software should be able automatically to analyse these data. Further overlap regions are usable if both first and second order spectra are used. The ability to distinguish orders and to handle them accordingly will be needed.

3.5.1.4 Comparison with external data.

There are suggestions that a comparison (of line strengths?) with, for instance, radio flux data will aid in calibration. We need access to external data (see section 3.2.3) and software to do the comparison.

3.5.1.5 Comparison with SUMER data.

NIS2 & GIS4 wavelength regions overlap with SUMER. If simultaneous (or nearly so) observations can be arranged, we need the ability to receive (or read) SUMER data and to compare it with CDS data.

3.5.1.6 Use SUMER and UVCS data to obtain further line pair comparisons.

The BB has a list of line-pairs that may be useful if appropriate SUMER and/or UVCS data are available. This requires the ability to read SUMER and UVCS data and to perform a comparison with techniques still TBD.

3.5.1.7 Flat-field corrections.

An initial flat-field correction will be provided from laboratory data for the VDS. This correction can be monitored to some degree by data from within the line profiles observed (by using the wise slit to determine the effects of particular lines 'burning' into the detector), but much of this process is still TBD.

3.5.1.8 CCD dark count.

Although the VDS dark count is expected to be negligible, there is always the possibility that problems could arise - what if the cooler fails for instance? We will require the ability to analyse any 'dark' exposures (ie. CCD integration with the MCP voltage off).

3.5.1.9 Maintenance of intensity calibration records.

The latest, approved intensity calibrations must be written to files for general use. A system of logical names for the directories involved must be set up. An association between each raw observation and its calibration must be kept.

3.5.1.10 GIS zero-order detector (ZOD).

Data from the ZOD (a near-infrared detector) might be of use in the calibrations, but is most likely to be used for pointing offset calibration.

3.5.2 Wavelength calibration.

The conversion of pixel numbers to wavelength for each spectrum is a prerequisite for most science analysis. Without the possibility of an on-board wavelength calibration source, this calibration will have to be derived from a mixture of pre-launch data and careful monitoring of the observed spectral data.

3.5.2.1 Pre-launch calibration.

Laboratory measurements of HC lamp lines will provide an initial calibration. This, or subsequently updates to it, will be entered into the level-1 FITS files and will be read by the analysis programs.

3.5.2.2 In situ self-calibration.

Assuming that the problems caused by dynamics of the observed plasmas can be catered for or eliminated, it may be possible to monitor the wavelength calibration from normal science observations. The BB suggests that 'hot' coronal lines may be static and hence most easily usable. This requires research on some real data before much progress can be expected.

3.5.2.3 Buildup of an 'average' wavelength calibration.

Despite the problems of dynamics mentioned above it may still be possible to build up an 'average' wavelength calibration by careful selection of lines or parts of lines. Depending on the eventual complexity of this process it may be possible to automate it.

3.5.2.4 Cross calibration with SUMER.

SUMER will observe chromospheric lines in the overlap region 517-787 Angstroms. These observations could be used to derive or check the wavelength calibration in the NIS2 & GIS4 windows. The use of second order lines will then allow calibration of NIS1 & GIS2 by method or methods TBD.

3.5.2.5 Storage of wavelength calibration.

If the wavelength calibration changes significantly with time, each observation must have an associated calibration ie. catalogue entry, and this must be easily retrievable for application to the data during analysis. This latter point is true, of course, of all of the calibration data.

3.5.3 CDS point spread function definition.

The PSF achieved pre-flight will be recorded and available.

It is required that the profile be checked and then periodically monitored during the mission. Observation of a bright UV stellar source passing within one degree of sun centre appears the most promising method. The BB (section 4.4) gives some possible candidates and their predicted count rates. This should be explored further.

In any case, stellar sources will have a motion relative to the sun of approximately 0.04 arcsec/sec. Finding the source and centering it in the aperture will be non-trivial. Software that might possibly be required on the ground under the current remit is:

3.5.3.1 Source location prediction.

We will have access to the SOHO ancillary data (see SOP 3.1.3 and section 3.2.4 here) and this will provide details of the spacecraft's orbital parameters and attitude. We will then need to have software to calculate the source-sun offset at intervals ranging from days (for planning) to seconds (for observing).

3.5.3.2 Special observation planning.

We will need special observation planning software to create a super-sequence or series of sequences to cater for the source acquisition and tracking. If the required integration time turns out to be very long (tens of minutes) and the results are deemed worthy of that commitment then we will need near-realtime correction of the CDS pointing (see section 3.1.6) to cater for both the sidereal motion and any spacecraft attitude changes. Note that the CDS's own sun sensor will give positional information (except roll) which could be used.

3.5.3.3 Analysis of the observed profile.

The PSF is currently described as a gaussian with half-energy width of approximately 2 arcseconds, and beyond about 4 arcseconds, an exponential tail out to several arcminutes. No doubt these details will change, but we will need programs to analyse such a profile.

3.5.4 Pointing and alignment calibration.

In most cases, for the complete analysis of CDS data we will need to know the absolute location of the aperture as accurately as possible. Many scientific analyses will depend on intercomparison of data from CDS and other SOHO and groundbased instruments. A constant check must, therefore, be kept on the absolute pointing (ie the CDS offset relative to the spacecraft attitude) and the pointing relative to other SOHO instruments. This implies a need for software to provide:

3.5.4.1 Access to the spacecraft attitude information.

Software to derive (and interpolate) the spacecraft attitude will be needed as a prerequisite to matching images, overlays etc. (see section 3.2.4). The attitude data themselves are to be provided at the EOF as ancillary data.

The attitude value to be associated with a particular observation is still TBD. It will depend on whether we can use the CDS offset pointing detector information or must wait for the spacecraft data.

An indication of the variations in attitude experienced during the raster will be kept in the catalogue.

3.5.4.2 Coordinate conversion routines.

The SOHO coordinate system is defined with its origin at disk centre and positive axes to the North and West, measured in arcsecs. We must provide the means to convert between this and other systems (eg. heliographic latitude and longitude) as well as between solar coordinates and the mirror/slit/offset pointing coordinate system. This will be complicated by the fact that the CDS offset pointing adjustments introduce a slight roll relative to the mirror/slit coordinates.

Note that an accuracy of 4 arcseconds (TBC) is needed in the calculated position of solar features for inter-instrument flag messages.

3.5.4.3 CDS offset relative to spacecraft.

The CDS-spacecraft offset must be periodically monitored in order to ascribe absolute locations to CDS observations.

The best ways of doing this are still TBD. One possibility is to take scans of opposite limb regions to locate the sun-centre or we could make use of the CDS sun sensor and/or the GIS zero order detector data.

Whatever the technique, we will need software to analyse CDS data and to compare it with the spacecraft attitude. The results must be saved in the catalogue.

The CDS offset will be a slowly varying function of time so it can either be recorded in each FITS file, recorded in a separate calibration or be stored in the catalogue.

3.5.4.4 CDS offset relative to other SOHO instruments.

Collaborative observations of well-defined surface features or limb regions can be compared to obtain the relative offsets from relevant instruments. We need, in particular, to have access to SUMER and EIT data for this analysis, and to have software to perform the comparison. (See observing proposal in BB for suggested wavelengths and sequences.) This offset information must be stored in the same way as for the CDS-SOHO values discussed above.

3.5.5 Other detector characteristics.

Detectors are wonderful things. There are likely to be other 'features' or characteristics we need to monitor and calibrate but they are as yet TBD. The SPIE papers by Thompson et al. (VDS) and Breeveld et al. (GDS) describe some of the performance characteristics of the detectors.

3.5.6 Stray and scattered light.

The current aim is to reduce the scattered light from the CDS components to a level below that scattered from the gratings. The BB (section 4.4) gives some numerical estimates.

We need to have software to search for, detect and monitor the existence of any stray light.

The VDS performance paper gives details of scattering within the VDS itself. This may be important for very bright, spatially-small sources and may require corrective action.

3.6 Calibration of raw data.

The first stage in most analyses will be to remove, as far as possible, the instrumental signature from the raw data.

3.6.1 Flat-fielding.

All VDS data will require flat-fielding on a pixel-by-pixel basis. The GIS detectors will also have a similar correction which is nominally only a one-dimensional correction.

3.6.2 Relative intensity correction.

There may be non-detector intensity corrections required which will be functions of the ageing/changing alignment of the optics. Various spectral regions could be 'normalized' by factors deduced from the calibration work described in section 3.5 or the factors could be folded into the flux calibration to be applied later.

3.6.3 Application of wavelength calibration.

Data which retain their spectral dimension will require a wavelength calibration (the option to make this a velocity scale should be provided). The offsets from a linear scale will be stored in the FITS files, although it will probably be convenient to parameterize this for use in the analysis routines.

3.6.4 Flux calibration.

For comparison with other sources of data, and for modelling, a flux calibration must be applied. This will be based on the pre-flight laboratory measures adjusted according to the results of in-flight monitoring observations. These will be either internal to CDS or relate to specially designed external observations (eg rocket flights).

3.7 Data manipulation.

The software described in this section is that which is required as a precursor to, and then an integral part of, the science analysis. It covers all the methods of data visualization and integration (in a non-mathematical sense). In truth, the list is probably almost infinite since it is impossible, and ultimately not very useful, to attempt to specify every last variation and nuance of data manipulation. IDL itself provides a wealth of routines and the main purpose of the software in this section will be to package lower level software to facilitate the handling of data in a way suited to CDS. In general, we should be careful to avoid overpackaging IDL since this can very quickly detract from one of its main advantages and joys, its flexibility.

As a matter of semantics, we will reserve the word **plot** for the handling of one-dimensional data (eg intensity versus wavelength) and the word **display** for the two-dimensional case (eg intensity as a function of x,y location).

The main capabilities to be provided are:

3.7.1 Data input to IDL.

The FITS files (section 3.3.7) will have already been created and these files must be read into IDL. All the auxiliary information that goes with the spectral data should also be read in at the same time. This will be stored in an IDL data structure, the contents of which are still TBD but which will match closely the FITS header information.

3.7.2 Deselection.

If the data taken on the spacecraft had to be **compressed** (a numerical process) before transmission, the correct decompression will have been applied before the creation of the level-1 FITS files (see section 3.3.7). If the data were merely **selected** (ie. a specific line list was used) to limit the telemetry requirements, the deselection should be an option that is available at any stage of the analysis but is most likely to be required at an early stage.

Deselection, as envisaged here, may actually be no more than a data-handling convenience in the sense that it might, for example, mean that only one data array, as opposed to several smaller ones, is being manipulated at any one time or that the array being used does not contain discontinuities in either the spatial or spectral dimensions. The information required to perform the deselection will be implicit in the individual data files themselves and upon reading the data, that information must be loaded into the accompanying data structure.

3.7.3 Plotting and displaying data.

The most general data set likely to be encountered is a three-dimensional array in which the dimensions correspond to two spatial ($\mathbf{x}$,$\mathbf{y}$) and a spectral (wavelength or velocity) coordinate (λ). Each spectral entity will also have a time tag ($\mathbf{t}$) associated with it. A series of such data cubes taken over a period of time could also constitute a 4-d array. We require the ability to plot or display data from any combination of these dimensions and to have the option of limiting the range of parameters used in the non-plotted or non-displayed dimension(s). The normal ability to limit the range of the plotted/displayed parameters is, of course, retained.

For instance, from any NIS or GIS data cube we might require to plot/display:

- Intensity against wavelength (x,y,t ranges are user-defined)
- Intensity image (2-d) (λ,t ranges are user-defined)
- Intensity against time (x,y,λ ranges are user-defined)
- A movie of images against time (x,y,λ ranges are user-defined)

The data selections which are implicit in the above examples also imply that various operations can be performed on the data before the data are displayed/plotted. Displaying a line intensity ratio image, for example, would simply reduce to a matter of dividing the data selected in two cases of the second example above before displaying them as a single image.

The means by which the user may define the required limits in each of the four dimensions must be considered. At its lowest level, this could be a simple numerical definition of start and stop values. Variations on this will allow the specification of well-defined, but possibly irregular, regions in the required coordinates (eg the interactive definition of an irregular region of spatial pixels to be used in an 'integrated intensity' plot against time or wavelength). It should also be possible to define the window in one or more dimensions by reference to the intensity values (eg use only wavelength range where the intensity is greater than some limit).

A useful utility would be a program to assist in the visualization of the overall contents and structure of a FITS data file.

3.7.4 Specials.

There will be cases in which the data format is well-defined and stable (eg NIS synoptic sequence) and where the plot/display requirements are equally stable. For these instances, it would be appropriate to build up more elaborate, automatic plotting/display procedures from the modules generally available.

3.7.5 Storing intermediate data.

It should be possible to write out processed data to FITS files for transport to other places and times. Care must be taken that all the necessary ancillary information is saved as well. For local, shortterm storage of intermediate results it might be sufficient to use the IDL 'save' command for the appropriate variables and structures.

3.7.6 Propagation of error arrays.

Any numerical manipulation of the data (including calibration) should keep track of data errors and their propogation.

3.8 Science analysis.

The subject of the science analysis of CDS data is a vast one, but it is the intention of the project to provide the means to perform most of the necessary basic analyses together with some of the tools required for more in-depth or esoteric studies.

3.8.1 Basic products.

Section 3 of the SDD included a list of some basic science product requirements. These are repeated here for completeness and are as follows:

- The calculation and mapping of temperature from specific line ratios.
- The calculation and mapping of density from specific line ratios.
- The mapping of emission measure.
- The production of flow pattern images (Dopplergrams).

The first three items of this list presuppose the existence of relevant atomic data in order to transform line ratio maps into the physical parameter maps. These data should be defined early, calculated using ADAS and stored for general use in defined directories.

3.8.2 Spectral line fitting.

When more information other than a total intensity is desired about a spectral line, some form of functional approximation to the data is needed. Indeed, this may still be a requirement for intensity estimates if the line is blended. When the complication of overlapping profiles, be they from different lines or different velocity components of the same line, is involved, spectral line fitting is usually essential.

3.8.2.1 Specifying the function.

We will require software to allow the user to 'design' the function to be fitted in terms of its functional form (Gauss, Lorentz, combination ...), and to supply initial estimates of the parameters (position, width, intensity ...) and the number of components to fit. Some or all of this should be done interactively according to feasibility and user preference.

3.8.2.2 Fitting the function.

The means with which to fit the chosen function must be provided. The IDL user library routine *CURVEFIT* could be packaged or any other suitable routine adopted as appropriate.

3.8.2.3 The output products.

The fitting can be applied to single spectra or a combination of spectra, the latter to be defined using the same techniques as the data window selection process used prior to plotting/display (section 3.7.3). The fitted parameters will be stored in a data structure whose dimensions match the input data. Single variables will result from fitting to a single spectrum and multi-dimensional arrays of parameters will result from fitting to data in a data cube. Hence it will be trivial to display maps of the profile parameters; the spatial variation of the line width, for instance.

3.8.2.4 Conversion to physical units.

The parameters from the profile fitting will need to be converted, where necessary, to physical parameters. Account should be taken of the contribution of the instrumental PSF (section 3.5.3.3).

3.8.3 Calculation of synthetic spectra.

We shall require the ability to calculate synthetic spectra in the wavelength range 150-1600A for comparison with the CDS and SUMER observations. This facility should have different levels of complexity.

1) "Quick look" synthetic spectra, compiled from previous solar observations, are already available. These should be used for planning purposes (eg observing sequences for different solar features) but not in analyses of the actual observations.

2) Theoretical, or semi-theoretical G(T) functions (see Blue Book) have been produced by several groups (eg Arcetri and NRL). These usually assume simple atomic models and near equilibrium conditions. If checked and updated these would be adequate for many straightforward analyses. ADAS (see section 3.8.4.) could also be used to produce G(T) functions and hence theoretical spectra for general solar models (quiet Sun, active region, coronal hole etc) in equilibrium.

3) Synthetic spectra could be produced by interfacing ADAS with solar models for different features and solving the time dependent ionisation/recombination and level population equations.

3.8.4 The Atomic Data and Analysis Structure (ADAS).

The ADAS package provides a method of atomic data analysis, storage and processing. It is a very powerful suite of subroutines, which solves the collisional radiative model for equilibrium or non-equilibrium conditions. The atomic data input needs to be optimised for application to the solar UV spectrum. The output products will be available for use by other CDS/SUMER analysis routines.

The electron excitation rates required to analyse the CDS and SUMER spectra were assessed at a workshop held in March 1992. The proceedings will be published as a single issue of ADNDT. The recommended data should be compiled in a machine readable form, input into ADAS, and made accessible to other CDS software. In order to provide the most comprehensive and accurate atomic data possible for the analysis of CDS/SUMER data, the ADAS package is being converted from IBM Fortran and provided with an IDL front-end. The heart of the calculations will still be performed using Fortran code transported from the JET machines. The output products will be available for use by other CDS/SUMER analysis routines.

Work on the Fortran conversion and the package's interface to IDL will start soon and a version is likely to be available for testing by April 1993.

3.8.5 The derivation of differential emission from line intensities.

R A Harrison and A M Thompson have reported (RAL-91-092) on the performance of a number of integral inversion codes used in the calculation of differential emission measure (DEM) from line intensities. It is anticipated that the Glasgow code will be the one which the CDS project will support. Support in the present context is taken to mean the conversion, as necessary, of the Fortran routines to run on the CDS machines and the provision of an IDL interface. Other DEM codes (eg MSSL, Arcetri) might be made available and should also be considered. The emphasis should be on providing a suitable interface between the atomic physics codes, the DEM codes and other CDS software.

3.8.6. Basic interpretation software.

Electron density and temperature diagnostic line ratios will form an important part of the CDS observations. It is necessary to specify sets of optimum spectral lines for different solar features. The best atomic data for these lines should be used to compile diagnostic plots and to enable "quick look" electron density contours to be produced from the solar observations.

More accurate analyses of the observations would require interfacing the atomic models (ADAS) with the solar models to provide physical parameters as a function of position and time. The dynamic nature of the solar atmosphere requires an iterative approach. It is likely that various groups will want to develop their own theoretical models, so the emphasis should be on how best to interface the different components of the software.

4. General programming and documentation considerations.

It has been decided that the bulk of the analysis software will be written in IDL. The known exceptions to this are the number-crunching routines in the ADAS and integral inversion packages (Fortran), the Spacetec software (C) and some database software.

As always, it is essential that the enterprise as a whole keeps a fairly tight rein on the quality of code produced. This should occur through peer review of its correctness, usability, supportability and documentation.

Some general points worth considering are:

∘ Use the features of the IDL language, especially the array handling to the full.

∘ Chapter 12 of the IDL Version 3 User's Guide gives a few handy hints on efficiency.

∘ Use only procedures and functions to help maintain modularity and flexibility. Take care that function names are not likely to be interpreted as variables.

∘ Use the circulated CDS-IDL documentation template header for all code. A blank version of this has been provided as a file for inclusion from any editor.

∘ DOC_LIBRARY routine is extremely useful, but is often inappropriate. Many times it would be nice to be quickly reminded or informed of individual items from the documentation and so it would be useful to have means by which to access these individually rather than en masse. In this context, note the existence of several widget-driven documentation routines written by Dominic Zarro.

∘ One very useful practice is to standardize the code so that any IDL procedure called with an inappropriate number of parameters will automatically return with a message which gives the correct calling format.

∘ To provide hardcopy user and/or programmer documentation (not to be confused with a users' manual which is a separate, but no less important, requirement) we should produce listings of the documentation template from each unit of code. In addition, we should provide separate listings of the **one-line** explanations sorted in as many ways as is considered useful eg alphabetical or by category. This can easily be done automatically, but only if the same documentation template is used from the outset.

∘ Care should be excercised in the choice of procedure/function names. At least we do not have multi-instrument confusion to worry about, but nevertheless names should be meaningful and specific. Uniqueness will always be checked by a search of the existing, project directories.

∘ It would be shortsighted to think that there will not be significant changes of programming and software support personnel over the lifetime of this mission. Therefore, until, or unless, efficiency considerations become a major concern in any particular application, and given that correctness and reliability are assured as far as is possible, the emphasis in coding should be on clarity and hence supportability. In low-level utilities the emphasis can take more account of efficiency, although in most cases this need not be at the expense of clarity.

∘ In-line comments/explanations will be the major programming documentation and should be used as such. Using them merely as brief aides-memoire for the programmer him/herself is not acceptable!

∘ We will create a CDS account at RAL where the latest software will be held so that all software developers can have access to it. The directory structure will be based on the definition of software categories which will be defined for purposes of the documentation.

It will be necessary to have parallel structures to cater for submitted code and for the version that is subsequently approved for release. Completed code will be placed in an appropriate /test/xxx/xxx/xxx/ directory. CDP will then be responsible for finding a suitable volunteer to review it in terms of the acceptance criteria. Only on completion of this process will the code be moved to the parallel /release/xxx/xxx/xxx/ directory.

Post-launch, when working on real data exposes problems, it will be important still to maintain this or a similar scheme. For it to be useful, however, the half-life of code in the /test/ directory must be kept as short as possible.

5. Acknowledgements.

Thanks to Bill Thompson for numerous comments on version 1 and for contributing section 3.4, and to Helen Mason for contributions to sections 3.8.3 - 3.8.6.