3. DPS - Data Processing Subsystem

3.1 Subsystem Overview

3.1.1 Introduction and Context

The Data Processing Subsystem is the collection of hardware and software components which are responsible for the management of the data processing resources at a provider site. These management responsibilities can be divided into the following general functional areas:

a. Managing the generation of Data Products and the operational environment used to produce these products.

b. Providing an Algorithm Integration and Test Environment for the introduction of science software into the EOSDIS environment.

c. Providing a Quality Assurance (QA) environment for testing the quality of data products.

The Data Processing Subsystem supports these functional areas through the following mechanisms:

a. It provides a queued processing environment to support the generation of data products. It manages, queues and executes Data Processing Requests on the processing resources at a provider site. A Data Processing Request can be defined as one processing job. Each Data Processing Request encapsulates all of the information needed to execute this processing job. Data Processing Requests are submitted from the Planning Subsystem; which in turn has been triggered by the arrival of data or internally through Planning itself (e.g., reprocessing). Data Processing Requests use Product Generation Executives (PGEs) to perform this processing. PGEs result from the integration and test of delivered science algorithms [ref.: ECS White Paper 193-00118] and also user specific methods into the subsystem. They will be encapsulated in the ECS environment through the SDP Toolkit. The Data Processing Subsystem also provides the Operational interfaces needed to monitor the execution of the science software (PGEs). More information on the execution of PGEs, and the support environment being provided to monitor the generation of data products is provided in Section 4, the Processing CSCI.

b. It supports the execution of science algorithms through the SDP Toolkit. The SDP Toolkit is a set of tools developed to standardize and provide a common interface for each science algorithm to the EOSDIS environment.

   The following documents provide guidance on the roles and responsibilities of the SDP Toolkit to support the execution of science software:

   333-CD-003-002 SDP Toolkit Users Guide for the ECS Project

   193-801-SD4-001PGS Toolkit Requirements Specification for the ECS Project, FINAL, 10/93 [AKA GSFC 423-06-02]

c. It supports the preliminary processing of data sets, i.e., L0 data products, which are required by the science algorithms, but are not in the proper format for use.

d. It provides the Algorithm Integration and Test environment used to integrate new science algorithms, new versions of existing science algorithms and user methods into the EOSDIS environment. The algorithm or method will be acquired by the system through an ingest client which will reflect local site policies on the acceptance of software for integration. Once acquired, the algorithm/method and its associated data files (test, calibration, etc.) will be registered in the local site Configuration Management (CM) system as part of the archival by the Data Server Subsystem. These activities are defined in more detail in Section 6, the Algorithm I&T CSCI.

e. It provides the DAAC Quality Assurance (QA) environment used to validate data products by personnel at the DAAC. All data products, both those produced by and input to a submitted job, can be examined by DAAC personnel to verify their content meets the quality standards set by that DAAC.

f. It provides a queued processing environment to support extended subsetting of data products in order to reduce data flows between DAACs. This extended subsetting consists of: (i) applying a geographic mask to set values for a portion of the pixels in a granule equal to a fill value, and then compressing the data to eliminate repeated fill values; (ii) decompressing previously compressed data to restore the fill values; or (iii) extracting the pixels between specified index values along a swath of the Level 1 or Level 2 instrument data. The specific algorithms and the data sets to which this extended subsetting will be applied are described in:

   160-TP-005-001 Reducing Inter-DAAC Data Transfers Through Subsetting.

   This extended subsetting will be performed by PGEs which are supported by the SDP Toolkit and which are initiated by Data Processing Requests from the Planning Subsystem. Therefore, these extended subsetting PGEs are supported by the same Data Processing Subsystem functions as the the science processing PGEs (see item a, above). Unlike the science processing PGEs, the extended subsetting PGEs will be supplied as part of the ECS development, and will be delivered with the ECS system.

3.1.2 Data Processing Subsystem Context

A context diagram illustrating the relationships between the Data Processing Subsystem and the other SDPS subsystems is shown in Figure 3.1-1. The key interfaces shown are:

a. The *Planning* interface is responsible for determining what processing activities are required to generate the data products as specified in a Production Request (Standard, Reprocessing, On Demand). These processing activities and associated information are defined and delivered as Data Processing Request(s) to the Data Processing Subsystem. One Production Request may result in one or more Data Processing Requests being sent to the Data Processing Subsystem. After the receipt of a Data Processing Request, the Data Processing Subsystem will deliver processing status to Planning, when requested. Also, provided by the Data Processing Subsystem in an unrelated off-line activity is information used to plan the execution of a PGE. This information is determined by the Algorithm (Science Software) Integration & Test services and provided to Planning and Processing by addition data to the PDPS Database.

b. The *Data Server* interface is for requesting access to data required as an input to a PGE and for requesting that generated output data be transferred to the Data Server. Also, The *Data Server* is used to archive PGEs and associated testing data during Algorithm Integration & Test.

c. The *Ingest* interface is for requesting access to Level 0 data to be pre-processed prior to being input to a PGE.

d. The *MSS* interface provides access to common ECS system management services and APIs to pass accounting and configuration management information.

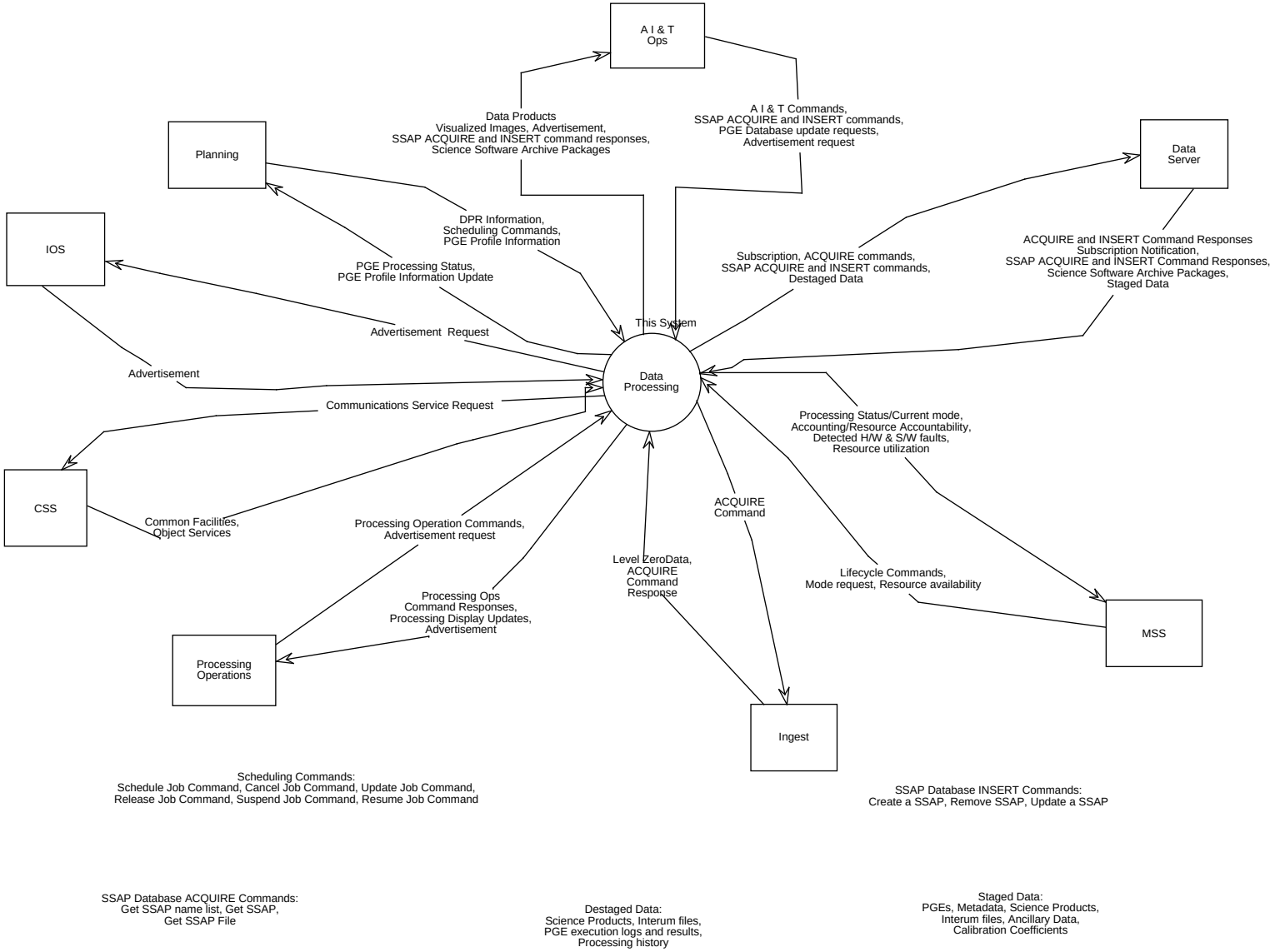

Figure 3.1-1. Context Diagram

In Table 3.1-1, where an exact number is unavailable, the data volume is estimated as low (less than 1 MB), medium (between 1 MB and 1 GB), or high (greater than 1 GB) per use defined in the frequency column.)

Processing requirements are driven by the frequency of Production Requests and their associated Data Processing Requests which vary by site. Therefore, processing volume and resulting capacity requirements are discussed in the DAAC specific documents of this DID.

Table 3.1-1. Subsystem Interfaces (1 of 2)

Source	Destination	Data Types	Data Volume	Frequency
Data Processing	Data Server	Acquire Commands (Staging Requests)	low	as required for processing
Data Processing	Data Server	Subscription Requests	low	as required for processing
Data Processing	Data Server	standard products	high	as required for processing
Data Processing	Data Server	Metadata	high	as required for processing
Data Processing	Data Server	q/a data	medium	as required for processing
Data Processing	Data Server	Science Algorithms	low	as requested
Data Processing	Planning	Schedule Job Command Response	low	In Response to Schedule Job Command
Data Processing	Planning	Cancel Job Command Response	low	In Response to Cancel Job Command
Data Processing	Planning	Suspend Job Command Response	low	In Response to Suspend Job Command
Data Processing	Planning	Resume Job Command Response	low	In Response to Resume Job Command
Data Processing	Planning	Release Job Command Response	low	In Response to Release Job Command Response
Data Processing	Planning	Update Job Command Response	low	In Response to Release Job Command Response
Data Processing	Planning	Processing Status	low	In Response to Request For Job Status
Data Processing	Planning	PGE Profile Information	low	as required
Data Processing	MSS	System Management Information (Accounting, Scheduling, Fault, Accountability, Security, and Performance)	low	as required
Data Processing	Operations	Processing Operations Information	low	in response to request
Data Processing	Operations	AI&T Operations Information	low	in response to request
Data Processing	CSS	Communication Service Request	low	as required
Data Processing	Interoperability (IOS)	Advertisement requests	low	as required
Planning	Data Processing	Schedule Job Command	low	in response to activating a Plan

Table 3.1-1. Subsystem Interfaces (2 of 2)

Source	Destination	Data Types	Data Volume	Frequency
Planning	Data Processing	Cancel Job Command	low	in response to activating a Plan
Planning	Data Processing	Suspend Job Command	low	in response to activating a plan
Planning	Data Processing	Resume Job Command	low	in response to activating a plan
Planning	Data Processing	Release Job Command	low	in response to activating a Plan
Planning	Data Processing	Update Job Command	low	in response to activating a Plan
Planning	Data Processing	Processing Status	low	in response to activating a Plan.
Planning	Data Processing	PGE Profile Information	low	as required
Data Server	Data Processing	Standard Products	high	in response to staging request
Data Server	Data Processing	Metadata	medium	in response to staging request
Data Server	Data Processing	Ancillary Data	high	in response to staging request
Data Server	Data Processing	Calibration Data	medium	in response to staging request
Data Server	Data Processing	Orbit/Attitude Data	medium	in response to staging request
Data Server	Data Processing	Algorithms	low	in response to staging request
Data Server	Data Processing	Science Software Delivery	low	as required
MSS	Data Processing	resource fault information	low	as required
MSS	Data Processing	resource performance utilization information	low	as required
CSS	Data Processing	Common facilities Object Services	low	as required
Interoperability	Data Processing	Advertisement response	low	as required
Operations	Data Processing	Processing Operations Commands	low	as required
Operations	Data Processing	AI&T Operations Commands	low	as required

3.2 Subsystem Overview

The following sections summarize the hardware and software structure of the Data Processing Subsystem and provide the design rationale used during the Detailed Design process.

3.2.1 Subsystem Structure

The Data Processing Subsystem is divided into the following six CSCIs and HWCIs:

- PRONG – Processing CSCI—The Processing CSCI provides the services required to manage and monitor the Science Data Processing environment which is used to generate data products using Science Software (PGEs) provided by the instrument teams. These services are discussed further in Section 4, the Processing CSCI.

- SDPTK – SDP Toolkit CSCI—The SDP Toolkit CSCI provides a set of software libraries which are used to integrate Science Software into the EOSDIS environment. By promoting the POSIX standard, these libraries allow the Science Data Processing environment to support the generation of data products in a heterogeneous computer hardware environment. The following documents provide guidance on the roles and responsibilities of the SDP Toolkit to support the execution of science software:

  333-CD-003-002 SDP Toolkit Users Guide for the ECS Project, 8/95

  193-801-SD4-001PGS Toolkit Requirements Specification for the ECS Project, FINAL, 10/93 [AKA GSFC 423-16-02]

- AITTL – Algorithm I&T CSCI—The Algorithm I&T CSCI is a set of tools which are used to integrate and test new science software, new versions of science software and user methods into the Science Data Processing operational environment. These services are discussed further in Section 6, the Algorithm I&T CSCI.
- SPRHW – Science Processing HWCI—The Science Processing HWCI is the collection of hardware resources being provided to support the generation of data products in a secure, monitored environment. These services are discussed further in Section 7, the Science Processing HWCI.
- AQAHW – Algorithm QA HWCI—The Algorithm QA HWCI is the collection of hardware resources being provided to support DAAC manual quality assurance activities. These services are discussed further in Section 8, the Algorithm QA HWCI.
- AITHW – Algorithm Integration & Test HWCI—The Algorithm Integration & Test HWCI is the collection of hardware resources being provided to support the integration and testing of Science Software in a secure, monitored environment. These services are discussed further in Section 9, the Algorithm Integration & Test HWCI.

3.2.2 Subsystem Design Rationale

The division of the subsystem into its current CSCIs and HWCIs has been driven by many different factors, but two are of particular importance:

a. Science algorithm development support

b. Efficient use of the Science Data Processing Environment

SDP Toolkit CSCI - because of the need to support the development of science algorithms, the SDP Toolkit was an early deliverable. Therefore, a need arose to group this software at a CSCI level for accountability reasons. The SDP Toolkit is an extensive software library with multiple language bindings. It is being incorporated into the science software to avoid the duplicate development of commonly used tools, in addition to providing a standard interface for integration of science software into the ECS. In particular, the library provides interfaces for accessing ECS data products, product metadata, ancillary data and processing parameter information. This Toolkit was needed by the Instrument Teams at an early date to support development, testing and early integration of science software in the Ir1 and Release A timeframes in addition to Release B.

The Processing CSCI was developed to provide a means of monitoring and managing the queued generation of scientific data products. DAAC Science Data Processing is not a real-time, automatically invoked, single data stream, uniform process such as spacecraft telemetry processing. Science data must be collected, stored, prepared, and "batched" for generation of lower-level data products. Processing occurs post (Satellite) pass and must be planned on the basis of data quality and availability, and the availability of appropriate system resources. The appropriate planning of the generation of data products is performed by the Planning Subsystem (see Planning Subsystem Detailed Design Specification for more details.). In support of Science Data Processing, the Processing CSCI provides the following capabilities:

1. Provides effective resource management for efficient use of hardware resources.
2. Supports error recovery mechanisms to allow the generation of data products in an operator attended or unattended processing environment.
3. Provides mechanisms to support many different Science Processing hardware configurations which are needed to support the differing aspects of data processing which occurs at each DAAC site.
4. Provides data pre-processing support functions for the introduction of Level 0, Ancillary data, Orbit and Attitude data into the ECS. This data may be received in formats which are not usable by the science software, and therefore, must undergo reformatting and/or other types of preparation.

The Algorithm I&T CSCI, on the other hand, is the grouping of tools and support functions related to the integration of science software and user methods into the ECS. This CSCI is a standalone function and requires special attention to support a secure, standalone Algorithm I&T environment.

The division of the Hardware CSCIs was driven by the need to support a Science Data Processing environment which would not be affected by the Algorithm I&T environment. The hardware to support the Science Data Processing Environment is grouped as the Science Processing HWCI. The Algorithm I&T environment is grouped as the Algorithm Integration & Test HWCI. The Algorithm QA HWCI was created to support DAAC manual quality assurance activities. This HWCI must support the display of generated data products and the updating of quality assurance metadata associated with a data product.

3.2.2.1 Performance

The Data Processing Subsystem design in Release B has been prepared to meet the overall performance objectives of the EOS-AM1 Release period. The performance for the subsystem for a given workload is determined by both the hardware and the software designs for the system.

The performance of the PRONG CSCI of the Data Processing Subsystem will be strongly affected by the COTS products, Platinum Technology's AutoSys and AutoXpert scheduling tools, that have been procured in the Release A timeframe. Prior to the release of the request for proposals for this tool, several such packages, including AutoSys were evaluated by HAIS. Certain performance measures were taken, and this information has been used in the design of the platforms to be used to support this element of the Data Processing Subsystem. Informal prototyping activities of the AutoSys and AutoXpert packages have been initiated and will continue until CDR, both to evaluate details of the COTS implementation and to evaluate the performance of the COTS package.

Results of this evaluation will be analyzed and may influence the details of the hardware sizing for the platforms that will support the PRONG CSCI.

Details concerning the performance of the platforms supporting the PRONG CSCI, including considerations of the RMA requirements, are included in the sections in this document devoted to the associated hardware platforms. It should be noted that the design for the PRONG CSCI and associated software reflects the high reliability option for AutoSys and AutoXpert, including redundant processors and high reliability Sybase servers. This will insure that the mean time to restore service is within the 30 minute recovery time required.

In addition, the design of the Planning and Data Processing Subsystem software have been prepared to minimize needless disk I/O and maximize local storage resources. Needless disk I/O is minimized by retaining on local disk the output files of one PGE, and scheduling the execution of another PGE that uses those files to occur soon afterwards. Local disk storage is maximized by planning the staging of data so that the data is available when needed, but does not take up local disk resources for an excessive period of time before the PGE that requires it begins execution. This way, needless disk I/O and network loading is avoided, and local storage is maximized, ensuring that the ECS performance goals can be met with the least cost.

In some cases, even with the minimization of needless disk I/O the process of staging data prior to PGE execution may take a significant period of time, and hence lead to delays in PGE throughput. In order to address this issue, the ability to predictively stage data for PGEs has been added to the Release B design. This involves the loading of data ahead of the time at which the PGE is expected to run in order to further minimize the data transfer that occurs as the PGE prepares to execute.

To improve performance of inter-DAAC transfers, extended subsetting PGEs will in addition be provided as part of the Release B PDPS. Further details of these can be found in document 160-TP-005-001, Reducing Inter-DAAC Transfers Through Subsetting.

3.2.3 Data Processing Use of Key Design Mechanisms

The Data Processing Subsystem uses a number of key design mechanisms developed by ECS to encapsulate much of the interprocess communication and to handle the infrastructure services. Inheriting from or reusing this code saves on design and programming effort and increases maintainability.

3.2.3.1 Universal References/Advertising

Universal references are the key mechanism used by ECS to locate services and products; they are encapsulated by the object 'EcUrUR'. These references allow different parts of ECS to communicate about the services and products they provide. The Data Processing Subsystem uses Universal References within the Data Manager CSC of the Processing CSCI. The DpPrDataManager object uses the EcUrUR object in order to identify data granules in requests to the Data Server Subsystem. Universal References are also received from the Data Server Subsystem as the result of queries.

3.2.3.2 Distributed Object Framework

The Distributed Object Framework allows the intercommunication of processes across machines at an object level. No objects belonging to the Data Processing Subsystem are distributed; however, the Data Manager CSC of the Processing CSCI uses the Data Server distributed object DsClESDTReferenceCollector as part of the standard Data Server interface. This distributed object is also used by the DpAtSSAPGuiNB object in the AITTL CSCI.

3.2.3.3 Event Handling

Event handling is encapsulated by the EcEvent object. Events include abnormal conditions (errors or faults) as well as normal occurences. Not all events that occur within the system are of interest from a system operations and management perspective; those that are are called managed events, which must be reported and logged. This mechanism handles messages that require operator notification within the Data Processing Subsystem, as well as notification to a centralized MSS event logging facility. All software CSCIs within the Data Processing Subsystem use the Event Handling key mechanism, including the SDP Toolkit CSCI. In addition, notifications are made to the operator via Autosys COTS GUI notices.

3.2.3.4 Managed Process Framework

All Processing applications except the Autosys scheduling COTS use the managed process framework to handle mode management, MSS agent interface and the OODCE infrastructure interface. With the exception of the Autosys COTS all Data Processing programs are "unmanaged clients" that are executed briefly when necessary. The Autosys COTS is handled by a Proxy Agent as an "unmanaged server".

3.2.3.5 Mode Management

Mode Management is supported by all Data Processing Subsystem programs. It may be used in order to ensure that Science Software integration and test activities do not affect Production activities by creating an "AIT" mode version of PDPS running on separate hardware to the "Production" mode version.