THE SPACE ENVIRONMENT DATA SYSTEM (SEDAT)

Richard Stamper[(1)] and Mike Hapgood[(1)]

[(1)]Rutherford Appleton Laboratory, Chilton, near Didcot, Oxfordshire, OX11 0QX, UK, Email: M.Hapgood@rl.ac.uk

ABSTRACT/RESUME

This paper presents the Space Environment Data system (SEDAT). This is an ESA-funded project to develop a networked tool for analysis of space environment data based around a central database. This paper outlines the overall structure of SEDAT and the objects through which it can be used. The main emphasis is on the user view of the system but some appropriate implementation details are included.

1. BACKGROUND

The aim of the SEDAT project has been to develop a new approach to the engineering analysis of the spacecraft charged-particle environment. It comprises three main elements:

- A database containing a large and comprehensive set of data about that environment as measured in-situ by a number of space plasma missions. Thus the user can select a set of space environment data appropriate to the engineering problem under study.
- Support for software tools that can operate on the data retrieved from the SEDAT database. A core set of tools has been developed as part of the project, e.g. provision of system tools to open, read, write and close datasets.
- And, most importantly, SEDAT provides a software infrastructure to integrate the various elements of the system and to provide user access to those elements.

2. OVERALL STRUCTURE

The overall structure of SEDAT is shown in Fig. 1 below. It is a client-server system. The user interacts with SEDAT via a graphical user interface that runs on his or her local computer. This is the client software and communicates over the network with a server that runs the main SEDAT system. This client-server approach is attractive because the local user interface gives a good fast response to user inputs while the remote server allows consolidation of the major data storage and processing on a powerful central computer.

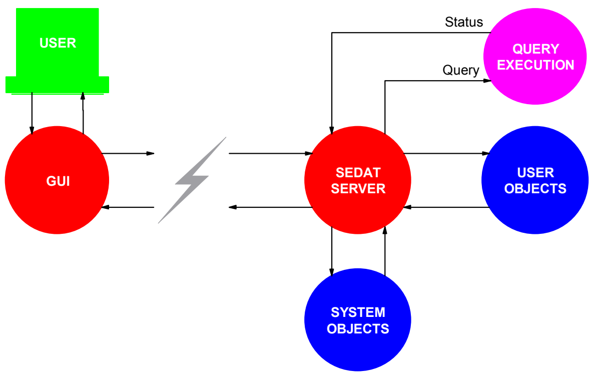

Fig 1 : Overview of the SEDAT system.

The server provides the user with access to three groups of facilities:

- Query server. This is the heart of SEDAT where tools are applied to datasets to perform the analyses specified by the user.
- System objects – these are the core datasets and tools that the user may use in his or her analyses.
- User objects – these include the tools and datasets created in the course of the user's work. They also include (a) queries, which are objects that allow tools to be associated with data and then executed on the query server, and (b) figures, graphical output from tools. Finally, there is also a class of objects called projects. These are containers for logical groupings of objects from the other classes, e.g. all objects associated with a particular study or project.

3. SEDAT OBJECTS

3.1 The GUI

The SEDAT graphical user interface (generally referred to as the GUI) is written in the Java programming language and thus can be run on most, if not all, computers. For example, within the development work it has been run on both Unix and Windows platforms. All that is required of the host platform is a copy of the Java language (specifically JDK 1.1.7 or later) and a network connection to the SEDAT server.

The function of the GUI is to provide access to all SEDAT objects available to the user. These include system objects, his or her own objects and objects owned by other users (subject to authorisation by those users). A separate window is displayed for each object. It shows all pertinent information about the object and provides facilities for the user to inspect and manipulate the object (see example in Fig. 2 below). Each window is initiated by context from other windows – typically by double-clicking on the object name.

3.2 Datasets

SEDAT provides extensive support for datasets. It includes a wide range of system datasets as shown in Table 1 below. It also allows users to create their own datasets. The datasets may be personal datasets loaded into SEDAT or be the output from a SEDAT tool.

This support is underpinned by the use within SEDAT of a RAL software tool called STPDF (Solar-Terrestrial Physics Data Facility). This allows us to store data in a wide variety of internal formats but to provide homogeneous access to them through a small set of applications programming interfaces. In principle, any sensible format (and even some bizarre formats) can be ingested into STPDF; it just requires the creation of appropriate descriptions within the rules used by STPDF. For SEDAT purposes the main data formats in use are ASCII flat files, binary tables and, most importantly, the NASA Common Data Format (CDF). CDF is a well-known standard in the space science community and much space environment data is available in this form.

The functionality of SEDAT includes support for key metadata, e.g. null value representations, labels for listing and plotting (parameter names and units), ratio for conversion to SI units, etc. It is now widely recognised that these "data about the data" greatly facilitate the operation of any tools that process data and thus metadata are an essential part of any dataset. To achieve this SEDAT exploits existing STPDF facilities to store and access metadata, e.g. by providing auxiliary tables to store metadata associated with ASCII and binary files, and for CDF by directly exploiting its power to store metadata in "global and variable attributes". To make best use of the latter facility, SEDAT encourages use of CDF files that conform to the IACG Guidelines on CDF usage [1].

Table 1. SEDAT system datasets.

Spacecraft	Instr.	Particles	Source
IMP		p+, 1. to 999. MeV	NSSDC
GOES	SEM	p+, 1. to 999. MeV	NOAA
GOES	SEM	e-, 2. to 999. MeV	NOAA
GOES	SEM	He++, 3.8 to 500. MeV	NOAA
GOES	SEM	X-ray	NOAA
Meteosat-3	SEM 2	e-, 0.043 to 0.3 MeV	A. Coates
ISEE 1	MEPI	p+, .024 to 2.081 MeV	D.J. Williams
ISEE 1	MEPI	e-, .022 to 1.2 MeV	D.J. Williams
ISEE 2	KED	e-, 0.018 to 1. MeV	E. Keppler
STRV-1b	REM	p+, 35. to 300. MeV	P. Buehler
STRV-1b	REM	e-, 2. to 10. MeV	
MIR	REM	p+, 35. to 300. MeV	P. Buehler
MIR	REM	e-, 2. to 10. MeV	
CRRES	MEA	e- .1 to 2. MeV	A. Vampola
AZUR	EI-88	p+, 1.5 to 104. MeV	D. Hovestadt
AZUR	EI-88	He2+, 6. to 19. MeV	
SAMPEX	PET	p+ /He2+, 18. to 85. MeV	R. Mewaldt
UARS	HEPS	p+, .1 to 160. MeV	D. Winingham
AMPTE-UKS	ELX	e-, .01 to 16. keV	M. Hapgood

SEDAT includes facilities to inspect datasets via the GUI. Fig. 2 below shows an example of the GUI window for a SEDAT dataset. This displays a text description of the dataset, the number of records in the dataset, a list of the fields in the dataset and a list of the users authorised to access the dataset.

The window also provides access to drop-down menus (by clicking on the Dataset, Users and Help fields in the traditional manner). These allow the user to inspect the data values (e.g. by generating an ASCII listing in a new window), change user privileges on personal datasets, etc.

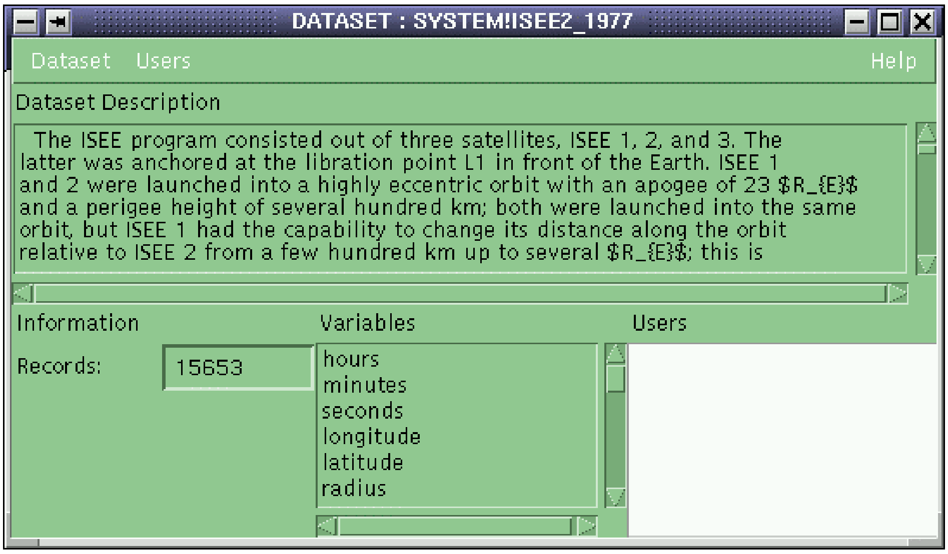

Fig. 2 : SEDAT dataset window medium energy electron data from the KED instrument on ISEE2

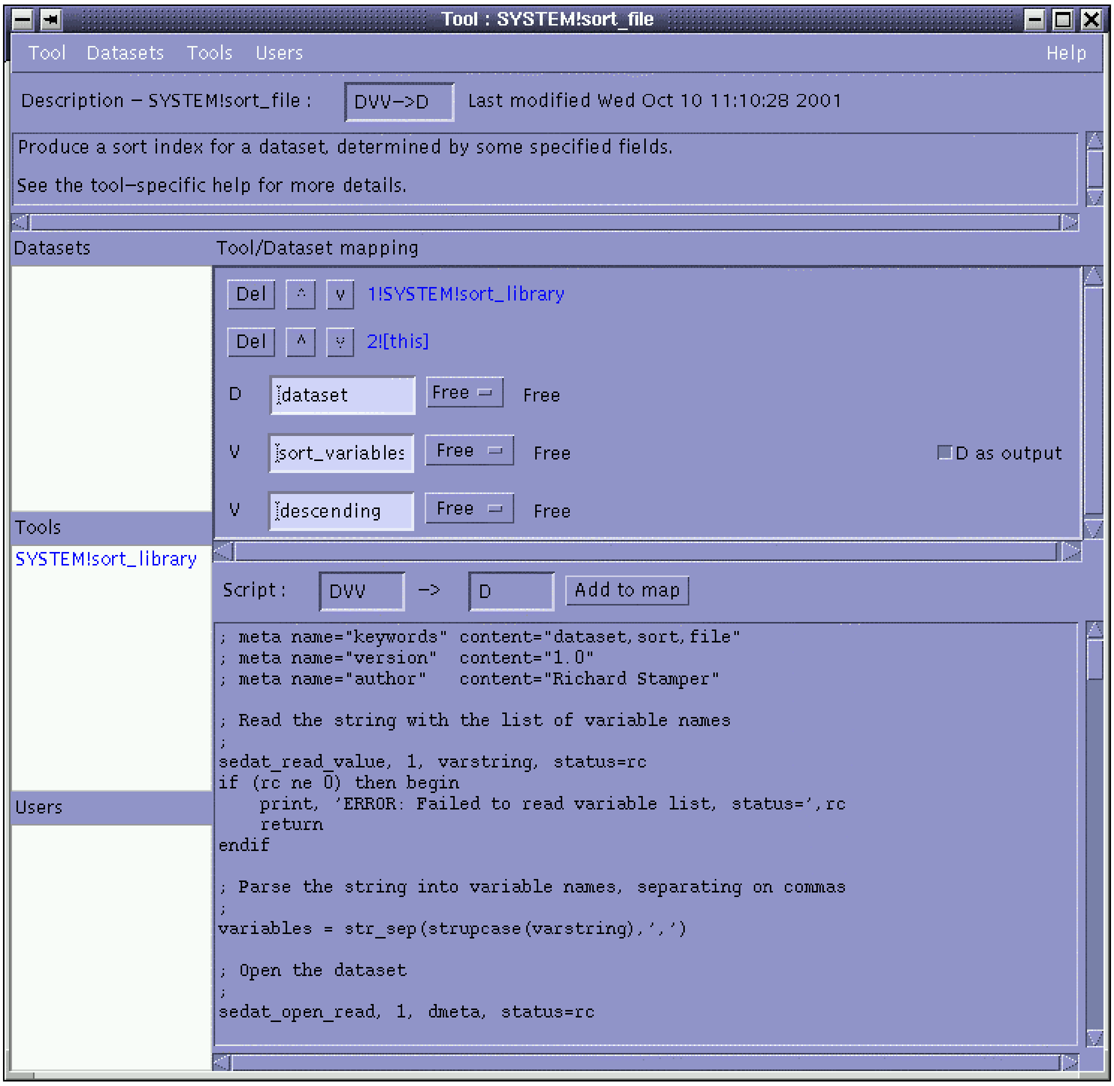

Fig. 3 : SEDAT tool window showing the system sort tool

3.3 Tools

SEDAT tools are primarily written in the IDL programming language. This commercial product was chosen because of its great power for data analysis, e.g. to carry out mathematical manipulations and generate graphics. It is also attractive because of its wide usage in the space science community.

Each SEDAT tool comprises an IDL script together with a “mapping” that specifies options through which the tool is linked to SEDAT datasets and to any external parameters to be specified at run time. The mapping also specifies the names and compilation order of any subsidiary tools that must be linked with the current tool. These subsidiary tools are typically IDL functions and procedures. Subsidiary tools may not be linked to datasets or external parameters.

System tools may include procedures that call programs external to IDL such as C and Fortran subroutine libraries. Examples of these include the data access via STPDF as well as access to RAL’s Clustran co-ordinate transformation library and the well-known Shielddose program for calculating dose-depth curves. External programs cannot be called directly from user tools because of the security problems raised by that approach. Instead user tools must do this directly via the appropriate system tool.

As shown in Fig.3, the GUI window for tools provides a series of separate panels that allows the user to crreate (a) a text description of the tool, (b) the IDL script at the heart of the tool, (c) the mapping described above, (d) the list of datasets to be associated with the tool and (e) to create a list of the users authorised to access the tool.

3.4 Queries

SEDAT tools are run by creating and executing user objects called queries. Each query is linked to a specific top-level tool, i.e. not to a subsidiary tool. When the query is executed it compiles and runs this tool together with subsidiary tools specified in the tool mapping. The query window (see Fig.4) has a mapping in which tool options to link to datasets and external parameters are converted into links to specific datasets and parameter values

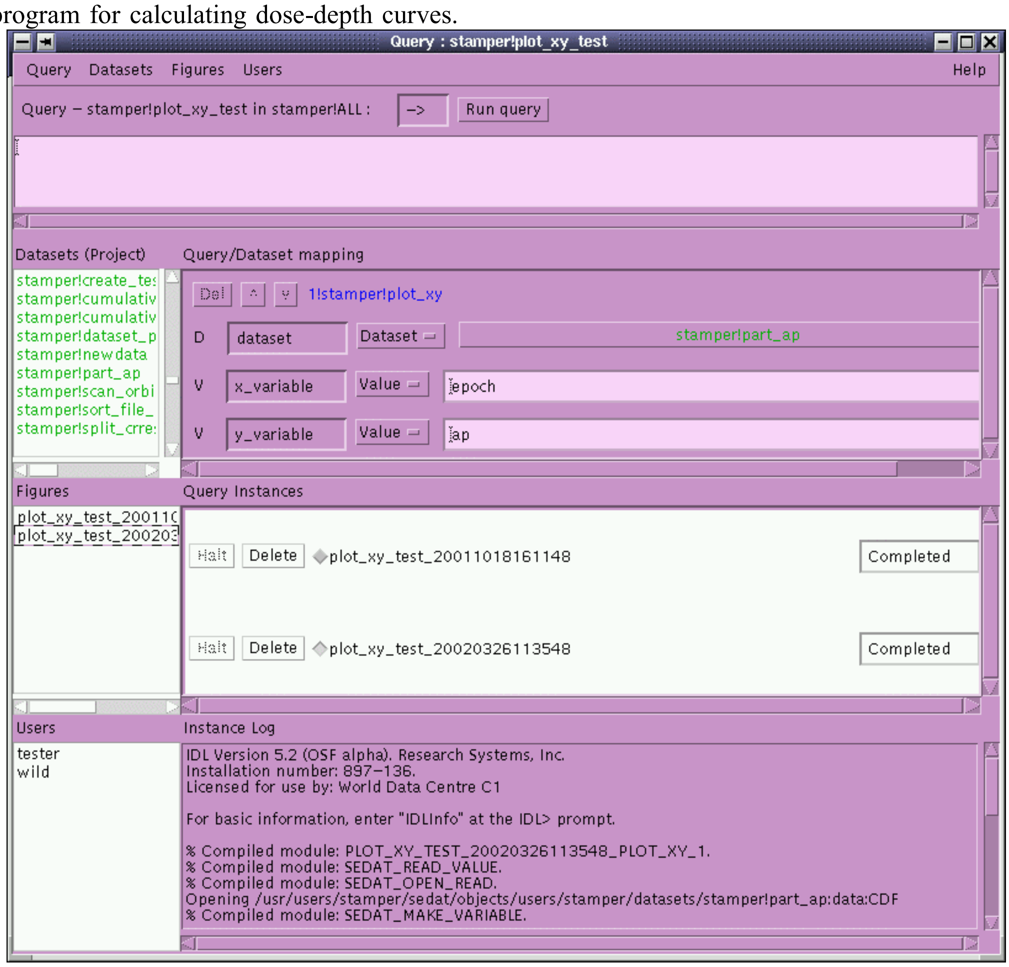

Fig. 4 : SEDAT query window showing a query that runs the stamper!plot_xy XY-plot generating tool

The GUI window for queries allows the user to configure each query (i.e. set the mapping via the mapping panel) and then can call a drop-down menu to send the query for execution. A query instance panel shows the status of the currently executing instance of query and also of any previous instances. On selecting a specific instance its log file is displayed in another panel. There is also a facility to delete query instances and their log files. A further panel displays any figures created by the query. Finally the query window also provides facilities to (a) create a text description of the query, (b) manipulate the datasets to be associated with the query and (e) to create a list of the users authorised to access the query.

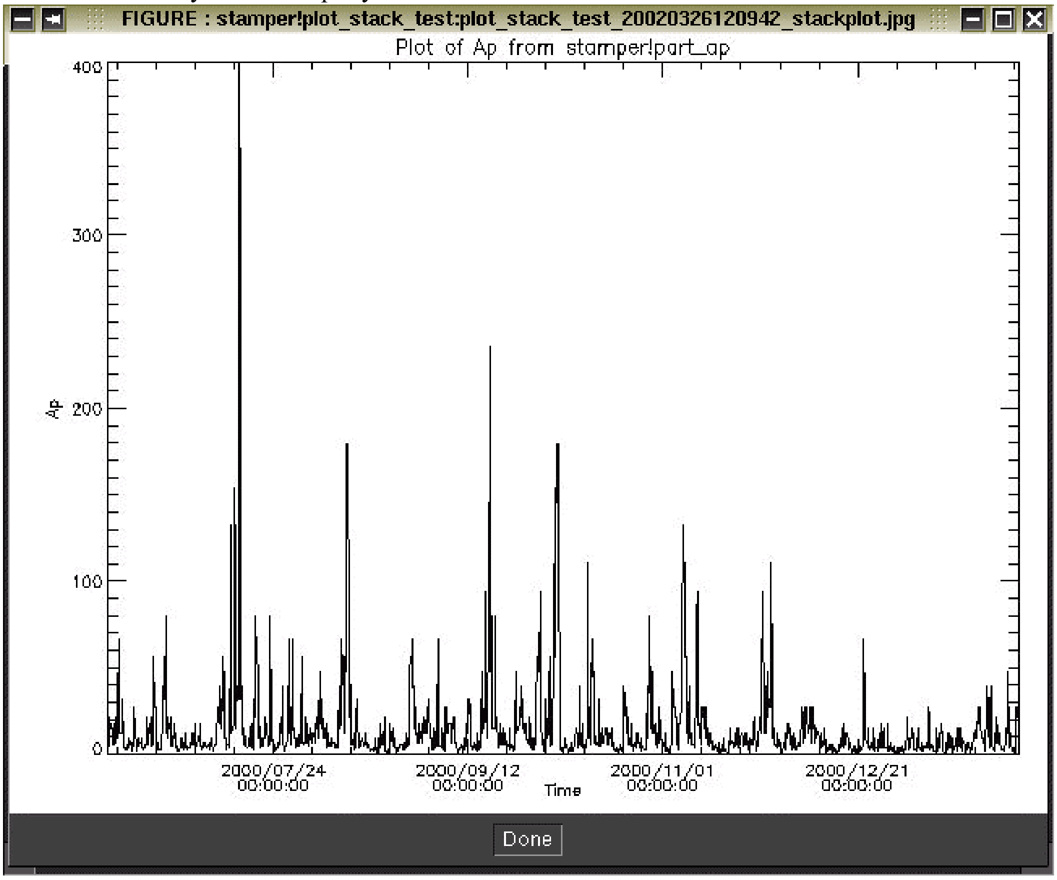

Fig. 5 : SEDAT Figure window showing a plot of Ap against time

3.5 Figures

SEDAT tools can use the full range of graphics supported by IDL, e.g. line plots, histograms, contour and surface plots, false colour plots, etc. However, the graphics output can currently only be written to a JPEG file and not to other IDL graphics devices. This restriction is imposed by the security requirements that follow from the SEDAT architecture, i.e. that the user does not have direct access to operating system on the SEDAT server computer. The JPEG graphics file is created and closed by system tools provided for the purpose. These also allow the user to specify the colour tables used to create the file.

The GUI window for figures displays the JPEG file (see Fig. 5). It can be invoked by double-clicking on the figure name, e.g. in the figures panel (left-hand side of Fig. 4) of the query that created the figure.

3.6 Projects

As already noted projects are logical collections of other SEDAT objects. Thus a user can create a project for each specific task to which SEDAT is applied. This is just a way of organising the display of SEDAT objects for the convenience of the user. After successful login to SEDAT, the user's main window displays a list of all projects accessible to a user.

Note that projects are not exclusive in any way. Any object can be placed in multiple projects if this is appropriate – as may well be the case for some utilities. Similarly any object accessible to a user can be placed in a project; the content of a project is not restricted to the user's own objects. For example, it is often important to include relevant system datasets in a project.

4. SUMMARY

The SEDAT system is a new tool for engineering analysis of space environment data. This paper summarises the user view of SEDAT and outlines the objects through which the user can exploit this system. For further information on SEDAT see the web site at http://www.wdc.rl.ac.uk/SEDAT.

5. ACKNOWLEDGEMENTS

We would like to thank all the scientists and data centres who have supplied for inclusion in SEDAT. In many cases, these people have also provided much advice on metadata and on the use of the data. Particular thanks go to Daniel Heynderickx and his team at BIRA for their help and encouragement. Finally we thank ESA for their financial support through contract 12854/98/NL/NB.

6. REFERENCES

[1] http://spdf.gsfc.nasa.gov/istp_guide/istp_guide.html