≈ Sept. '92

PRODUCT GENERATION SYSTEM TOOLKIT STUDY REPORT

Ted Meyer

Goddard Space Flight Center

1 INTRODUCTION

This is the first status report on a study that will define the functionality and usage of the Earth Observing System (EOS) Data and Information System (EOSDIS) Product Generation System (PGS) Toolkit. The study of the PGS Toolkit is part of an overall EOS Investigator-to-EOSDIS Interface Study to investigate and define the relationships between the EOS investigators, their software, software development efforts and systems, and the EOSDIS. The goal of the PGS Toolkit study is to define the science product software interface to the Toolkit as completely as possible. Early definition allows EOS science programmers and managers to make design decisions early in the software development life cycle. This paper is the first version of an evolving and expanding report that is based on the needs of the EOS science software developers.

The audience for this study is members of the EOS instrument and interdisciplinary investigator teams associated with the development of software and the staffs of the Distributed Active Archive Centers (DAACs) that will host PGSs. In its final form this study will be provided to the EOSDIS Core System (ECS) contractor to provide a basis for direction in the realization of the Toolkit in the PGSs and the Science Computing Facilities (SCFs).

1.1 EOS Investigator-to-EOSDIS Interface Study

The EOS Investigator-to-EOSDIS Interface Study is intended to define the interfaces required by science investigators to successfully integrate science software into EOSDIS. The goal of the study is to define the

- EOSDIS services required by investigators;
- Tools to emulate EOSDIS in investigator SCF environments;
- Common libraries, languages, utilities, and Commercial-Off-the-Shelf (COTS) Software required;
- Science investigator approaches to developing portable, maintainable software;
- Approaches to data management; and

- Direction of SCF development.

The study approach is to meet with the EOS instrument teams and to establish a common understanding of EOSDIS and science software development and management. The intent is to promote communications and a dialog with the instrument teams and DAAC staffs via continued meetings and discussions, the Data Processing Working Group (DPWG), the *EOSDIS Science Data Processor Newsletter*, and the *EOS Science Software Developer's Handbook*. Study status reports, such as this one, based on feedback from the EOS community and project research will be presented during follow-up meetings with instrument teams, DPWG meetings, and through the Handbook and newsletter. The goal of the study is to identify and resolve conflicts and misinterpretations of EOSDIS and to develop a consensus view of the interfaces between the EOS Investigator and EOSDIS.

1.2 EOSDIS Architecture

The architecture of EOSDIS and the ECS and the relationship of the PGS Toolkit with other components of EOSDIS, as defined primarily by the ECS Request for Proposals (RFP), are described in this section. Because of its size and complexity, ECS has been defined in the RFP as a hierarchy of segments, elements, subsystems, and components. Three ECS segments are defined to support three major operational areas: flight operations, communications and system management, and science data processing. The segments are further divided into ECS functional elements to provide the support required by the operational segments.

Figure 1 depicts the EOS Ground System. The ECS segments and elements are shown as shaded boxes in the figure. It also shows external supporting systems, physical interconnections, and interfaces. There are other EOSDIS systems that are themselves outside the scope of ECS and will be separately provided by the Government. These are shown as unshaded boxes in the figure. These include Space Network (SN), EOS Data Operations System (EDOS), Network Control Center (NCC), Space Network Control Center (SNCC), Flight Dynamics Facility (FDF), International Partner (IP) facilities, Platform Test and Training System (PTTS), National Oceanic and Atmospheric Administration (NOAA) data centers, Affiliated Data Centers

(ADC), and Other Data Centers (ODC). The NASA Communications network (NASCOM), EOS Communications network (ECOM), Tracking and Data Relay Satellite (TDRS), Deep Space Network (DSN), Ground Network (GN), and Wallops Tracking Station (WTS) are also shown. Shaded areas in the IST, SCF, and FST boxes in Figure 1 represent ECS software toolkits that will reside on compatible external hardware.

The first of the three ECS segments is the Flight Operations Segment (FOS). The FOS manages and controls the EOS spacecraft and instruments. The FOS elements include the EOS Operations Center (EOC), Instrument Control Centers (ICCs), and Instrument Support Terminals (ISTs). Several ICCs constitute an Instrument Control Facility (ICF). ICCs will be functionally similar and will interface with the EOC but will be used to manage different instruments. The EOC provides mission planning and scheduling and the control and monitoring of mission operations of the EOS spacecraft and instruments. ICCs schedule, command, and operate the science instruments and monitor instrument performance. ISTs are defined to be software to connect science team representatives to an ICC in support of remote instrument control and monitoring.

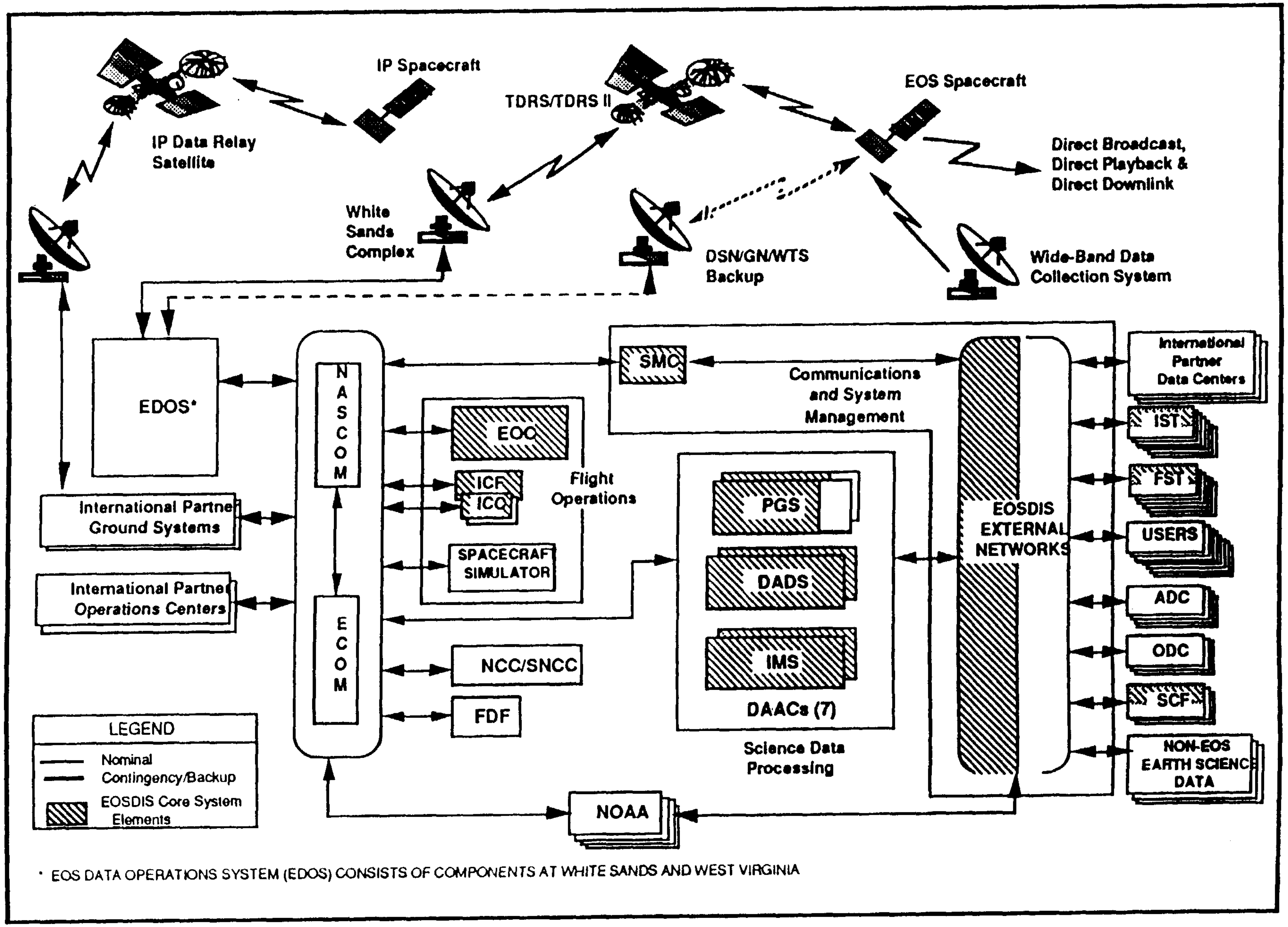

Figure 1. EOS GROUND SYSTEM ARCHITECTURE

The Communications and System Management Segment (CSMS) provides overall ECS management and operations of the ground system resources, provides facilities and communications and networking services for the science data communications network, and manages the interfaces to NASA networks and other communications networks. The CSMS elements include the System Management Center (SMC) and the EOSDIS Science Network (ESN). The SMC provides system management services for EOSDIS ground system resources. ESN provides an internal network for ECS communications, a network interface to the science user network, network services at the application layer and a network management and help facility.

The Science Data Processing Segment (SDPS) provides a set of processing and distribution elements for science data and a data information system for the entire EOSDIS. The SDPS elements include Distributed Active Archive Centers (DAACs) consisting of ECS and institutional facilities including Product Generation System (PGS), Data Archive and Distribution System (DADS), and Information Management System (IMS). PGS and DADS facilities process data from the EOS instruments to standard Level 1 through 4 data products; provide short- and long-term storage for EOS, non-EOS earth science, and other related data, software, and results; and distribute the data to EOSDIS users. IMS provides a distributed data and information management service for the ECS including a catalog system in support of user data selection and ordering. IMS also provides Field Support Terminals (FSTs). These are portable terminals that can be employed at field investigation sites to provide access to one or more DAACs.

SDPS toolkits include those for IMS, DADS, and PGS. The configuration of toolkits for a particular installation is tailored to the user's site-specific requirements and responsibilities for the EOS mission. The PGS toolkit is the subject of this study report and is discussed in the following sections. Figure 2 illustrates the relationships between the PGS, IMS, and DADS Toolkits on the PGSs and the SCFs.

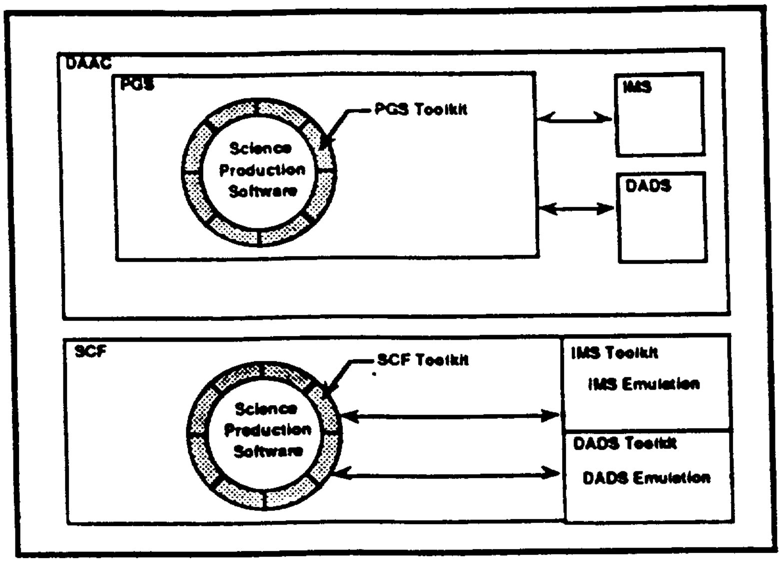

Figure 2. IMS and DADS Toolkit Interfaces

The IMS toolkit will provide data visualization tools, local SCF information management capabilities, and the capability to execute IMS services in a client-server mode. The IMS toolkit will be installed on both the IMS server computers and on the SCFs. The IMS toolkit will provide the capability to browse subsetted, subsampled and summary data products using the IMS server computer resources and local workstation resources. When the IMS toolkit is installed on local workstations, the workstations will provide a local staging area for data products and will host the data visualization tools. The IMS toolkit will also provide data base management capabilities at a local SCF to support the ingest and management of information. In providing these capabilities to the local sites, standards will be imposed that will facilitate the migration of special products into ECS.

The DADS toolkit will provide tools for data format conversion of EOS data, data product subsetting and subsampling, lossy and lossless data compression, and data transformation.

1.3 The PGS Toolkit

The SCF and the PGS are the components of the EOSDIS on which the development, maintenance, and execution of software for the production of EOS science data products will occur. Although the SCF and PGS will have other functions and interfaces, for purposes of the PGS Toolkit study, only the interfaces available to the science data production software will be considered.

The ECS contractor will develop the PGS and SCF implementations of the Toolkit. A SCF will reside with each EOS investigator and be the host computer for science software development, science data validation, and processing of special data products. The SCFs will be procured by the science investigators based on the investigators' requirements and preferences.

1.3.1 Purpose of the Toolkit

The EOS Science Software Toolkit will define the software environment in which the science data production algorithms will be developed and reside, both at the investigators' SCFs and the EOSDIS PGSs. Science software developers will be expected to use only the Toolkit as a standard library in designing and writing software. From the perspective of the science software implementer, the Toolkit will replace parts of other standard libraries, such as the ANSI C library or the IMSL math tools, for most uses. The Toolkit will serve to isolate the science software from hardware, system, and language interface library dependencies.

The Toolkit will be a library in the programmer's sense of the word, and will be implemented to support science software programmers and programs on the SCFs and PGSs.

The Toolkit will be production oriented. It will be designed for the purpose of developing software to generate standard science products. Therefore, it will not have an extensive set of routines for presentation graphics since, in a production system, it is unlikely that such graphics would be viewed by scientists or users.

The Toolkit will

- Be a library of callable functions and subroutines ;
- Be production oriented;
- Serve to isolate the science software from system dependencies;
- Include routines for input/output;
- Include routines for math and statistics support;

- Include interfaces to various science and geographic standards ;

The Toolkit will **not**
- Include stand-alone applications or utilities;
- Include stand-alone programs for
  - data analysis,
  - graphic presentation,
  - software management,
  - data base management,
  - communications,
  - Computer Aided Software Engineering (CASE), or
  - configuration management.

1.3.1.1 Portability and Maintainability

As mentioned above, the Toolkit will reside on the PGS at each DAAC and on each investigator's SCF. It is entirely possible that the DAACs will not have common hardware configurations for the PGS host computers. It is almost certain that the EOS investigator's will have a variety of SCF computers depending on local preferences, current systems, and cost and sizing factors.

Although UNIX is an acknowledged standard for EOS, it is well known that major portability issues often arise between UNIX platforms. Even by further defining standards such as the Portable Operating System Interface for Computer Environments (POSIX) or the X/Open Portability Guide, Release 3 (XPG3) for EOS associated systems, we have only assured ourselves that each system has a basic set of shared system functions. Each vendor of UNIX platforms offers a variety of useful features that extends its function library. By limiting science development to only those shared functions, forgoing use of vendor-specific extensions, developers can improve portability. One goal of the Toolkit is to provide such a shared library in order to make integration into the PGS feasible and yet give science software programmers access to the functionality which is available on most science workstations.

To maximize the portability of algorithms from the SCFs to a PGS, science software developers will only use the defined interfaces to the Toolkit library to implement their software. All program input and output will be routed through these tools to ensure correct execution and implementation in the operational environment.

The life span of the programs generated by EOS scientists may be decades. In that time frame, and even before EOSDIS is actually implemented, we can be certain that data processing technology will make major and sometimes unexpected advances. It is hoped, that whatever the advancements, the interfaces to the Toolkit will remain a constant in order to continue to use decades-old code with minimal software maintenance. The Toolkit itself will be designed to be portable and will be upgraded to make use of advancements in technology such as high speed mass storage systems, high performance processing hardware, and improved communications technology. By using the Toolkit science developers will create software that may be able to benefit from these advances.

Another goal of the Toolkit is to separate the maintenance of the science algorithm and code from the maintenance of the support software. Maintenance of the science code is likely to be a difficult problem over the EOS time frame. The higher the percentage of the code that can be placed into the institutionally maintained Toolkit facilities, the less imposition and reliance on the science maintenance resources.

1.3.1.2 Product Standardization

The Toolkit will provide investigator teams access to common procedures, data models, reference frame models, physical parameters, platform position information, and data structures. Usage of these parameters will aid in the generation of standard products which will improve interoperability, simplify the sharing of data sets, and allow more accurate and concise interdisciplinary correlation of data.

1.3.1.3 Science Resources

With hundreds of investigators developing software intended to be used in production of standard EOS science products, it is almost certain that many will be developing or obtaining software with similar or same functionality.

The Toolkit can help to minimize the resources that the investigators will invest in development and maintenance of software that is likely to be redundant.

1.4 Roadmap for this Paper

This paper consists of three major sections. In Section One we have introduced and described the PGS Toolkit and its context, its surrounding architecture, and its purpose. Section Two reports on the current status of the PGS Toolkit study and the need for science involvement and describes further plans and direction. Finally, Section Three reports the result of the study to date and describes the tools.

In later versions of this paper, Appendix A will be used to present preliminary proposals for calling syntax, and input and output argument requirements, and Appendix B will be used to present science software implementation scenarios as examples of how tools or tool sets might be used.

2 THE PGS TOOLKIT STUDY REPORT

The purpose of this paper is to report on the status of the study of the PGS Toolkit. It is intended to promote discussion with and review from the EOS science community. This paper is intended to inform the investigator community of the current working perceptions of the functionality of the Toolkit resulting from the study and to promote feedback pertaining to the needs of the investigators.

It is important that the Toolkit is designed to meet the data processing needs of those designing the science software. As much as possible, software likely to be used by multiple investigators should be made part of the Toolkit, rather than redundantly developed.

The study's purpose is to define the investigators' software interface to the Toolkit, as much and as soon as possible. This study is based on input from the EOS science community through discussions, interviews, working group meetings, and correspondence. It will include information garnered from the development of prototype tools by science groups and the EOS Project. Other similar projects, such as the Upper Atmosphere Research System (UARS) Project, are also sources for input to this study. A final version of this study report, based on this iterative feedback, will be delivered to the ECS contractor once the contractor is selected and begins work.

The ECS contractor will provide an interface tools specification, based, in part, on this study. At that time, full science software development can begin using the provided specification as the basis for interfacing with the operational environment.

During the period from now until the ECS contractor provides the specification, science software developers should, as much as possible, design and prototype science software in a modular fashion based on the understanding of the Toolkit interfaces that this study report provides. It is important to realize, though, that this first version of the report is tentative and subject to change based on science community feedback.

2.1 PGS Toolkit Study Activities

The study is being undertaken through discussions, interviews, working group meetings, and correspondence with the EOS investigators, DAACs, and other science software experts and developers.

EOS Ground System and Operations Project (GSOP) personnel have met with the instrument teams listed in Table 1 and have presented an initial overview of the Toolkit. Information culled from the discussions and concerns presented at these meetings as well as the EOS Mission Operations Working Group (EMOWG) and EOSDIS Version 0 System Engineering Meetings over the last year are considered part of the Toolkit study and have been incorporated in this status report.

Another source of information used in this report was the ECS RFP. The RFP delineates, at a high level, aspects of a PGS Toolkit the ECS contractor will provide.

GSOP has also received presentations from the UARS Software Development Team members on lessons learned in their developmental effort. The programmer's guide for the UARS Central Data Handling Facility (CDHF) and the UARS CDHF User's Guide were other sources for helping to define parts of the preliminary outline of the tool groups listed in Section 3.

Instrument Team	Date of Meeting
MODIS	June 6, 1991
ASTER	June 10 and 13, 1991
AIRS	June 11, 1991
STIKSCAT	June 12, 1991
ACRIM	June 12, 1991
HIRIS/JPL	June 13, 1991
MISR	June 13, 1991
HIRDLS	August 26, 1991
MOPITT	August 26, 1991
HIRIS/Team Leader	August 27, 1991
CERES	January 22, 1992

Table 1. PGS Toolkit Presentations

2.2 PGS Toolkit Study Plan and Schedule

The GSOP began presenting an initial outline of the PGS Toolkit to investigator teams in June 1991. The presentation of the outline and the subsequent feedback provided the source to produce and distribute this first draft of the PGS Toolkit Study Report in April 1992.

The ECS contract is expected to be awarded in late 1992, when a final version of this report will be delivered to the ECS contractor. Between now and then, GSOP will continue to solicit feedback on the Toolkit and continue to refine and expand this report. Prototype or heritage tools from the EOS science community and other projects will be examined and used as possible models for PGS tools.

Current GSOP plans call for the ECS Contractor to provide an initial PGS Toolkit Interface Specification for review in the second quarter of 1993. This will be distributed and again feedback from the investigator community will be requested. A final version of this specification is planned for delivery in the last quarter of 1993. EOS science software developers will use this specification as the basis for designing production code.

GSOP plans for a beta-test version of the PGS Toolkit library to be made available to the SCFs in the first quarter of 1995. On delivery of this library, science software developers can be implementing their production code on their SCFs in a PGS emulation mode. A beta-test DAAC is expected to be ready in the third quarter of 1995. At that time, science software developers will have the opportunity to test software portability in a preliminary PGS environment. EOS AM flight instrument investigators will be encouraged to begin testing early versions of their software in the beta-test DAAC.

2.3 Science Community Feedback

It is hoped that the Toolkit study and this status report will motivate feedback from the investigators. There is strong interest in receiving specific proposals and requirements for specific tools, definitions of science data processing conventions and standards, calling arguments, numeric methodology, and operations concepts from the EOS science data processing community. It is desired that interested scientists will take a proactive role in the Toolkit design. Section 3 and Appendix A will be expanded to incorporate specific information pertaining to tools as it becomes definitive during this study.

Figure 3 portrays the importance of investigator participation in the Toolkit Study. Limited science software developer participation will result in an incomplete Toolkit specification and likely portability problems. By taking an active role in developing this interface specification, developers can help to minimize the problems in porting the operational code from the SCF to the PGS and from a PGS machine to a replacement PGS machine.

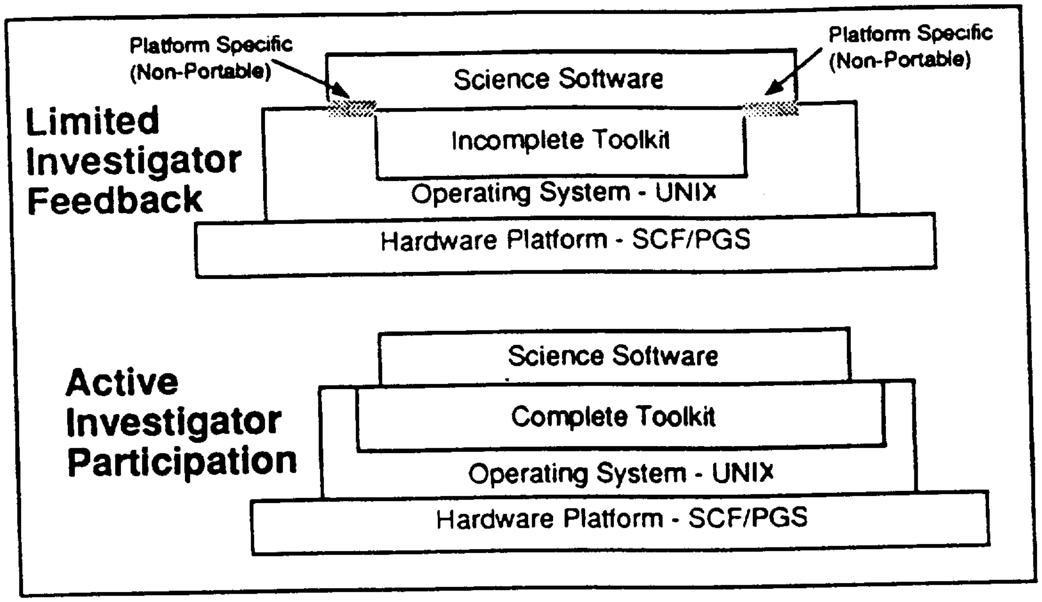

Figure 3. The Effect of Investigator Participation

3.0 THE TOOLS

The PGS Toolkit Study has provided a current perception of what the PGS Toolkit may be and what is required by the EOS science software developers. This section describes the tools as defined by the study. The current concept of the purposes and methodologies of using these tools are presented to act as a starting point for discussion and further defining of EOS investigator requirements.

3.1 Using the Tools

In future versions of this paper this section will define the methods for incorporating library calls into science software. The tools will be designed to allow bindings to FORTRAN, C, and other programming languages required by the science community. General syntax description and procedures for linking to and accessing the Toolkit library with these languages will be presented here when defined. Appendix A will provide specific linking and syntax information for each tool when it is developed.

3.2 Language Considerations

The ANSI C library provides access to system capabilities. The C Library, like the Toolkit, provides a common interface to varying hardware and system platforms. In some cases the science software programmer will not directly access standard ANSI C Library functions. Instead, the Toolkit will act as an interface to the C functions. For example, C file creation and open functions, such as fopen, will be mapped to PGS Toolkit functions with options to implement high speed file access, or temporary or production stream file access. A related situation exists when using FORTRAN, although FORTRAN's I/O functions are intrinsic in the language and are not part of an external library. In using FORTRAN, science software programmers will avoid certain functions, like the OPEN statement, and will instead call Toolkit functions. The Toolkit "open" functions will provide information to the PGS Scheduler pertaining to the type of file, the ID of the program or production stream, and the criteria for file closing. Standard ANSI C fopen and FORTRAN OPEN statements will not be used by programmers.

This will be true for much of the C library such as memory allocation functions, I/O functions, and bit manipulation functions. The C library functions or FORTRAN intrinsic functions that are not replaced by Toolkit calls are those that are system independent and, therefore, preserve the Toolkit's purpose to isolate the science software from system dependencies. Figure 4 illustrates the relationship between the Toolkit and the ANSI C Library and FORTRAN.

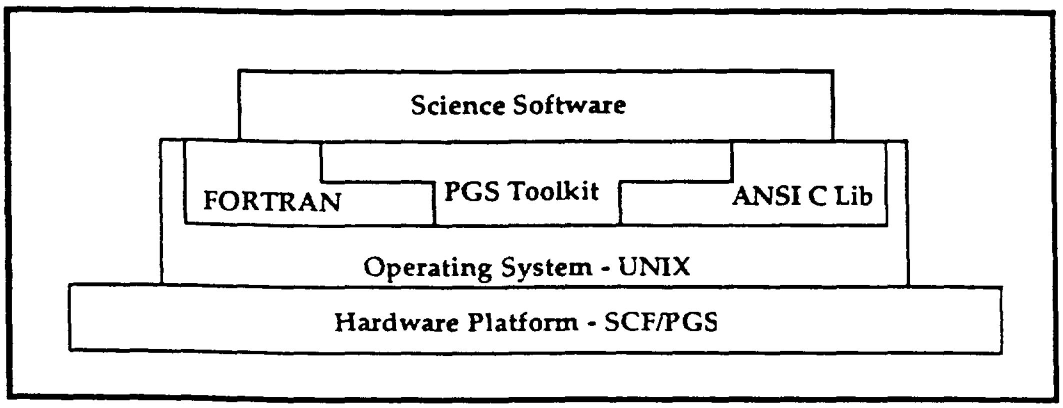

Figure 4. PGS Toolkit, ANSI C Library, and FORTRAN

3.3 Tool Descriptions

This section divides the tools into collections of related functionality and provides high level descriptions of these tool groups. The descriptions include general discussions of the currently perceived purpose, capabilities, limitations, and issues related to the tool groups and the EOSDIS environment.

If prototypes of tools are developed by science groups or the Project these will be defined and information regarding the prototypes will be documented and included in this report.

In later versions of this paper, as information pertaining to specific tools or sets of tools becomes definitive, Appendix A will be used to present tool lists, tool names, preliminary proposals for calling syntax, and input/output argument requirements.

The tool descriptions in the following sections use the terms "program" and "production stream." The term "program" refers to a single process or execution which might produce a single product or complete a processing step on a data set. "Production stream" refers to a logically related collection or flow of programs and program control that might produce a range of outputs from a single data product to multiple level 1 to 4 data products.

3.3.1 Production Control and Scheduling Tools

The problem of controlling the production and scheduling of a distributed system with seven nodes, hundreds of interdependent product generation processes, human interaction for product validation, and strict production deadline requirements is obviously not a simple one. Since none of the science data production can occur without the availability of data, the PGS will require a data driven or event driven scheduler, rather than a time driven system. Events might include data set availability (processing starts when the input data sets are staged), data set validation (processing starts when the input data sets are validated by investigators), and data set requests (processing starts when the data products from a process are requested by users). The PGS Scheduler may provide a prioritization to a production stream which will determine the sequence in which runs are executed and what resources may be allotted to a production stream.

The PGS Scheduler will have to handle partial execution and be able to restart production streams. Science programs and production streams will need to be designed with breakpoints or restart points and flags so that if interruptions occur, for example due to non-availability of data, previous processing will not have been wasted.

Significant reprocessing of data is being planned in sizing the PGS. As science understanding changes or bugs are detected, improvements and fixes to production software will be made. The Scheduler will have to manage the reprocessing decisions along with on-going processing requirements.

A production stream may require data from remote DAACs in order to start or continue processing. Two options, not necessarily exclusive, may be available for acquiring these data: staging these data in preparation for the processing, or accessing the data across network links during processing. In the first case the PGS Scheduler will have to come into play to query for the availability of data before starting production streams. For the second case the science software may query for data availability and then inform the Scheduler of the need to request the data and complete processing, or suspend and restart processing when data become available.

3.3.1.1 Data Availability Tools

Although the specifics of the scheduling system are unknown, it is obvious that certain tools will be required. Tools to query for availability and validity of data will be required by science production runs. Toolkit interfaces to the Scheduler at the start of a production run may provide metadata about the dimensions of input data sets which would be used to refine processing methods.

Higher level data products may require partial processing of data before science algorithms can define what additional data may be needed to complete products. In this case mid-production run data availability queries might be needed to determine whether processing should be continued or put on hold.

3.3.1.2 Initialization Tools

Certain information will be obtained from the PGS Scheduler on starting a production stream. In order to be able to accurately track and identify data sets and processing, Run IDs, Stream IDs, and Product IDs will be provided by the Scheduler to insert into a standard product's metadata. These IDs will provide information for tracking the production process in order to support possible reprocessing, and to provide scientists concise information about the environment and procedures used to create a data set. Initialization tools will also provide the Scheduler information about the production stream. For example, a quicklook product stream will receive a higher priority from the Scheduler.

Tools for science software to query the availability of memory and other system resources and to request them might allow science software developers to provide for more efficient data and processing management within their software. Tool sets might be provided to initialize resources for both programs and production streams.

3.3.1.3 Termination Tools

Tools will be required to inform the Scheduler of the completion and validation of data products and associated meta- and browse data. This will allow the Scheduler to start other production runs dependent on those data products and pass metadata and browse products to the appropriate systems.

On termination of a program or production stream either the production stream must relinquish control of requested system resources or the Scheduler will automatically recover resources used by the stream. System resources requested by one program may need to be shared through a production stream. In this case, tools need to be designed to differentiate between resource requests for single executions and requests allowing the passing of resources between programs within a production stream.

3.3.2 I/O Tools

There are three broad categories of I/O Tools described here: temporary file I/O, production stream I/O, and product and auxiliary data access. Temporary file I/O refers to using files that exist for the duration of a single program (i.e., process or execution). Production stream I/O refers to using files that exist for the duration of a complete flow of programs and program control. Product and Auxiliary data access refers to interfacing with permanent files, data, and processes such as science data products and archives, metadata and catalog information, and orbit, attitude, and digital terrain elevation data.

Tools to access temporary and production stream files will allow various methods of access. Direct (or random access) and sequential (or serial) are two common types of access that will supported by the toolkit. Another option, high speed, has been suggested as a method of creating and accessing temporary data files. High speed file access might be implemented in a variety of ways including RAM cache buffers for mass storage media, RAM based mass storage emulation similar to the FORTRAN internal file type, or other new technology. High speed access would allow scientists to vastly improve performance of algorithms that might normally have I/O bottlenecks or be otherwise I/O intensive.

3.3.2.1 Temporary File I/O Tools

Temporary file I/O refers to files that exist for the duration of a single process or program execution. These files can be considered to be equivalent to FORTRAN's SCRATCH file type. The purposes of these files include providing temporary storage for high volume data structures and staging data in preparation of creating the final data product.

The content and organization of a temporary file is left entirely to the science software designer. However, tools to create, read, write, and otherwise manipulate standard data structures will be available and compatible with data structure tools for production stream files and standard data products. Access to a temporary file will be direct or sequential access and can be high speed or not depending on the priority of the production stream and the requirements of the science software.

3.3.2.1.1 Temporary File Housekeeping Tools

These tools are similar to standard file interface functions for creating, opening, closing, and defining the parameters of files. For temporary files though, the tools will only allow the files to exist during the execution of the program. Access methods for a temporary file (direct or sequential, high speed or normal) will be determined by the parameters passed to these tools.

In FORTRAN, examples of similar functions include INQUIRE, OPEN, REWIND, BACKSPACE, CLOSE, and ENDFILE. In the standard ANSI C library some of the related functions include feof, fclose, fflush, ferror, fopen, and fseek.

3.3.2.1.2 Temporary File Access Tools

Temporary file access tools will include function to write, read, or append data to a temporary file. Examples of similar FORTRAN functions include READ, WRITE, and PRINT. In C similar functions include fgets, fgetc, fprintf, fread, and fputs.

In addition to these, common function tools to read and write standard science and administrative data structures will be available for temporary file access. Examples of these tools might include write_image, read_image, write_registered_image, read_browse_header. Note that a separate study to determine the data structure requirements of EOS investigators is underway.

3.3.2.2 Production Stream File I/O Tools

Production stream I/O refers to files that exist for the duration of a complete flow of programs and program control. Production stream files may exist until some criteria such as processing deadline or production stream completion is met. File deletion criteria may be designated through tools by the programmer and may be met by the setting of flags by various production processes, completion of specific data products, or termination of specific production streams or programs.

Production stream files may be made accessible by foreign production streams allowing a method for multiple programs and investigator teams to share access to current or related science data.

3.3.2.2.1 Production Stream File Housekeeping Tools

These tools are similar to standard file interface functions for creating, opening, closing, and defining the parameters of files. For production stream files though, the tools will only allow the files to exist as long as the programmer defined criteria are met. Tools that would allow the definition of criteria for file deletions, foreign program access, and passing of file ownership from one stream to another would be included in this group.

Access methods for a production stream file (direct or sequential, high speed, or normal) will be determined by the parameters passed to these tools.

In FORTRAN, examples of similar functions include INQUIRE, OPEN, REWIND, BACKSPACE, CLOSE, and ENDFILE. In the standard ANSI C library some of the related functions include feof, fclose, fflush, ferror, fopen, and fseek.

3.3.2.2.2 Production Stream File Access Tools

As with temporary file tools, production stream file access tools will include functions to write, read, or append data to a production stream file. Examples of similar FORTRAN functions include READ, WRITE, and PRINT. In C similar functions include fgets, fgetc, fprintf, fread, and fputs.

Common function tools to read and write standard science and administrative data structures will be available for production stream file access. Examples of these tools might include write_image, read_image, read_palette, write_registered_image, read_browse_header.

3.3.2.3 Product and Auxiliary Data Access Tools

Product and Auxiliary data access refers to using the permanent files and data that exist outside of the execution of an individual production stream, such as science data products and archives, metadata and catalog information, and orbit, attitude, and digital terrain elevation data. As much as possible, the toolkit interfaces to these files and data sets will be implemented using well defined data structures. As mentioned above, a study to determine the data structure requirements of EOS investigators is under way. As investigators design the EOS standard data products, the Toolkit will need to support the component data structures which make up the products.

The tools within this category will have to be implemented differently on a SCF and a PGS, although the interface to the tools on both systems will be the same. In the DAAC environment, programs will have access to data and information provided through interfaces to the IMS and the DADS. When a data product is completed at the PGS, the Toolkit and PGS will pass metadata and browse products to the IMS, and pass standard products to the DADS for archiving. The SCFs will use the IMS or DADS emulation toolkits to support these activities.

3.3.2.3.1 Level 0 Data Access Tools

This group of tools will allow the program to access the raw level 0 data as either the raw packets or data reconstituted into the instruments output data stream. Tools to open, read and close level 0 files will be included in this set. The production program will have to provide an instrument specifier and a time period for the requested data.

3.3.2.3.2 Level 1 - 4 Product Access Tools

This tool set will allow the reading and writing of standard level 1 to 4 science data products. Input to and output from these product files will be in terms of standard structures. Tools will be defined for all standard structures and will be used to create these products.

It is obviously very important that investigators take strong interest in the study to determine the data structure requirements of EOS investigators. This study will be the basis for the definitions of the tools which will create the level 1 to 4 data sets.

3.3.2.3.3 Metadata and IMS Access Tools

This tool set will include tools to pass metadata and catalog information in the appropriate formats and data structures to the IMS on completion of data products. Tools for querying the IMS or DADS for the availability and content of data products may also be provided.

3.3.2.3.4 Telemetry & Command Access Tools

Relatively small adjustments to the spacecraft or other instruments during observation periods may affect science data. These tools will allow access to instrument and spacecraft command history files to provide investigators data to adjust processing parameters based on the estimated impacts of such actions.

3.3.2.3.5 Spacecraft Ephemeris & Attitude Data Access Tools

This tool set will provide spacecraft orbit and attitude data. Options will allow programmers to request data in various reference frames such as earth centered, earth fixed, or inertial. Tools will allow the user to request and report availability of predicted or precise ephemeris and attitude data. These tools will provide interpolation to given times or provide spans of time tagged ephemeris.

3.3.2.3.6 Lunar/Solar/Major Body Position Access Tools

This tool set will provide access to algorithms that compute the positions of major celestial bodies such as the Moon and Sun. Options will allow programmers to request positioning data in various reference frames such as spacecraft centered, earth centered or inertial. This information will provide programmers data to compute solar angle and determine if light from celestial bodies may be impacting instrument observations.

3.3.2.3.7 Instrument Calibration Data Access Tools

These tools provide access to calibration parameters that are derived by the instrument developers or measured during instrument testing and evaluation. These parameters will be under configuration management, accessible from the PGS, and used as part of the science production processing.

3.3.2.3.8 Time and Date Access Tools

The time and date access tools will provide access to current clock and calendar data and time conversion routines for translation between EOS and other time frames and different time formats. Functions may include Julian day conversion, transformation between satellite and UTC or GPS time, and day-of-week and time and date format functions to support metadata generation.

3.3.2.3.9 Browse Output Tools

As standard data products are produced, browse products will be generated to allow EOSDIS users to visually inspect representations of the data product. This tool set will provide the capability to create, write, and deliver browse files using defined browse data structures.

3.3.2.3.10 Quicklook Product Tools

Quicklook products are generated to test the health of the instrument or in response to a high priority science need. Tools will be provided to open and read unprocessed and Level 0 processed quicklook data from a single Tracking and Data Relay Satellite System (TDRSS) contact. Other than these special open and read requirements quicklook products will be generated using the same functions as those used to create standard products. Quicklook product production streams will request and receive a high priority from the Scheduler.

3.3.2.3.11 Error Output Tools

Error output tools will provide the ability for science software to report errors or problems to the PGS system managers and investigator teams.

3.3.2.3.12 Status Tools

The status tools will be used to report status of production streams and product availability to the PGS system managers and investigator teams.

3.3.2.3.13 Data Validation Graphic Output Tools

Visual inspection of data products by trained personnel is still one of the most accurate methods for recognizing and reporting data problems. However, the volume of data expected to be processed by EOSDIS precludes visual inspection of all data products. Science production software should be designed to automate data validation as much as possible. This tool set is included in order to provide production software the ability to present validation graphics to PGS system managers or investigators for data that has failed an automated inspection.

3.3.2.3.14 Geographic Standards Tools

This set of tools will provide science software developer teams a standard data base and set of methodologies for integrating geographic information into their products. It is hoped that the availability and use of these tools and database access functions will provide commonality and interoperability between the data products of various EOS disciplines and simplify many aspects of intra- and interdisciplinary data correlation.

3.3.2.3.14.1 Digital Terrain Elevation Data Access Tools

These tools will provide access to a database of a worldwide grid surface elevation data. Functions to interpolate for any given geographic coordinate, estimate elevation accuracy and precision, generate contour data, and produce mean sea-level, geodetic, or ellipsoid elevations are possible candidates for this tool group.

3.3.2.3.14.2 Geographic Information Access Tools

Geographic Information tools will provide access to various data bases including geo-political features, geoid height, geo-magnetics data, gravity anomaly data, and other geographic features and objects. Access to these tools will allow product developers to correlate data to geographic features, and to pass geopolitical metadata to the IMS.

3.3.2.3.14.3 Coordinate Transformation Tools

These tools will allow transformations from one coordinate reference frame to another. Transformations may be provided between earth-centered earth-fixed Cartesian, inertial, geographic, WGS-84, and other reference ellipsoids and frames.

3.3.2.3.15 Tools to Access Common Physical Constants

These tools will provide access to a data base of physical constants that will be used by many investigators and that should be standardized to provide commonality and interoperability between the data products. Examples of such constants might include pi, the speed of light, the earth's gravitational constant, and earth's angular velocity.

3.3.3 Math & Modeling Support Tools

The ECS RFP suggests that the PGS Toolkit will include standard mathematical operations such as matrix inversion and fast Fourier transforms. This functionality is likely to be provided by COTS libraries of math and statistics capabilities.

3.3.3.1 Math and Statistics Libraries

A survey is underway to determine the math and statistics library preferences of EOS science investigators and software developers. The functionality of these libraries will be incorporated into the Toolkit. IMSL and MATLIB 77 are examples of these libraries.

3.3.3.2 Image Analysis and Manipulation Tools

Although image presentation tools are not appropriate for the production environment of the PGS, routines to create, manipulate, and analyze image data structures to incorporate into data products will be included in the Toolkit. Such tools might include pattern search, pattern match, image overlay, palette definition, image statistics, pixel selection, area computation, and area selection functions. Some uses of these tools might include advanced automated data validation by pattern matching known geographic surface or related atmospheric features from another instrument's data sets, generating appropriate color palettes for a browse product, and using image statistics to measure the surface area of a particular surface or data feature.

3.3.3.3 High Performance Processing Tools

The Toolkit will be designed to be optimized for specialized PGS architectures. To as large a degree as possible, the Toolkit will be implemented using the vector processing or parallel processing capability the PGSs have to offer. Some investigators in special circumstances, however, may require direct access to architecture specific functions to achieve the performance levels necessary to process data in a timely manner.

Currently, these specialized processors come from a variety of vendors and do not have standard interfaces. However, in order to develop science software which makes use of such architectures on an SCF, special tools that emulate the PGS architecture-specific functions may be required.

3.3.4 Data Structure Manipulation Tools

This tool set will include functions to efficiently manipulate common data structures. Some examples include extracting subsets of data structures such as arrays, searching memory or data structures for specific data patterns, and sorting of data for product generation or for improving processing efficiency.

3.3.5 Memory Management Tools

Many processors have differing implementations for byte and word ordering, memory page and bank edges, alignments, and sizes that have to be taken into account during a porting process. The memory management tool set will be designed to normalize these differences in the SCF and PGS environments.

3.3.5.1 Bit, Byte and Word Manipulation Tools

Different processors often have individual requirements for internal data representation. Processors may have differing methods for aligning data items to word boundaries and may have varying machine and mass storage byte and word significance. This tool set will provide a common interface for manipulating bits, bytes and words.

3.3.5.2 Memory Request Tools

Science algorithms implemented in the C programming language may require requesting and maintaining memory space. Toolkit functions to provide this capability will be provided. However, in the Toolkit these tools will be subject to limits provided by the Scheduler for the program or production stream. In C memory request and related functions include malloc, calloc, realloc, and free.

3.3.6 Special Purpose Tools

The tools sets described in this section are tentatively included in this report with the caveat that additional study of the viability of these tools is required.

3.3.6.1 Remote Database Access Tools

Various databases of surface observed data are maintained at various agencies and organizations. These databases include data that are updated on a continual basis and may be accessible through a network. Some investigator's current production processes access such databases through a network in order to produce products that incorporate the most current and accurate information. This tool group would provide basic access to remote databases for this purpose. This tool set requirement requires further study. Tools of this nature could have a large impact on the PGS throughput and processing efficiency. Alternatives include pre-requesting and staging of data from external databases before production streams requiring such data are executed.

3.3.6.2 DAAC Specific Tools

The architecture and capabilities of the DAACs are not expected to be uniform. Each DAAC will likely have some value added features which will be specific to the needs of the science community being served by that DAAC. If required, the PGS Toolkit may incorporate tools which will provide access to the features which are not common to all DAACs.

3.3.6.3 Debugging Tools

This tool group would consist of a set of debugging support functions that would provide science programmers a common set of tools during code development and integration to the PGS. Examples of such tools might include functions to present variables and source code line pointers during execution; set debugging flags; and set breakpoints with variable, register, or memory dumps.

APPENDIX A TOOL TABLES

Appendix A will be used to present preliminary proposals for calling syntax, and input and output argument requirements and will be expanded to incorporate specific information pertaining to tools as it becomes definitive during this study. The following are the current proposed tool categories.

- Production Control & Scheduling Tools
  - Data Availability Tools
  - Initialization Tools
  - Termination Tools
- I/O Tools
  - Temporary File I/O Tools
    - Temporary File Housekeeping Tools
    - Temporary File Access Tools
  - Production Stream File I/O Tools
    - Production Stream File Housekeeping Tools
    - Production Stream File Access Tools
  - Product and Auxiliary Data Access Tools
    - Level 0 Data Access Tools
    - Level 1 - 4 Product Access Tools
    - Metadata and IMS Access Tools
    - Telemetry & Command Access Tools
    - Spacecraft Ephemeris & Attitude Data Access Tools
    - Lunar/Solar/Major Body Position Access Tools
    - Instrument Calibration Data Access Tools
    - Time and Date Access Tools
    - Browse Output Tools
    - Quicklook Product Tools
    - Error Output Tools
    - Status Tools
    - Data Validation Graphic Output Tools
    - Geographic Standards Tools
      - Digital Terrain Elevation Data Access Tools
      - Geographic Information Access Tools
      - Coordinate Transformation Tools
    - Tools to Access Common Physical Constants

- Math & Modeling Support Tools
    - Math and Statistics Libraries
    - Image Analysis and Manipulation Tools
    - High Performance Processing Tools
- Data Structure Manipulation Tools
- Memory Management Tools
    - Bit, Byte and Word Manipulation Tools
    - Memory Request Tools
- Special Purpose Tools
    - Remote Database Access Tools
    - DAAC Specific Tools
    - Debugging Tools

APPENDIX B TOOLKIT USAGE SCENARIOS

In later versions of this paper Appendix B will be used to present science software implementation scenarios as examples of how tools or tool sets might be used.