PDS4 Information Model Specification

PDS Data Model Team

September 10, 2007

DRAFT

Contents

List of Figures

1 Introduction

This document presents a draft engineering specification for the Planetary Data System (PDS) Information Model. Commonly called the Draft PDS4 Information Model (DPIM) it was initiated as an engineering specification for the PDS3 standards and is a working draft that will evolve to meet PDS4 information model requirements. Its source was the information model specification developed for the International Planetary Data Alliance (IPDA) as part of their Archive Data Standards project. As shown in the highlighted sections of Figure 1, the focus of the project was the Information Model component of the Data Architecture Standards, namely the object models, a data dictionary, and a set of data formats.

1.1 Background

The International Planetary Data Alliance (IPDA) is a joint effort by national space exploration agencies, research institutions, and universities to enable global access and exchange of high quality planetary science data, and to establish archive standards that make it easier to share the data across international boundaries.

The IPDA has defined a Reference System Architecture that will provide a set of best practice specifications to be used for guiding the implementation of archive data systems. This reference architecture, outlined in Figure 1, consists of three core components, namely the process, data, and technology architecture standards. These standards will provide the means for enabling interoperability between planetary science archive data systems.

1.2 Scope

The term Information Model is used for this deliverable for two reasons. First, it comprehensively defines a large and complex domain using several related object models and a data dictionary. In this document the term object model means an abstract model that describes how data is represented and which uses object-oriented concepts, namely object classes as the key modeling construct. Secondly, an Information Model is the foundation on which an information system is built. It guides the systems design and implementation by identifying and defining the items to be processed, the context for the items, and the relationships that provide meaning.

For this project the Information Model was captured in an ontology modeling tool. This allows an Information Model to be developed and maintained independent from any implementation choices. Typically an Information Model evolves at a speed different from and outlasts any

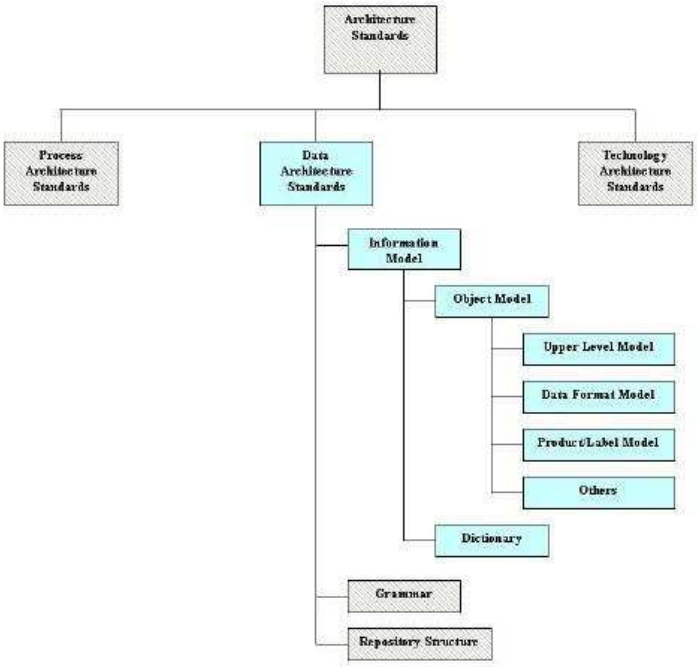

Figure 1: IPDA Reference System Architecture

implementation technology choice.

The Information Model consists of three key related object models and the data dictionary. The upper level object model defines the object classes that exist in the planetary science community. These include object classes such as mission, instrument, and data set and provide the science and programmatic context within which data products are collected and archived. The data format object model defines the object classes that describe the logical and physical structure of the digital data to be archived and include such commonly used object classes as Image and Table. The data product object model includes the object classes that are used to package the data and instances of object classes (metadata) that describe the data. For example an image data product is a package that contains a digital image, an instance of the Image object class that describes the structure of the digital image, and additional descriptive information for understanding and using the product. Finally, the data dictionary is the set of attributes that have been used in the object class definitions.

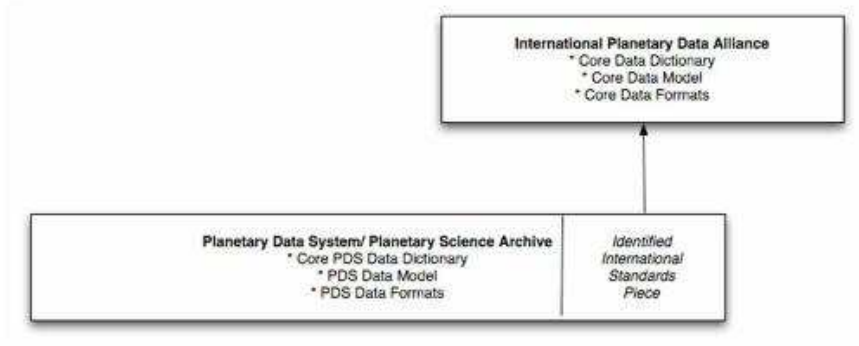

Figure 2: Project Approach

1.3 Approach

The projects approach is illustrated in Figure 2, namely documenting the current PDS data model and then identifying core elements for consideration as archive data standards for the IPDA.

As previously mentioned, the current PDS data model has been captured in an ontology modeling tool. Several sources have been used, including the Planetary Science Data Dictionary (PSDD), elements of the planetary science archive repository, and design documents produced during the PDS design and implementation phases from 1988 through 1990. In addition, each PDS discipline node has submitted commonly used, well-formed data products from their local archive repositories to be considered in identifying the set of core data formats. The resulting database is the Information Model captured in this document.

The Information Model has been documented in the following sections by exporting the ontology modeling tool database to various ASCII file formats and then transforming the information into LaTEX format. The first section presents the upper level object model and includes a brief introduction, a class hierarchy tree, a Unified Modeling Language (UML) class hierarchy diagram, and finally the individual classes presented in a table format with their class hierarchy, attributes, and associations. Figure 3 presents a conceptual view of some the object classes in PDS data model. The upper level object classes appear in the upper part of the figure and include object classes related to Data Set such as Instrument and Target.

The next section presents the data format object classes. Data format object classes appear in the bottom right of Figure 3 and include Image

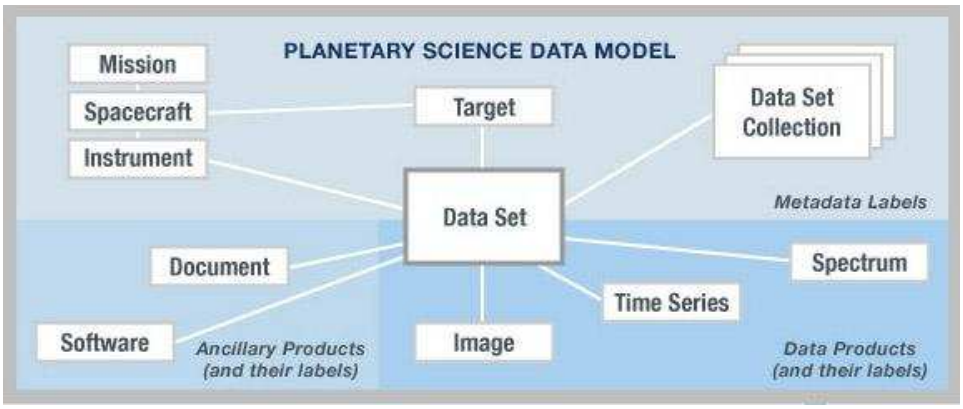

Figure 3: Conceptual PDS Data Model

and Time Series. Figure 4 shows a portion of a PDS image product label as an example of the how the PDS describes the data format of a digital camera image in an Object Description Language (ODL) data product label.

There are three groups of data format object classes presented in this section. The first group includes the data formats as currently defined in the PDS archive data standards with both required and optional attributes. The second group includes each PDS data format defined using only the required attributes from the PDS standards. This group forms the parent classes for this object classes in this section. The final group consists of the proposed core data formats. These result from considering the submitted example data products. Similar to the previous section, this section includes an introduction, a class hierarchy tree, a UML class diagram, and finally the individual classes presented in a table format. The section immediately following contains the data format object classes defined from the original data products submitted by the nodes.

The data product object classes have been separated into two sections, the first focusing on the object classes needed to define the components of a data product label and the latter on classes of data products. Both sections are presented in a manner similar to the upper level and data format models. As is evident in Figure 5, many of the components of a data product and its label are not formally defined as object classes but are simply groups of attributes delimited by comments. For the Information Model, these groups have been defined as object classes.

The data dictionary is presented in a simple definitional layout and includes only those attributes used in the object class definitions.

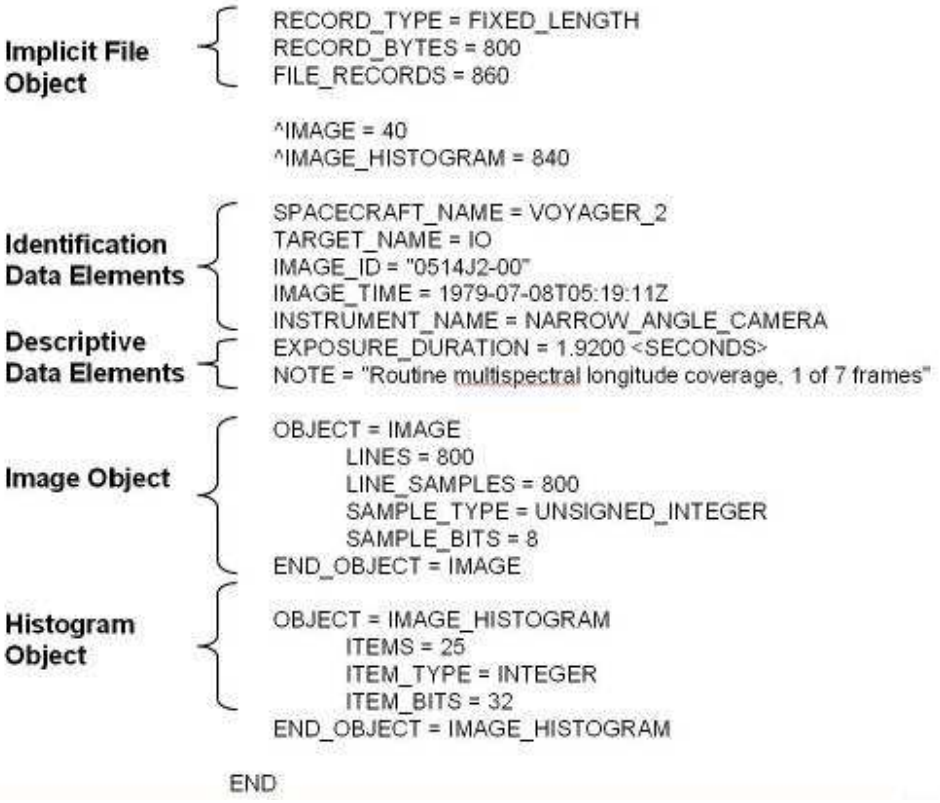

Figure 4: PDS Label for Image Data Product

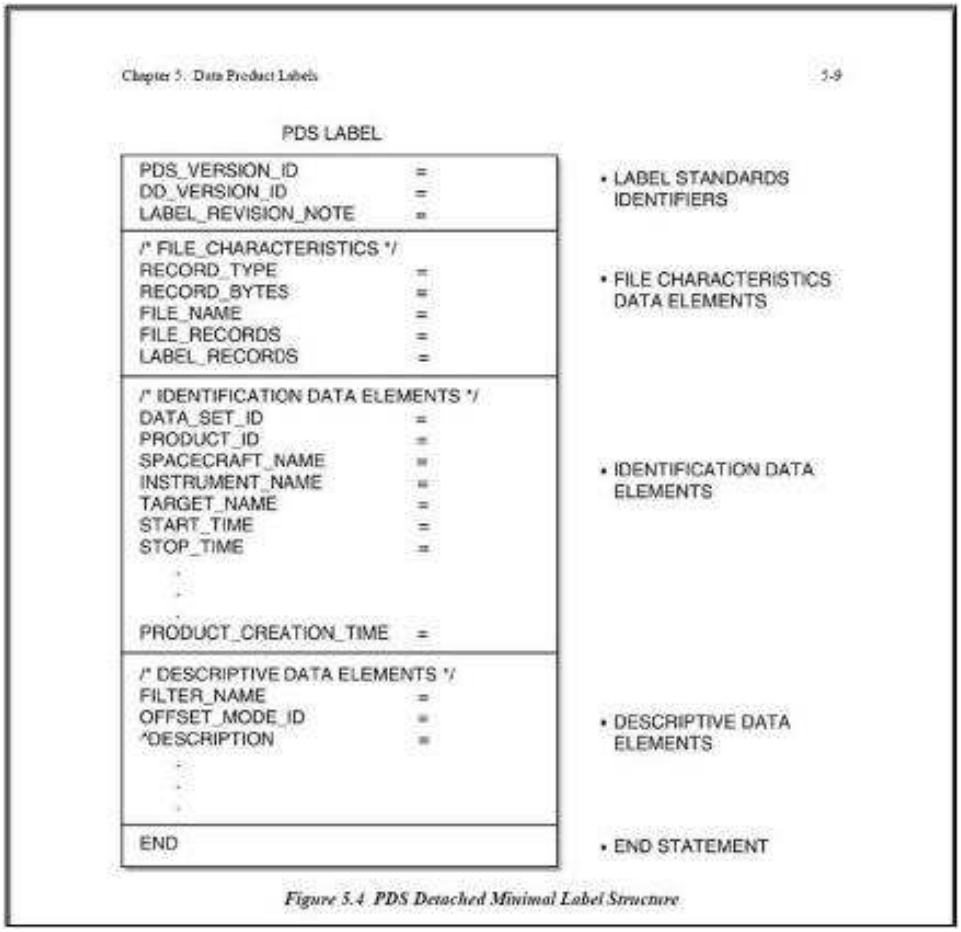

Figure 5: PDS Product Label

2 Upper Level Object Classes

The upper level object model describes the object classes that exist in the planetary science community and that provide a context within which science data products are collected, located, and used. For example, the Mars Viking Digital Image Mosaic is a data set created from images that were collected by the two vidicon cameras that flew on the Viking Orbiters. The upper level object model provides object classes such as planetary missions, instruments, and data sets that are subsequently used to create objects that describe the Viking mission, the two Vidicon cameras, and the resulting data set. These objects and their relationships provide the context for the digital images collected.

The upper level object class hierarchy is illustrated in the following diagram. This diagram presents the subclassOf relation for each object class in a hierarchical (tree) format and provides a visual representation of the object classes in relation to their parent classes. As currently modeled the upper level class hierarchy is flat however it will become more complex as we continue to develop the model.

```
. Upper_Level_Object_Description
. . Data_Set
. . Instrument
. . Instrument_Host
. . . Earth_Based
. . . Rover
. . . Spacecraft
. . Mission
. . Node
. . Personnel
. . Personnel_Electronic_Mail
. . Reference
. . Resource
. . Target
. . Volume
```

The class hierarchy above includes 15 unique classes.

The upper level object model is illustrated using the UML class hierarchy diagram in Figure 6. This diagram describes the object classes that belong to the planetary science domain and that provides a context in which scientific data products are collected, located, and used. The relations between object classes are one directional. Inverse relations are defined when necessary. For example, to model the many-to-many relation between the data set and target object classes, the has_Target relation

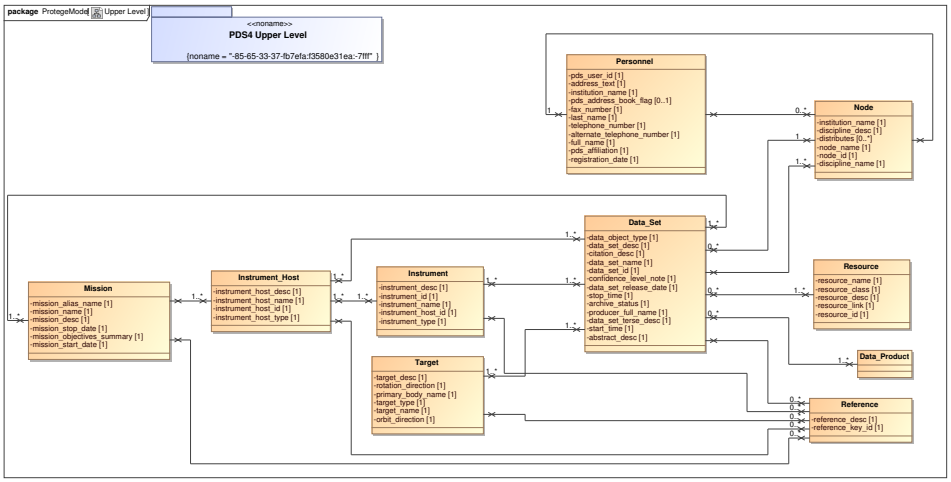

Figure 6: Upper Level UML Class Diagram

relates the data set object class to the target object class. The inverse, has_Target_I, relates the target object class back to the data set object class. The following sections present the upper level object classes in a table format. The table includes the class hierarchy, class attributes, and class associations. The class attributes and associations listed include both those used to define the object class and those inherited from parent classes. Cardinalities are provided where appropriate.

2.1 DATA_SET

Object Type: Upper_Level_Object_Description
Object Description: A collection of related data products

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description . Data_Set		
Attribute	abstract_desc archive_status citation_desc confidence_level_note data_object_type data_set_desc data_set_id data_set_name data_set_release_date data_set_terse_desc producer_full_name start_time stop_time	1 1 1 1 1 1 1 1 1 1 1 1 1	
Inherited Attribute	none		
Association	curated_by distributed_by has_Host has_Instrument has_Mission has_Product–Implicit has_Reference has_Resource has_Target has_Volume	1 1..* 1..* 1..* 1..* 1..* 0..* 1..* 1..* 1..*	Node Node Instrument_Host Instrument Mission Data_Product Reference Resource Target Volume
Inherited Association	none		

2.2 EARTH_BASED

Object Type: Upper_Level_Object_Description
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description . Instrument_Host . . Earth_Based		
Attribute	instrument_host_type	1	Earth_Based
Inherited Attribute	instrument_host_desc instrument_host_id instrument_host_name	1 1 1	
Association	none		
Inherited Association	has_Host_I has_Host_Instrument_I has_Reference	1..* 1..* 0..*	Data_Set Instrument Reference

2.3 INSTRUMENT

Object Type: Upper_Level_Object_Description
Object Description: An entity that collects data.

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description . Instrument		
Attribute	instrument_desc instrument_host_id instrument_id instrument_name instrument_type	1 1 1 1 1	
Inherited Attribute	none		
Association	has_Host_Instrument has_Instrument_I has_Reference	1..* 1 0..*	Instrument_Host Data_Set Reference
Inherited Association	none		

2.4 INSTRUMENT_HOST

Object Type: Upper_Level_Object_Description
Object Description: An entity upon which an instrument is mounted

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description . Instrument_Host		
Attribute	instrument_host_desc instrument_host_id instrument_host_name instrument_host_type	1 1 1 1	
Inherited Attribute	none		
Association	has_Host_I has_Host_Instrument_I has_Reference	1..* 1..* 0..*	Data_Set Instrument Reference
Inherited Association	none		

2.5 MISSION

Object Type: Upper_Level_Object_Description
Object Description: An entity reponsible for managing a project directed toward the collection of data.

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description . Mission		
Attribute	mission_alias_name mission_desc mission_name mission_objectives_summary mission_start_date mission_stop_date	1 1 1 1 1 1	
Inherited Attribute	none		
Association	has_Mission_Host has_Mission_I has_Reference	1..* 1..* 0..*	Instrument_Host Data_Set Reference
Inherited Association	none		

2.6 NODE

Object Type: Upper_Level_Object_Description
Object Description: An entity responsible for the management of science data that is associated with a specific planetary science discipline

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description . Node		
Attribute	discipline_desc discipline_name institution_name node_id node_name	1 1 1 1 1	
Inherited Attribute	none		
Association	curates da_contact distributes node_manager operations_contact	1..* 1 1..* 1 1	Data_Set Personnel Data_Set Personnel Personnel
Inherited Association	none		

2.7 PERSONNEL

Object Type: Upper_Level_Object_Description
Object Description: A person which has an association with the planetary science community

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description . Personnel		
Attribute	address_text alternate_telephone_number fax_number full_name institution_name last_name pds_address_book_flag pds_affiliation pds_user_id registration_date telephone_number	1 1 1 1 1 1 0..1 1 1 1 1	
Inherited Attribute	none		
Association	has_Electronic_Mail is_affiliated_with	0..* 0..*	Personnel_Electronic_Mail Node
Inherited Association	none		

2.8 PERSONNEL_ELECTRONIC_MAIL

Object Type: Upper_Level_Object_Description
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description . Personnel_Electronic_Mail		
Attribute	electronic_mail_id electronic_mail_type preference_id	1 1 1	
Inherited Attribute	none		
Association	none		
Inherited Association	none		

2.9 REFERENCE

Object Type: Upper_Level_Object_Description
Object Description: An entity providing a citation reference to a publication

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description . Reference		
Attribute	reference_desc reference_key_id	1 1	
Inherited Attribute	none		
Association	none		
Inherited Association	none		

2.10 RESOURCE

Object Type: Upper_Level_Object_Description
Object Description: An entity providing information about a PDS resource

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description . Resource		
Attribute	resource_class resource_desc resource_id resource_link resource_name	1 1 1 1 1	
Inherited Attribute	none		
Association	has_Resource_I	0..1	Data_Set
Inherited Association	none		

2.11 ROVER

Object Type: Upper_Level_Object_Description
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description . Instrument_Host . . Rover		
Attribute	instrument_host_type	1	Rover
Inherited Attribute	instrument_host_desc instrument_host_id instrument_host_name	1 1 1	
Association	none		
Inherited Association	has_Host_I has_Host_Instrument_I has_Reference	1..* 1..* 0..*	Data_Set Instrument Reference

2.12 SPACECRAFT

Object Type: Upper_Level_Object_Description
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description . Instrument_Host . . Spacecraft		
Attribute	instrument_host_type	1	Spacecraft
Inherited Attribute	instrument_host_desc instrument_host_id instrument_host_name	1 1 1	
Association	none		
Inherited Association	has_Host_I has_Host_Instrument_I has_Reference	1..* 1..* 0..*	Data_Set Instrument Reference

2.13 TARGET

Object Type: Upper_Level_Object_Description
Object Description: An entity which is the object of data collection

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description . Target		
Attribute	orbit_direction primary_body_name rotation_direction target_desc target_name target_type	1 1 1 1 1 1	
Inherited Attribute	none		
Association	has_Reference has_Target_I	0..* 1	Reference Data_Set
Inherited Association	none		

2.14 UPPER_LEVEL_OBJECT_DESCRIPTION

Object Type: Upper_Level_Object_Description
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Upper_Level_Object_Description		
Attribute	none		
Inherited Attribute	none		
Association	none		
Inherited Association	none		

2.15 VOLUME

Object Type: Upper_Level_Object_Description
Object Description: An entity that organizes science data

Relationship	**Entity**	**Card**	**Value**
Hierarchy	Upper_Level_Object_Description . Volume		
Attribute	medium_type publication_date volume_desc volume_format volume_id volume_name volume_set_id volume_version_id	1 1 1 1 1 1 1 1	
Inherited Attribute	none		
Association	has_Document has_Software has_Volume_I	0..* 0..* 1	Document Software Data_Set
Inherited Association	none		

3 Data Format Object Classes

The data format model defines the data format object classes to be used to describe the structure of data objects. For example, an Image object class uses attributes to define an image data object as a two-dimensional array of values, all of the same type, each of which is referred to as a sample.

The PDS archive data standards provide a set of generic object classes for this purpose and are included in the Information Model. These object classes have two sets of attributes, namely required and optional. The optional attributes are employed as needed in the design of a label for a data object. Also included in the Information Model are base object classes created by considering only the required attributes for each PDS generic object class. Finally a set of proposed core data formats are included that were created by combining the data formats extracted from the example data products submitted by the PDS discipline nodes. As a simple example, VICR HEADER objects submitted by the nodes were very similar and so were logically combined, resulting in a core VICR HEADER object class.

The data format object class hierarchy is illustrated in the following diagram. This diagram presents the subclassOf relation for each object class in a hierarchical (tree) format and provides a visual representation of the object classes in relation to their parent classes.

```
. Data_Object_Description
. . Alias
. . Array
. . . Array_Core
. . Bit_Column
. . . Bit_Column_Core
. . Column
. . . Column_Core
. . Container
. . . Container_Core
. . Document
. . Element
. . . Element_Core
. . Field
. . . Field_Core
. . File
. . . Explicit_File
. . . Implicit_File
. . . . Implicit_File_Attached
. . Header
```

```
. . . Header_FITS_Core
. . . Header_VICAR_Core
. . Histogram
. . . Histogram_Core
. . Image
. . . Banded_Image_Core
. . . Simple_Image_Core
. . Palette
. . Software
. . Software_Online
. . Spreadsheet
. . . Spreadsheet_Core
. . Table
. . . Series
. . . . Series_Core
. . . . Time_Series
. . . . . Time_Series_Core
. . . Series_Binary
. . . . Series_Binary_Core
. . . . Time_Series_Binary
. . . . . Time_Series_Binary_Core
. . . Spectrum
. . . . Spectrum_Core
. . . Table_ASCII
. . . . Table_ASCII_Core
. . . . . Table_ASCII_Key_Core
. . . Table_Binary
. . . . Table_Binary_Core
. . . . Table_Binary_Keyed_Core
. . . Table_Keyed_Core
. . Text
. . . Text_Core
```

The class hierarchy above includes 52 unique classes.

The data format object classes are illustrated using a UML class hierarchy diagram in Figure 7. This diagram defines the object classes that are used to describe how the digital bits or the data object is structured. The following sections present the data format object classes in a table format. The table includes the class hierarchy, class attributes, and class associations. The class attributes and associations listed include both those used to define the object class and those inherited from parent classes. Cardinalities are provided where appropriate.

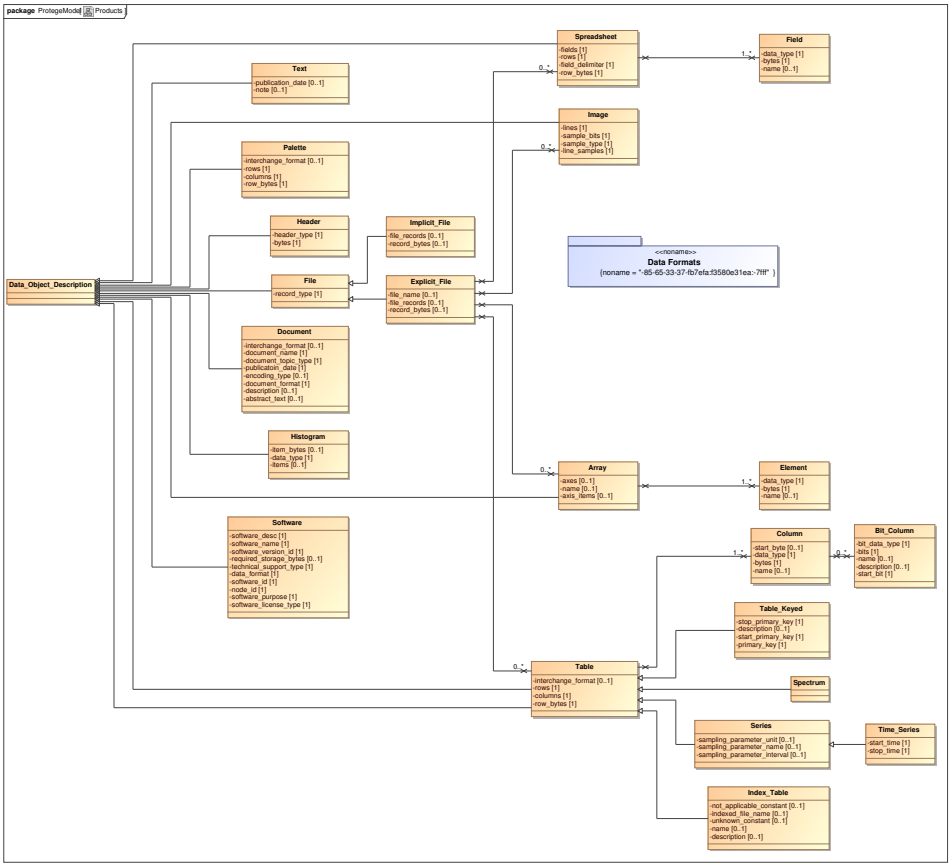

Figure 7: Datal Format UML Class Diagram

3.1 ALIAS

Object Type: Data_Object_Description
Object Description: The ALIAS object provides a method for identifying alternate terms or names for approved data elements or objects within a data system. The ALIAS object is an optional sub-object of the COLUMN object.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Alias		
Attribute	alias_name usage_note	1 1	
Inherited Attribute	none		
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.2 ARRAY

Object Type: Data_Object_Description
Object Description: The ARRAY object is provided to describe dimensioned arrays of homogeneous objects.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Array		
Attribute	axes axis_items name_	1 1 1	
Inherited Attribute	none		
Association	has_Element	1..*	Element
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.3 ARRAY_CORE

Object Type: Data_Object_Description
Object Description: Derived from data products from the followng data sets DI/EAR-C-KECK1LWS-3-9P-IMAGES-PHOT-V1.0, MEX-Y/M-SPI-2-IREDR-RAWXCRUISE/MARS-V1.0

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Array . . Array_Core		
Attribute	axis_name axis_order_type description interchange_format start_byte	0..1 0..1 0..1 0..1 0..1	
Inherited Attribute	axes axis_items name_	1 1 1	
Association	has_Element	1..*	Element_Core
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.4 BANDED_IMAGE_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data set: CLEM1-L-U-5-DIM-UVVIS-V1.0, MRO-M-CRISM-2-EDR-V1.0

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Image . . Banded_Image_Core		
Attribute	band_name band_storage_type bands checksum high_instr_saturation high_repr_saturation low_instr_saturation low_repr_saturation maximum minimum null offset sample_bit_mask scaling_factor valid_maximum valid_minimum	0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1	
Inherited Attribute	line_samples lines sample_bits sample_type	1 1 1 1	
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.5 BIT_COLUMN

Object Type: Data_Object_Description
Object Description: The BIT_COLUMN object identifies a string of bits that do not fall on even byte boundaries and therefore cannot be described as a distinct COLUMN. BIT_COLUMNs defined within columns are analogous to columns defined within rows.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Bit_Column		
Attribute	bit_data_type bits description name_ start_bit	1 1 1 1 1	
Inherited Attribute	none		
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.6 BIT_COLUMN_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data sets: CO-V/E/J/S/SS-RPWS-3-RDR-LRFULL-V1.0, MGS-M-RSS-1-EXT-V1.0.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Bit_Column . . Bit_Column_Core		
Attribute	unit	0..1	
Inherited Attribute	bit_data_type bits description name_ start_bit	1 1 1 1 1	
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.7 COLUMN

Object Type: Data_Object_Description
Object Description: The COLUMN object identifies a single column in a data object.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Column		
Attribute	bytes data_type name_ start_byte	1 1 1 0..1	
Inherited Attribute	none		
Association	has_Alias has_Bit_Column	0..* 0..*	Alias Bit_Column
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.8 COLUMN_CORE

Object Type: Data_Object_Description
Object Description: Derived from data products from the followng data sets VG2-SR/UR/NR-PPS-2/4-OCC-V1.0, EAR-A-5-DDR-ALBEDOS-V1.1, CO-D-CDA-3/4/5-DUST-V1.0, EAR-C-COMPIL-5-COMET-NUC-PROPERTIES-V1.0, NEAR-A-NIS-5-EDR-ALL-PHASES-PDSREV-V1.0, MRO-M-CRISM-2-EDR-V1.0 and others.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Column . . Column_Core		
Attribute	bit_mask column_number description format item_bytes item_offset items maximum minimum missing_constant not_applicable_constant unit valid_maximum valid_minimum	0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1 0..1	
Inherited Attribute	bytes data_type name_ start_byte	1 1 1 0..1	
Association	none		
Inherited Association	has_Alias has_Bit_Column uses_pointer	0..* 0..* 0..1	Alias Bit_Column Data_Object_Pointer

3.9 CONTAINER

Object Type: Data_Object_Description
Object Description: The CONTAINER object is used to group a set of sub-objects (such as COLUMNs) that repeat within a data object (such as a TABLE). Use of the CONTAINER object allows repeating groups to be defined within a data structure.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Container		
Attribute	bytes description name_ repetitions start_byte	1 1 1 1 1	
Inherited Attribute	none		
Association	has_Container_Column	1	Column
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.10 CONTAINER_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data set: NEAR-A-NIS-5-EDR-ALL-PHASES-PDSREV-V1.0 - Note several similar columns were combine.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Container . . Container_Core		
Attribute	column_number	0..1	
Inherited Attribute	bytes description name_ repetitions start_byte	1 1 1 1 1	
Association	none		
Inherited Association	has_Container_Column uses_pointer	1 0..1	Column Data_Object_Pointer

3.11 DATA_OBJECT_DESCRIPTION

Object Type: Data_Object_Description
Object Description: Digital Object Descriptions are object classes that are used to provide descriptions of the data objects in the PDS archive.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description		
Attribute	none		
Inherited Attribute	none		
Association	uses_pointer	0..1	Data_Object_Pointer
Inherited Association	none		

3.12 DOCUMENT

Object Type: Data_Object_Description
Object Description: The DOCUMENT object is used to label a particular document that is provided on a volume to support an archived data product. A document can be made up of one or more files in a single format.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Document		
Attribute	abstract_text description document_format document_name document_topic_type encoding_type interchange_format publicatoin_date	0..1 0..1 1 1 1 0..1 1 1	
Inherited Attribute	none		
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.13 ELEMENT

Object Type: Data_Object_Description
Object Description: The ELEMENT object provides a means of defining a lowest-level component of a data object, and which can be stored in an integral multiple of 8-bit bytes.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Element		
Attribute	bytes data_type name_	1 1 1	
Inherited Attribute	none		
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.14 ELEMENT_CORE

Object Type: Data_Object_Description
Object Description: Derived from data products from the followng data sets DI/EAR-C-KECK1LWS-3-9P-IMAGES-PHOT-V1.0, MEX-Y/M-SPI-2-IREDR-RAWXCRUISE/MARS-V1.0

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Element . . Element_Core		
Attribute	maximum minimum unit	0..1 0..1 0..1	
Inherited Attribute	bytes data_type name_	1 1 1	
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.15 EXPLICIT_FILE

Object Type: Data_Object_Description
Object Description: The Explicit File object is used in attached or detached labels to define the attributes or characteristics of a data file. An Explicit File object is used when a file reference is needed.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . File . . Explicit_File		
Attribute	file_name file_records record_bytes	1 1 1	
Inherited Attribute	record_type	1	FIXED_LENGTH VARIABLE_LENGTH STREAM UNDEFINED
Association	has_File_Data_Object_Desc	0..*	Image Table Spreadsheet Array
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.16 FIELD

Object Type: Data_Object_Description
Object Description: The FIELD object identifies a single variable-width field in a SPREADSHEET object.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Field		
Attribute	bytes data_type name_	1 1 1	
Inherited Attribute	none		
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.17 FIELD_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data set: MEX-M-ASPERA3-2-EDR-NPI-V1.0

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Field . . Field_Core		
Attribute	description field_number format item_bytes items	0..1 0..1 0..1 0..1 0..1	
Inherited Attribute	bytes data_type name_	1 1 1	
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.18 FILE

Object Type: Data_Object_Description
Object Description: The FILE object is used in attached or detached labels to define the attributes or characteristics of a data file.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . File		
Attribute	record_type	1	FIXED_LENGTH VARIABLE_LENGTH STREAM UNDEFINED
Inherited Attribute	none		
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.19 HEADER

Object Type: Data_Object_Description
Object Description: The HEADER object is used to identify and define the attributes of commonly used header data structures such as VICAR or FITS.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Header		
Attribute	bytes header_type	1 1	
Inherited Attribute	none		
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.20 HEADER_FITS_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data sets: DIF-C-HRII-3/4-9P-ENCOUNTER-V1.0, DII-C-ITS-3/4-9P-ENCOUNTER-V1.0, NEAR-A-NIS-5-EDR-ALL-PHASES-PDSREV-V1.0, DI/EAR-C-KECK1LWS-3-9P-IMAGES-PHOT-V1.0

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Header . . Header_FITS_Core		
Attribute	description header_type interchange_format records	0..1 1 1 0..1	 FITS BINARY
Inherited Attribute	bytes	1	
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.21 HEADER_VICAR_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data sets: MEX-M-HRSC-3-RDR-V2.0, MEX-M-HRSC-5-REFDR-MAPPROJECTED-V1.0, VG1/VG2-S-ISS-2/3/4/6-PROCESSED-V1.0 - Note that the Header and Extension Header are combined.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Header . . Header_VICAR_Core		
Attribute	description header_type interchange_format records	0..1 1 1 0..1	 VICR2 ASCII
Inherited Attribute	bytes	1	
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.22 HISTOGRAM

Object Type: Data_Object_Description
Object Description: The HISTOGRAM object is a sequence of numeric values that provides the number of occurrences of a data value or a range of data values in a data object.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Histogram		
Attribute	data_type item_bytes items	1 1 1	
Inherited Attribute	none		
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.23 HISTOGRAM_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data sets: MGN-V-RDRS-5-DIM-V1.0 and VO1/VO2-M-VIS-5-DIM-V2.0

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Histogram . . Histogram_Core		
Attribute	none		
Inherited Attribute	data_type item_bytes items	1 1 1	
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.24 IMAGE

Object Type: Data_Object_Description
Object Description: An IMAGE object is a two-dimensional array of values, all of the same type, each of which is referred to as a sample.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Image		
Attribute	line_samples lines sample_bits sample_type	1 1 1 1	
Inherited Attribute	none		
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.25 IMPLICIT_FILE

Object Type: Data_Object_Description
Object Description: The Implicit File object is used in attached or detached labels to define the attributes or characteristics of a data file. The label for the Implicit File starts at the top of the file containing the label. For an attached label, the file being described is the file containing the label and data. For a detached label, the file being described is the file being pointed to.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . File . . Implicit_File		
Attribute	file_records record_bytes	1 1	
Inherited Attribute	record_type	1	FIXED_LENGTH VARIABLE_LENGTH STREAM UNDEFINED
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.26 IMPLICIT_FILE_ATTACHED

Object Type: Data_Object_Description
Object Description: The Implicit File object is used in attached or detached labels to define the attributes or characteristics of a data file. The label for the Implicit File starts at the top of the file containing the label. For an attached label, the file being described is the file containing the label and data.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . File . . Implicit_File . . . Implicit_File_Attached		
Attribute	label_records	1	
Inherited Attribute	record_type	1	FIXED_LENGTH VARIABLE_LENGTH STREAM UNDEFINED
	file_records record_bytes	1 1	
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.27 PALETTE

Object Type: Data_Object_Description
Object Description: The PALETTE object, a sub-class of the TABLE object, contains entries which represent color table assignments for values (i.e., SAMPLEs) contained in an IMAGE.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Palette		
Attribute	columns interchange_format row_bytes rows	1 1 1 1	
Inherited Attribute	none		
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.28 SERIES

Object Type: Data_Object_Description
Object Description: The SERIES object is a sub-class of the TABLE object. It is used for storing a sequence of measurements organized in a specific way (e.g., chronologically, by radial distance, etc.). The SERIES uses the same physical format specification as the TABLE object with additional sampling parameter information describing the variation between elements in the series.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Series		
Attribute	interchange_format sampling_parameter_interval sampling_parameter_name sampling_parameter_unit	1 1 1 1	ASCII
Inherited Attribute	columns row_bytes rows	1 1 1	
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.29 SERIES_BINARY

Object Type: Data_Object_Description
Object Description: The SERIES object is a sub-class of the TABLE object. It is used for storing a sequence of measurements organized in a specific way (e.g., chronologically, by radial distance, etc.). The SERIES uses the same physical format specification as the TABLE object with additional sampling parameter information describing the variation between

elements in the series.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Series_Binary		
Attribute	interchange_format sampling_parameter_interval sampling_parameter_name sampling_parameter_unit	1 1 1 1	BINARY
Inherited Attribute	columns row_bytes rows	1 1 1	
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.30 SERIES_BINARY_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data set:

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Series_Binary . . . Series_Binary_Core		
Attribute	none		
Inherited Attribute	interchange_format sampling_parameter_interval sampling_parameter_name sampling_parameter_unit columns row_bytes rows	1 1 1 1 1 1 1	BINARY
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.31 SERIES_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following

data set: VG2-SR/UR/NR-PPS-2/4-OCC-V1.0

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Series . . . Series_Core		
Attribute	description maximum_sampling_parameter minimum_sampling_parameter	0..1 0..1 0..1	
Inherited Attribute	interchange_format sampling_parameter_interval sampling_parameter_name sampling_parameter_unit columns row_bytes rows	1 1 1 1 1 1 1	ASCII
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.32 SIMPLE_IMAGE_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data sets: MEX-M-HRSC-3-RDR-V2.0, MEX-M-HRSC-5-REFDR-MAPPROJECTED-V1.0, VO1/VO2-M-VIS-5-DIM-V2.0, MGN-V-RDRS-5-DIM-V1.0, DIF-C-HRII-3/4-9P-ENCOUNTER-V1.0, DII-C-ITS-3/4-9P-ENCOUNTER-V1.0, VG1/VG2-S-ISS-2/3/4/6-PROCESSED-V1.0

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Image . . Simple_Image_Core		
Attribute	axis_order_type	0..1	
	band_storage_type	0..1	
	bands	0..1	
	checksum	0..1	
	horizontal_fov	0..1	
	horizontal_pixel_fov	0..1	
	interchange_format	0..1	
	line_display_direction	0..1	
	maximum	0..1	
	mean	0..1	
	median	0..1	
	minimum	0..1	
	missing_constant	0..1	
	note	0..1	
	offset	0..1	
	reflectance_scaling_factor	0..1	
	sample_bit_mask	0..1	
	sample_display_direction	0..1	
	scaling_factor	0..1	
	standard_deviation	0..1	
	unit	0..1	
	vertical_fov	0..1	
	vertical_pixel_fov	0..1	
Inherited Attribute	line_samples	1	
	lines	1	
	sample_bits	1	
	sample_type	1	
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.33 SOFTWARE

Object Type: Data_Object_Description
Object Description: The SOFTWARE catalog object provides general information about a software tool including description, availability information, and dependencies.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Software		
Attribute	data_format node_id required_storage_bytes software_desc software_id software_license_type software_name software_purpose software_version_id technical_support_type	1 1 1 1 1 1 1 1 1 1	
Inherited Attribute	none		
Association	has_Software_Online	0..*	Software_Online
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.34 SOFTWARE_ONLINE

Object Type: Data_Object_Description
Object Description: The SOFTWARE_ONLINE object, a sub-object of SOFTWARE catalog object, provides identifying information for each PDS node providing access to a particular SOFTWARE object.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Software_Online		
Attribute	node_id on_line_identification on_line_name platform protocol_type	1 1 1 1..* 1	
Inherited Attribute	none		
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.35 SPECTRUM

Object Type: Data_Object_Description
Object Description: The SPECTRUM object is a form of TABLE used for storing spectral measurements. The SPECTRUM object is assumed to have a number of measurements of the observation target taken in different spectral bands.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Spectrum		
Attribute	none		
Inherited Attribute	columns interchange_format row_bytes rows	1 1 1 1	
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.36 SPECTRUM_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data set:

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Spectrum . . . Spectrum_Core		
Attribute	none		
Inherited Attribute	columns interchange_format row_bytes rows	1 1 1 1	
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.37 SPREADSHEET

Object Type: Data_Object_Description
Object Description: The SPREADSHEET is a natural storage format for data products in which the data rows are sparsely populated or field values have variable lengths.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Spreadsheet		
Attribute	field_delimiter fields row_bytes rows	1 1 1 1	
Inherited Attribute	none		
Association	has_Field	1..*	Field
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.38 SPREADSHEET_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data set: MEX-M-ASPERA3-2-EDR-NPI-V1.0.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Spreadsheet . . Spreadsheet_Core		
Attribute	interchange_format name_	0..1 0..1	
Inherited Attribute	field_delimiter fields row_bytes rows	1 1 1 1	
Association	has_Field	1..*	Field_Core
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.39 TABLE

Object Type: Data_Object_Description
Object Description: TABLEs are a natural storage format for collections of data from many instruments. They are often the most effective way of storing much of the meta-data used to identify and describe instrument observations.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table		
Attribute	columns interchange_format row_bytes rows	1 1 1 1	
Inherited Attribute	none		
Association	has_Column	1..*	Column
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.40 TABLE_ASCII

Object Type: Data_Object_Description
Object Description: TABLEs are a natural storage format for collections of data from many instruments. They are often the most effective way of storing much of the meta-data used to identify and describe instrument observations.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Table_ASCII		
Attribute	interchange_format	1	ASCII
Inherited Attribute	columns row_bytes rows	1 1 1	
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.41 TABLE_ASCII_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data sets: EAR-C-COMPIL-5-COMET-NUC-PROPERTIES-V1.0, EAR-A-5-DDR-ALBEDOS-V1.1, CO-D-CDA-3/4/5-DUST-V1.0, SDU-C-NAVCAM-5-WILD2-SHAPE-MODEL-V2.1, GO-J-MAG-3-RDR-HIGHRES-V1.0

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Table_ASCII . . . Table_ASCII_Core		
Attribute	description interchange_format name_	0..1 1 0..1	 ASCII
Inherited Attribute	columns row_bytes rows	1 1 1	
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.42 TABLE_ASCII_KEY_CORE

Object Type: Data_Object_Description
Object Description: Suggested by Mitch Gordon

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Table_ASCII . . . Table_ASCII_Core . . . . Table_ASCII_Key_Core		
Attribute	none		
Inherited Attribute	columns row_bytes rows description interchange_format name_	1 1 1 0..1 1 0..1	 ASCII
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.43 TABLE_BINARY

Object Type: Data_Object_Description
Object Description: TABLEs are a natural storage format for collections of data from many instruments. They are often the most effective way of storing much of the meta-data used to identify and describe instrument observations.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Table_Binary		
Attribute	interchange_format	1	BINARY
Inherited Attribute	columns row_bytes rows	1 1 1	
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.44 TABLE_BINARY_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data set: CO-V/E/J/S-RADAR-3-LBDR-V1.0, MRO-M-CRISM-2-EDR-V1.0, NEAR-A-NIS-5-EDR-ALL-PHASES-PDSREV-V1.0, ODY-M-GRS-4-CGS-V1.0, CO-E/SW/J/S-MAG-2-REDR-RAW-DATA-V1.0, CO-E/J/S/SW-CAPS-2-UNCALIBRATED-V1.0, CO-V/E/J/S/SS-RPWS-2-REFDR-WBRFULL-V1.0, CO-V/E/J/S/SS-RPWS-3-RDR-LRFULL-V1.0, MGS-M-RSS-1-EXT-V1.0,

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Table_Binary . . . Table_Binary_Core		
Attribute	description interchange_format name_ row_suffix_bytes sampling_parameter_interval sampling_parameter_name sampling_parameter_unit	0..1 1 0..1 0..1 0..1 0..1 0..1	 BINARY
Inherited Attribute	columns row_bytes rows	1 1 1	
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.45 TABLE_BINARY_KEYED_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data set: CO-S-CIRS-2/3/4-REFORMATTED-V1.0

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Table_Binary . . . Table_Binary_Keyed_Core		
Attribute	description primary_key start_primary_key stop_primary_key	0..1 1 1 1	
Inherited Attribute	columns row_bytes rows interchange_format	1 1 1 1	 BINARY
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.46 TABLE_KEYED_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data set: CO-S-CIRS-2/3/4-REFORMATTED-V1.0

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Table_Keyed_Core		
Attribute	description interchange_format primary_key start_primary_key stop_primary_key	0..1 1 1 1 1	 ASCII
Inherited Attribute	columns row_bytes rows	1 1 1	
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.47 TEXT

Object Type: Data_Object_Description
Object Description: The TEXT object describes a file which contains plain text. It is most often used in an attached label, so that the text begins immediately after the END statement of the label.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Text		
Attribute	note publication_date	1 1	
Inherited Attribute	none		
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.48 TEXT_CORE

Object Type: Data_Object_Description
Object Description: The TEXT object describes a file which contains plain text. It is most often used in an attached label, so that the text begins immediately after the END statement of the label.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Text . . Text_Core		
Attribute	none		
Inherited Attribute	note publication_date	1 1	
Association	none		
Inherited Association	uses_pointer	0..1	Data_Object_Pointer

3.49 TIME_SERIES

Object Type: Data_Object_Description
Object Description: The object name TIME_SERIES? is used when the series is chronological. In this case the label keywords START_TIME and STOP_TIME are assumed to indicate the minimum and maximum times in the file. If this is not the case, the MINIMUM_SAMPLING_PARAMETER and MAXIMUM_SAMPLING_PARAMETER keywords should be used to specify the corresponding time values for the series.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Series . . . Time_Series		
Attribute	start_time stop_time	1 1	
Inherited Attribute	interchange_format sampling_parameter_interval sampling_parameter_name sampling_parameter_unit columns row_bytes rows	1 1 1 1 1 1 1	ASCII
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.50 TIME_SERIES_BINARY

Object Type: Data_Object_Description
Object Description: The object name TIME_SERIES? is used when the series is chronological. In this case the label keywords START_TIME and STOP_TIME are assumed to indicate the minimum and maximum times in the file. If this is not the case, the MINIMUM_SAMPLING_PARAMETER and MAXIMUM_SAMPLING_PARAMETER keywords should be used to specify the corresponding time values for the series.

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Series_Binary . . . Time_Series_Binary		
Attribute	start_time stop_time	1 1	
Inherited Attribute	interchange_format sampling_parameter_interval sampling_parameter_name sampling_parameter_unit columns row_bytes rows	1 1 1 1 1 1 1	BINARY
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.51 TIME_SERIES_BINARY_CORE

Object Type: Data_Object_Description
Object Description: Modeled from data products from the following data set: ODY-M-GRS-4-CGS-V1.0, CO-V/E/J/S/SS-RPWS-2-REFDR-WBRFULL-V1.0

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Series_Binary . . . Time_Series_Binary . . . . Time_Series_Binary_Core		
Attribute	none		
Inherited Attribute	interchange_format sampling_parameter_interval sampling_parameter_name sampling_parameter_unit columns row_bytes rows start_time stop_time	1 1 1 1 1 1 1 1 1	BINARY
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

3.52 TIME_SERIES_CORE

Object Type: Data_Object_Description
Object Description: Derived from data products from the followng data sets

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Description . Table . . Series . . . Time_Series . . . . Time_Series_Core		
Attribute	description name_ row_prefix_bytes	0..1 0..1 0..1	
Inherited Attribute	interchange_format sampling_parameter_interval sampling_parameter_name sampling_parameter_unit columns row_bytes rows start_time stop_time	1 1 1 1 1 1 1 1 1	ASCII
Association	none		
Inherited Association	uses_pointer has_Column	0..1 1..*	Data_Object_Pointer Column

4 Label Object Classes

This section provides object classes for defining a generic data product label and includes data product identification, description, and ancillary object classes, and associations with data format object classes.

The data product label object class hierarchy is illustrated in the following diagram. This diagram presents the subclassOf relation for each object class in a hierarchical (tree) format, providing a visual representation of the object classes in relation to their parent classes.

```
. Data_Object
. Data_Object_Pointer
. . Data_Object_Pointer_Name_and_Offset
. . Data_Object_Pointer_Offset_Only
. Descriptive_Data_Elements
. Identification_Data_Elements
. Label_Standards_Identifiers
. Labeled_Data_Object
. . Labeled_Array
. . Labeled_Document
. . Labeled_File
. . . Labeled_File_Explicit
. . . Labeled_File_Implicit
. . . . Labeled_File_Implicit_Attached
. . Labeled_Header
. . . Labeled_Header_FITS
. . . Labeled_Header_VICR
. . Labeled_Histogram
. . Labeled_Image
. . Labeled_Palette
. . Labeled_Software
. . Labeled_Spreadsheet
. . Labeled_Table
. . . Labeled_Index_Table
. . . Labeled_Series
. . . . Labeled_Time_Series
. . . Labeled_Spectrum
. . . Labeled_Table_ASCII
. . . Labeled_Table_Binary
. . Labeled_Text
```

The class hierarchy above includes 30 unique classes.

The data product label object classes are illustrated using a Unified

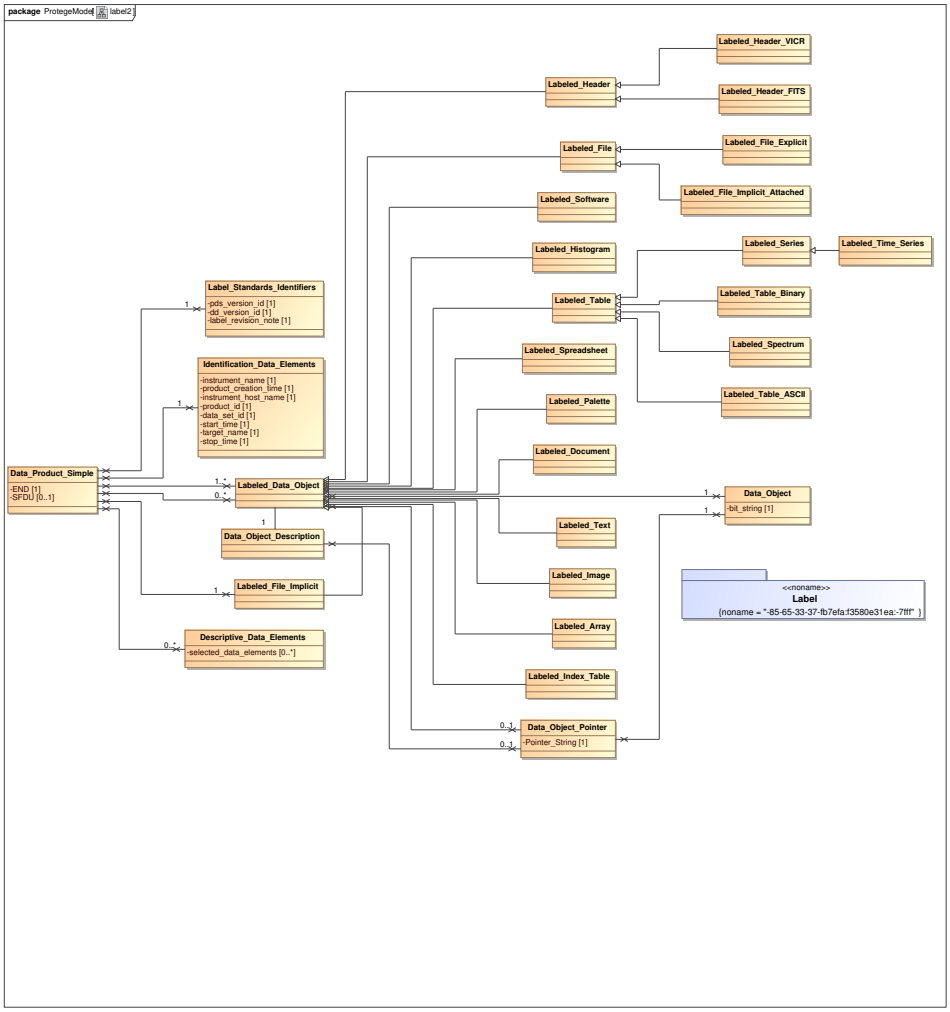

Figure 8: Label UML Class Diagram

Modeling Language (UML) class hierarchy diagram in Figure 8. This diagram defines the object classes that are used to describe the composition of a data product label. The following sections present the data product label object classes in a table format. The table includes the class hierarchy, class attributes, and class associations. The class attributes and associations listed include both those used to define the object class and those inherited from parent classes. Cardinalities are provided where appropriate.

4.1 DATA_OBJECT

Object Type: Data_Object
Object Description: A sequence of digital bits.

Relationship	Entity	Card	Value
Hierarchy	Data_Object		
Attribute	bit_string	1	
Inherited Attribute	none		
Association	none		
Inherited Association	none		

4.2 DATA_OBJECT_POINTER

Object Type: Data_Object_Pointer
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Pointer		
Attribute	Pointer_String	1	
Inherited Attribute	none		
Association	points_to	1	Data_Object
Inherited Association	none		

4.3 DATA_OBJECT_POINTER_NAME_AND_OFFSET

Object Type: Data_Object_Pointer
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Pointer . Data_Object_Pointer_Name_and_Offset		
Attribute	Pointer_String	1	File_Name_and_Record_
Inherited Attribute	none		
Association	none		
Inherited Association	points_to	1	Data_Object

4.4 DATA_OBJECT_POINTER_OFFSET_ONLY

Object Type: Data_Object_Pointer
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Object_Pointer . Data_Object_Pointer_Offset_Only		
Attribute	Pointer_String	1	Record_or_Byte_Offset
Inherited Attribute	none		
Association	none		
Inherited Association	points_to	1	Data_Object

4.5 DESCRIPTIVE_DATA_ELEMENTS

Object Type: Descriptive_Data_Elements
Object Description: IIn addition to the data identification elements required for various types of data, PDS strongly recommends including additional data elements related to specific types of data. These descriptive elements should include any elements needed to interpret or process the data objects or which would be needed to catalog the data product to support potential search criteria at the product level.

Relationship	**Entity**	**Card**	**Value**
Hierarchy	Descriptive_Data_Elements		
Attribute	selected_data_elements	0..*	
Inherited Attribute	none		
Association	none		
Inherited Association	none		

4.6 IDENTIFICATION_DATA_ELEMENTS

Object Type: Identification_Data_Elements
Object Description: The data identification elements provide additional information about a data product that can be used to relate the product to other data products from the same data set or data set collection.

Relationship	**Entity**	**Card**	**Value**
Hierarchy	Identification_Data_Elements		
Attribute	data_set_id	1	
	instrument_host_name	1	
	instrument_name	1	
	product_creation_time	1	
	product_id	1	
	start_time	1	
	stop_time	1	
	target_name	1	
Inherited Attribute	none		
Association	collected_about	0..*	Target
	collected_by	0..*	Instrument
	collected_in	1..*	Data_Set
Inherited Association	none		

4.7 LABEL_STANDARDS_IDENTIFIERS

Object Type: Label_Standards_Identifiers
Object Description: Each PDS label must begin with the

PDS_VERSION_ID data element. This element identifies the published version of the Standards to which the label adheres, for purposes of both validation as well as software development and support. For labels adhering to the standards described in this document the appropriate value is PDS3. The DD_VERSION_ID element identifies the version of the PDS Data Dictionary to which a label complies. Current PDS practice is to identify a Data Dictionary version with the identifier used for the PDS catalog build in which it resides, e.g., pdscat1r47, pdscat1r48, and so on. This keyword will use the upper case representation of the catalog identifier, e.g., PDSCAT1R47, PDSCAT1R48, etc. The LABEL_REVISION_NOTE element is a free form, unlimited-length character string providing information regarding the revision status and authorship of a PDS label. It should include at least the latest revision date and the author of the current version, but may include a complete editing history.

Relationship	**Entity**	**Card**	**Value**
Hierarchy	Label_Standards_Identifiers		
Attribute	dd_version_id	1	
	label_revision_note	1	
	pds_version_id	1	PDS3
Inherited Attribute	none		
Association	none		
Inherited Association	none		

4.8 LABELED_ARRAY

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	**Entity**	**Card**	**Value**
Hierarchy	Labeled_Data_Object . Labeled_Array		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Array
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.9 LABELED_DATA_OBJECT

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in

association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object has_Data_Object_Description has_Data_Object_Pointer	1 1 0..1	Data_Object Data_Object_Pointer
Inherited Association	none		

4.10 LABELED_DOCUMENT

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Document		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description has_Data_Object_Pointer	1 None	Document Data_Object_Pointer
Inherited Association	has_Data_Object	1	Data_Object

4.11 LABELED_FILE

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_File		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description has_Data_Object_Pointer	1 None	File Data_Object_Pointer
Inherited Association	has_Data_Object	1	Data_Object

4.12 LABELED_FILE_EXPLICIT

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_File . . Labeled_File_Explicit		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Explicit_File
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 None	Data_Object Data_Object_Pointer

4.13 LABELED_FILE_IMPLICIT

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_File . . Labeled_File_Implicit		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description has_Data_Object_Pointer	1 None	Implicit_File Data_Object_Pointer
Inherited Association	has_Data_Object	1	Data_Object

4.14 LABELED_FILE_IMPLICIT_ATTACHED

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_File . . Labeled_File_Implicit . . . Labeled_File_Implicit_Attached		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Implicit_File_Attached
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 None	Data_Object Data_Object_Pointer

4.15 LABELED_HEADER

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Header		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Header
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.16 LABELED_HEADER_FITS

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Header . . Labeled_Header_FITS		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Header_FITS_Core
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.17 LABELED_HEADER_VICR

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	**Entity**	**Card**	**Value**
Hierarchy	Labeled_Data_Object . Labeled_Header . . Labeled_Header_VICR		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Header_VICAR_Core
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.18 LABELED_HISTOGRAM

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	**Entity**	**Card**	**Value**
Hierarchy	Labeled_Data_Object . Labeled_Histogram		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Histogram
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.19 LABELED_IMAGE

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Image		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Image
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.20 LABELED_INDEX_TABLE

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Table . . Labeled_Index_Table		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Index_Table_Generic
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.21 LABELED_PALETTE

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Palette		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Palette
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.22 LABELED_SERIES

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Table . . Labeled_Series		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Series
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.23 LABELED_SOFTWARE

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Software		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Software
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.24 LABELED_SPECTRUM

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Table . . Labeled_Spectrum		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Spectrum
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.25 LABELED_SPREADSHEET

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Spreadsheet		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Spreadsheet
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.26 LABELED_TABLE

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Table		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Table
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.27 LABELED_TABLE_ASCII

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Table . . Labeled_Table_ASCII		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Table_ASCII
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.28 LABELED_TABLE_BINARY

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Table . . Labeled_Table_Binary		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Table_Binary
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

4.29 LABELED_TEXT

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Text		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description has_Data_Object_Pointer	1 None	Text Data_Object_Pointer
Inherited Association	has_Data_Object	1	Data_Object

4.30 LABELED_TIME_SERIES

Object Type: Labeled_Data_Object
Object Description: A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Relationship	Entity	Card	Value
Hierarchy	Labeled_Data_Object . Labeled_Table . . Labeled_Series . . . Labeled_Time_Series		
Attribute	none		
Inherited Attribute	none		
Association	has_Data_Object_Description	1	Time_Series
Inherited Association	has_Data_Object has_Data_Object_Pointer	1 0..1	Data_Object Data_Object_Pointer

5 Data Product Object Classes

This section provides a draft set of object classes for data products. It uses the data product label classes and defines a set of data product object classes based on the proposed core data formats. Since each data product class is dependent on the data format object class used and since the core data format object classes are not yet approved, only the upper level of this object class hierarchy has been modeled.

The data product object class hierarchy is illustrated in the following diagram. This diagram presents the subclassOf relation for each object class in a hierarchical (tree) format, providing a visual representation of the object classes in relation to their parent classes.

```
. Data_Product
. . Combined_Detached_Label
. . Data_Product_Simple
. . . Data_Product_Attached_Label
. . . . Data_Product_Image_Attached
. . . . . Data_Product_Image_Mapped
. . . Data_Product_Detached_Label
. . . . Data_Product_Array
. . . . . Data_Product_Array_FITS
. . . . Data_Product_Image
. . . . . Data_Product_Image_FITS
. . . . . Data_Product_Image_VICR
. . . . Data_Product_Series_ASCII
. . . . Data_Product_Series_Binary
. . . . Data_Product_Spreadsheet
. . . . Data_Product_Table_ASCII
. . . . Data_Product_Table_Binary
. . . . . Data_Product_Table_FITS_Binary
```

The class hierarchy above includes 18 unique classes.

The data product object classes are illustrated using a Unified Modeling Language (UML) Class Hierarchy diagram in Figure 9. This diagram defines the object classes that comprise a data product. The following sections present the data product object classes in a table format. The table includes the class hierarchy, class attributes, and class associations. The class attributes and associations listed include both those used to define the object class and those inherited from parent classes. Cardinalities are provided where appropriate.

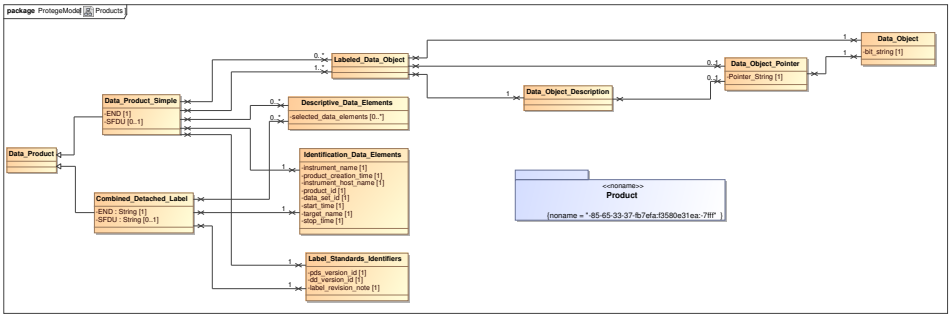

Figure 9: Data Product UML Class Diagram

5.1 COMBINED_DETACHED_LABEL

Object Type: Data_Product
Object Description: A single PDS detached data product label file is used to describe the contents of more than one data product file. The combined detached label contains pointers to individual data products.

Relationship	**Entity**	**Card**	**Value**
Hierarchy	Data_Product . Combined_Detached_Label		
Attribute	END SFDU	1 0..1	END SFDU
Inherited Attribute	none		
Association	has_DDE has_IDE has_LSI has_Labeled_Explicit_File_Object	0..* 1 1 1..*	Descriptive_Data_Elements Identification_Data_Elements Label_Standards_Identifiers Labeled_File_Explicit
Inherited Association	none		

5.2 DATA_PRODUCT

Object Type: Data_Product
Object Description: At its simplest, a data product consists of a PDS label and the data object that it describes. More complex data products may contain several mutually dependent data objects, a primary object and one or more secondary objects, or both. In all cases, a single label is used to describe all parts of the product (even if they are held in separate physical files). A single PRODUCT_ID value is defined for the entire set in that PDS label. [StdRef Chap 4) - An entity consisting of a science data object, metadata, and ancillary files and that is orderable.

Relationship	Entity	Card	Value
Hierarchy	Data_Product		
Attribute	none		
Inherited Attribute	none		
Association	none		
Inherited Association	none		

5.3 DATA_PRODUCT_ARRAY

Object Type: Data_Product
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Detached_Label . . . Data_Product_Array		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_Primary_LDO has_Secondary_LDO	1..* 0..*	Labeled_Array Labeled_Header
Inherited Association	has_AOD has_DDE has_IDE has_ILF has_LSI	0..* 0..* 1 1 1	Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Labeled_File_Implicit Label_Standards_Identifiers

5.4 DATA_PRODUCT_ARRAY_FITS

Object Type: Data_Product
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Detached_Label . . . Data_Product_Array . . . . Data_Product_Array_FITS		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_Secondary_LDO	0..*	Labeled_Header_FITS
Inherited Association	has_Primary_LDO has_AOD has_DDE has_IDE has_ILF has_LSI	1..* 0..* 0..* 1 1 1	Labeled_Array Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Labeled_File_Implicit Label_Standards_Identifiers

5.5 DATA_PRODUCT_ATTACHED_LABEL

Object Type: Data_Product
Object Description: The PDS data product label is attached at the beginning of the data product file. There is one label attached to each data product file.

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Attached_Label		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_ILF	1	Labeled_File_Implicit_Attached
Inherited Association	has_AOD has_DDE has_IDE has_LSI has_Primary_LDO has_Secondary_LDO	0..* 0..* 1 1 1..* 0..*	Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Label_Standards_Identifiers Labeled_Time_Series Labeled_Table_Binary Labeled_Spectrum Labeled_Image Labeled_Series Labeled_Table_ASCII Labeled_Table Labeled_Histogram Labeled_Palette Labeled_Header Labeled_Data_Object

5.6 DATA_PRODUCT_DETACHED_LABEL

Object Type: Data_Product
Object Description: The PDS data product label is detached from the data and resides in a separate file which contains a pointer to the data product file. There is one detached label file for every data product file. The label file should have the same base name as its associated data file, but the extension .LBL .

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Detached_Label		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	none		
Inherited Association	has_AOD has_DDE has_IDE has_ILF has_LSI has_Primary_LDO has_Secondary_LDO 	0..* 0..* 1 1 1 1..* 0..* 	Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Labeled_File_Implicit Label_Standards_Identifiers Labeled_Time_Series Labeled_Table_Binary Labeled_Spectrum Labeled_Image Labeled_Series Labeled_Table_ASCII Labeled_Table Labeled_Histogram Labeled_Palette Labeled_Header Labeled_Data_Object

5.7 DATA_PRODUCT_IMAGE

Object Type: Data_Product
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Detached_Label . . . Data_Product_Image		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_Primary_LDO has_Secondary_LDO	1..* 0..*	Labeled_Image Labeled_Header Labeled_Histogram
Inherited Association	has_AOD has_DDE has_IDE has_ILF has_LSI	0..* 0..* 1 1 1	Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Labeled_File_Implicit Label_Standards_Identifiers

5.8 DATA_PRODUCT_IMAGE_ATTACHED

Object Type: Data_Product
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Attached_Label . . . Data_Product_Image_Attached		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_ILF has_Primary_LDO has_Secondary_LDO	1 1..* 0..*	Labeled_File_Implicit_Attach Labeled_Image Labeled_Histogram Labeled_Header
Inherited Association	has_AOD has_DDE has_IDE has_LSI	0..* 0..* 1 1	Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Label_Standards_Identifiers

5.9 DATA_PRODUCT_IMAGE_FITS

Object Type: Data_Product
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Detached_Label . . . Data_Product_Image . . . . Data_Product_Image_FITS		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_Secondary_LDO	0..*	Labeled_Header_FITS
Inherited Association	has_Primary_LDO has_AOD has_DDE has_IDE has_ILF has_LSI	1..* 0..* 0..* 1 1 1	Labeled_Image Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Labeled_File_Implicit Label_Standards_Identifiers

5.10 DATA_PRODUCT_IMAGE_MAPPED

Object Type: Data_Product
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Attached_Label . . . Data_Product_Image_Attached . . . . Data_Product_Image_Mapped		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_Image_Map_Projection	1	Image_Map_Projection_Obje
Inherited Association	has_ILF has_Primary_LDO has_Secondary_LDO has_AOD has_DDE has_IDE has_LSI	1 1..* 0..* 0..* 0..* 1 1	Labeled_File_Implicit_Attach Labeled_Image Labeled_Histogram Labeled_Header Ancillary_Object_Descriptio Descriptive_Data_Elements Identification_Data_Element Label_Standards_Identifiers

5.11 DATA_PRODUCT_IMAGE_VICR

Object Type: Data_Product
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Detached_Label . . . Data_Product_Image . . . . Data_Product_Image_VICR		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_Secondary_LDO	0..*	Labeled_Histogram Labeled_Header_VICR
Inherited Association	has_Primary_LDO has_AOD has_DDE has_IDE has_ILF has_LSI	1..* 0..* 0..* 1 1 1	Labeled_Image Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Labeled_File_Implicit Label_Standards_Identifiers

5.12 DATA_PRODUCT_SERIES_ASCII

Object Type: Data_Product
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Detached_Label . . . Data_Product_Series_ASCII		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_Primary_LDO has_Secondary_LDO	1..* 0..*	Labeled_Series Labeled_Header
Inherited Association	has_AOD has_DDE has_IDE has_ILF has_LSI	0..* 0..* 1 1 1	Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Labeled_File_Implicit Label_Standards_Identifiers

5.13 DATA_PRODUCT_SERIES_BINARY

Object Type: Data_Product
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Detached_Label . . . Data_Product_Series_Binary		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_Primary_LDO has_Secondary_LDO	1..* 0..*	Labeled_Series Labeled_Header
Inherited Association	has_AOD has_DDE has_IDE has_ILF has_LSI	0..* 0..* 1 1 1	Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Labeled_File_Implicit Label_Standards_Identifiers

5.14 DATA_PRODUCT_SIMPLE

Object Type: Data_Product
Object Description: At its simplest, a data product consists of a PDS label and the data object that it describes. More complex data products may contain several mutually dependent data objects, a primary object and one or more secondary objects, or both. In all cases, a single label is used to describe all parts of the product (even if they are held in separate physical files). A single PRODUCT_ID value is defined for the entire set in that PDS label. [StdRef Chap 4) - An entity consisting of a science data object, metadata, and ancillary files and that is orderable.

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple		
Attribute	END SFDU	1 0..1	END SFDU
Inherited Attribute	none		
Association	has_AOD has_DDE has_IDE has_ILF has_LSI has_Primary_LDO has_Secondary_LDO 	0..* 0..* 1 1 1 1..* 0..* 	Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Labeled_File_Implicit Label_Standards_Identifiers Labeled_Time_Series Labeled_Table_Binary Labeled_Spectrum Labeled_Image Labeled_Series Labeled_Table_ASCII Labeled_Table Labeled_Histogram Labeled_Palette Labeled_Header Labeled_Data_Object
Inherited Association	none		

5.15 DATA_PRODUCT_SPREADSHEET

Object Type: Data_Product
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Detached_Label . . . Data_Product_Spreadsheet		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_Primary_LDO	1..*	Labeled_Spreadsheet
Inherited Association	has_AOD has_DDE has_IDE has_ILF has_LSI has_Secondary_LDO	0..* 0..* 1 1 1 0..*	Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Labeled_File_Implicit Label_Standards_Identifiers Labeled_Histogram Labeled_Palette Labeled_Header Labeled_Data_Object

5.16 DATA_PRODUCT_TABLE_ASCII

Object Type: Data_Product
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Detached_Label . . . Data_Product_Table_ASCII		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_Primary_LDO has_Secondary_LDO	1..* 0..*	Labeled_Table_ASCII Labeled_Header
Inherited Association	has_AOD has_DDE has_IDE has_ILF has_LSI	0..* 0..* 1 1 1	Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Labeled_File_Implicit Label_Standards_Identifiers

5.17 DATA_PRODUCT_TABLE_BINARY

Object Type: Data_Product
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Detached_Label . . . Data_Product_Table_Binary		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_Primary_LDO has_Secondary_LDO	1..* 0..*	Labeled_Table_Binary Labeled_Header
Inherited Association	has_AOD has_DDE has_IDE has_ILF has_LSI	0..* 0..* 1 1 1	Ancillary_Object_Description Descriptive_Data_Elements Identification_Data_Elements Labeled_File_Implicit Label_Standards_Identifiers

5.18 DATA_PRODUCT_TABLE_FITS_BINARY

Object Type: Data_Product
Object Description: TBD description

Relationship	Entity	Card	Value
Hierarchy	Data_Product . Data_Product_Simple . . Data_Product_Detached_Label . . . Data_Product_Table_Binary . . . . Data_Product_Table_FITS_Binary		
Attribute	none		
Inherited Attribute	END SFDU	1 0..1	END SFDU
Association	has_Secondary_LDO	0..*	Labeled_Header_FITS
Inherited Association	has_AOD has_DDE has_IDE has_ILF has_LSI has_Primary_LDO	0..* 0..* 1 1 1 1..*	Ancillary_Object_Descri Descriptive_Data_Eleme Identification_Data_Ele Labeled_File_Implicit Label_Standards_Identif Labeled_Table_Binary

6 Ancillary Object Classes

This section provides the data product object classes to be used to provide ancillary information about data products. The table includes the class hierarchy, class attributes, and association relations. The class attributes and associations include both those used to define the object class and those inherited from parent classes. Cardinalities are provided where appropriate.

6.1 ANCILLARY_OBJECT_DESCRIPTION

Object Type: Ancillary_Object_Description
Object Description: Ancillary Object Descriptions are object classes that are used to provide descriptions of objects other than data objects. These objects may be physical or conceptual and often help to interpret a data object.

Relationship	**Entity**	**Card**	**Value**
Hierarchy	Ancillary_Object_Description		
Attribute	none		
Inherited Attribute	none		
Association	none		
Inherited Association	none		

6.2 DATA_SET_MAP_PROJECTION

Object Type: Ancillary_Object_Description
Object Description: The IMAGE_MAP_PROJECTION object is one of two distinct objects that define the map projection used in creating the digital images in a PDS data set. The name of the other associated object that completes the definition is DATA_SET_MAP_PROJECTION (see Appendix B.8). The map projection information resides in these two objects, essentially to reduce data redundancy and at the same time allow the in incllusion of elements needed to process the data at the image level. Basically, static information that is applicable to the complete data set reside in the DATA_SET_MAP_PROJECTION object, while dynamic information that is applicable to the individual images reside in the IMAGE_MAP_PROJECTION object.

Relationship	Entity	Card	Value
Hierarchy	Ancillary_Object_Description . Data_Set_Map_Projection		
Attribute	map_projection_desc map_projection_type rotational_element_desc	1 1 1	
Inherited Attribute	none		
Association	none		
Inherited Association	none		

6.3 IMAGE_MAP_PROJECTION_OBJECT

Object Type: Ancillary_Object_Description
Object Description: The IMAGE_MAP_PROJECTION object is one of two distinct objects that define the map projection used in creating the digital images in a PDS data set. The name of the other associated object that completes the definition is DATA_SET_MAP_PROJECTION (see Appendix B.8). The map projection information resides in these two objects, essentially to reduce data redundancy and at the same time allow the in incllusion of elements needed to process the data at the image level. Basically, static information that is applicable to the complete data set reside in the DATA_SET_MAP_PROJECTION object, while dynamic information that is applicable to the individual images reside in the IMAGE_MAP_PROJECTION object.

Relationship	**Entity**	**Card**	**Value**
Hierarchy	Ancillary_Object_Description . Image_Map_Projection_Object		
Attribute	a_axis_radius	1	
	b_axis_radius	1	
	c_axis_radius	1	
	center_latitude	1	
	center_longitude	1	
	coordinate_system_name	1	
	coordinate_system_type	1	
	eastern_most_longitude	1	
	first_standard_parallel	1	
	line_first_pixel	1	
	line_last_pixel	1	
	line_projection_offset	1	
	map_projection_rotation	1	
	map_projection_type	1	
	map_resolution	1	
	map_scale	1	
	maximum_latitude	1	
	minimum_latitude	1	
	positive_longitude_direction	1	
	reference_latitude	1	
	reference_longitude	1	
	sample_first_pixel	1	
	sample_last_pixel	1	
	sample_projection_offset	1	
	second_standard_parallel	1	
	western_most_longitude	1	
Inherited Attribute	none		
Association	has_DS_Map_Proj	1	Data_Set_Map_Projection
Inherited Association	none		

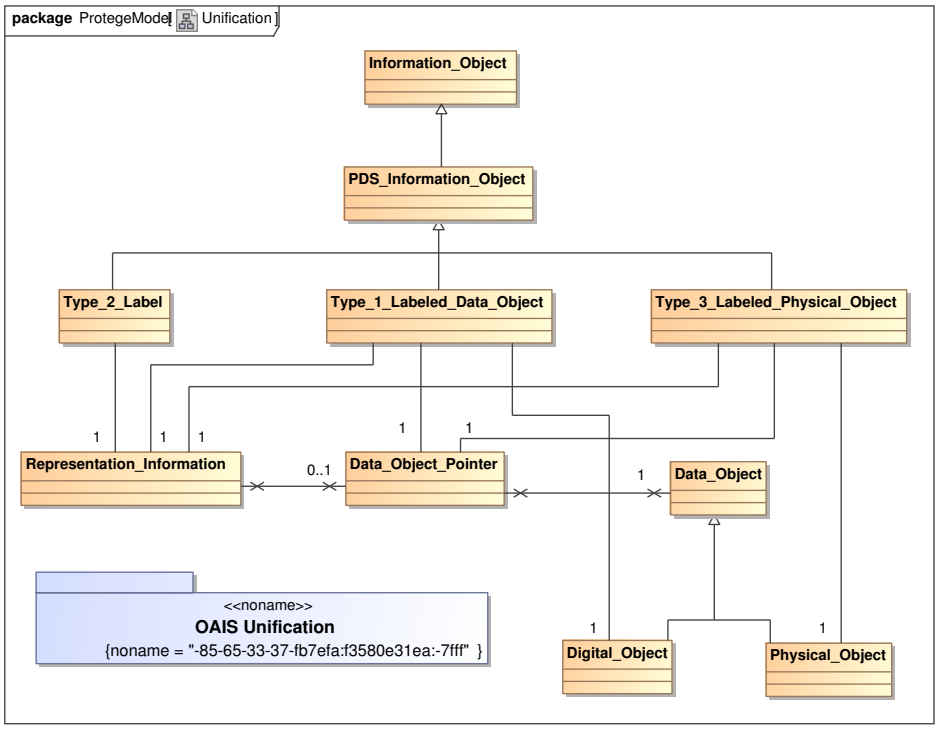

Figure 10: PDS4 Unification with OAIS Reference Model

7 Unification

This section maps the PDS concepts of Labeled_Data_Object and Upper_Level_Object_Description to the Open Archival Information System (OAIS) Reference Models Information Object. The OAIS Information Object has been extended to define a PDS_Information_Object that includes a pointer class. This class in turn is extended to define a Labeled_Data_Object class with a required Pointer and a required Data_Object, a simple Label with only metadata, and a Labeled_Physical_Object.

8 Data Dictionary

The primary purpose of the Data Dictionary is to allow members of the planetary science community to benefit from standards work done in the area of data product description. The work that supports it was originally done at the Jet Propulsion Laboratory by individuals who participate in U.S. and international standards efforts. As a result this data dictionary should serve as a guide to other data systems still in development, or to science data archives that wish to interoperate.

a_axis_radius The a_axis_radius element provides the value of the semimajor axis of the ellipsoid that defines the approximate shape of a target body. 'A' is usually in the equitorial plane.

abstract_desc The ABSTRACT_DESC contains an abstract for the product or DATA_SET_INFORMATION object in which it appears. It provides a string that may be used to provide an abstract for the product (data set) in a publication.

abstract_text The abstract_text element provides a free-form, unlimited-length character string that gives a brief summary of a labeled document, differing from DESCRIPTION in that the text could be extracted for use in a bibliographic context.

address_text The address_text data element provides an unlimited-length, formatted mailing address for an individual or institution.

alias_name The alias_name element provides an alternative term or identifier for a data element or object. Note: In the PDS, values for alias_name are accepted as input to the data system, but automatically changed into the approved term to which they relate.

alternate_telephone_number The alternate_telephone_number data element provides an alternate telephone number for an individual or node. (Includes the area code.)

archive_status The archive_status element provides the status of a data set that has been submitted for inclusion into the PDS archive. If a data set has been partially archived, the archive_status should be ACCUMULATING (e.g., this situation typically occurs when a data set is being produced over a period of time where portions of the data set may be archived, in lien resolution, in peer-review, and under construction). The archive_status_note element is available to describe the archive_status value in finer detail. STANDARD VALUES IN QUEUE - Received at the curation node but no action has been taken by the curation node. Use with caution. PRE PEER REVIEW - Being prepared for peer review under the direction of the curation node. Use with caution IN PEER REVIEW - Under peer review at the curation node but evaluation is not complete. Use with caution IN LIEN RESOLUTION - Peer review completed. Liens are in the process of being resolved. LOCALLY ARCHIVED - Passed peer reviewed with all liens resolved. Considered archived by the curation node but awaiting completion of the standard archiving process. Possible TBD items include the arrival of the archive volume at NSSDC and ingestion of catalog information into the Data Set Catalog. ARCHIVED - Passed peer review with all liens resolved. Available through the Data Set Catalog

and at NSSDC. SUPERSEDED - Superseded by a new version of the data set. This implies that the data set is not to be used unless the requester has specific reasons. When a data set has been superseded the CN will notify NSSDC that their databases need to be updated to advise users of the new status and the location of the replacement data set. SAFED - Received by the PDS with no evaluation. Data will not be formally archived. ACCUMULATING - Portions, but not all, of a data set are in one or more phases of completion (e.g., portions of a data set have been archived while portions remain in lien resolution). Note: If a data set crosses multiple phases of completion, select the highest status level and use the modifier ACCUMULATING. The status is, for example, ARCHIVED-ACCUMULATING, meaning that part of the data set has been archived, but there remains portions of the data set in process. The ARCHIVE_STATUS_NOTE keyword can be used to provide more information. ACCUMULATING value may be used as a modifier to any of the above valid values (e.g., 'ACCUMULATING ARCHIVED', 'ACCUMULATING IN PEER REIVEW').

axes The axes element identifies the number of axes or dimensions of an array or qube data object.

axis_items The axis_items element provides the dimension(s) of the axes of an array data object. For arrays with more than 1 dimension, this element provides a sequence of values corresponding to the number of axes specified. The rightmost item in the sequence corresponds to the most rapidly varying axis, by default.

axis_name The axis_name element provides the sequence of axis names of a qube or array data object, and identifies the order in which the axes are stored in the object. By default, the first axis name in the sequence identifies the array dimension that varies the slowest, followed by the next slowest, and continuing so the rightmost axis named varies the fastest. The number of names specified must be equal to the value of the axes element. Note: For ISIS qube data objects, the most frequently varying axis is listed first, or leftmost, in the sequence.

axis_order_type The AXIS_ORDER_TYPE element is used to identify the storage order for elements of a multidimensional ARRAY object. The default storage order for an ARRAY object presumes the rightmost or last index of a sequence varies the fastest. This is the ordering used in the C programming language and is equivilant to ROW_MAJOR storage order for COLUMN elements within tables. Specifying an AXIS_ORDER_TYPE of FIRST_INDEX_FASTEST may be used for ARRAYs that must be labelled and referenced in the reverse, and is the ordering used in the Fortran programming language.

b_axis_radius The b_axis_radius element provides the value of the intermediate axis of the ellipsoid that defines the approximate shape of a target body. 'B' is usually in the equatorial plane.

band_name BAND_NAME is the name given to a single band in a multi-band image or image qube. If the band is a spectral band, BAND_NAME refers to the associated spectral range; for example, RED, GREEN, BLUE, 415nm, 750nm, 900nm. Examples of names of non-spectral bands are 'Phase angle', 'Thermal inertia', 'Bolometric albedo', 'Latitude', 'Elevation in meters relative to MOLA'.

band_storage_type The band_storage_type element indicates the storage sequence of lines, samples and bands in an image. The values describe, for example, how different samples are interleaved in image lines, or how samples from different bands are arranged sequentially. Example values: BAND SEQUENTIAL, SAMPLE INTERLEAVED, LINE INTERLEAVED.

bands The BANDS element indicates the number of bands in an image or other object.

bit_data_type The bit_data_type element provides the data type for data values stored in the BIT_COLUMN or BIT_ELEMENT object. See also: data_type.

bit_mask The bit_mask element is a series of binary digits identifying the active bits in a value. This is determined by applying a bitwise AND (&) operation between the value and the bit_mask. For example, specifiying a BIT_MASK = 2#11110000# within a 1 byte unsigned integer COLUMN or ELEMENT object would identif only the high-order 4 bits to be used for the value of the object. If other data elements are included in the object description that may be dependent on a bit_mask operation (e.g. DERIVED_MINIMUM, DERIVED_MAXIMUM, INVALID), the rule is to apply the bit_mask first, and then apply or interpret the data with the other values. Byte swapping, if required, should be performed prior to applying the bit_mask.

bits The bits element identifies the count of bits, or units of binary information, in a data representation.

bytes The bytes element indicates the number of bytes allocated for a particular data representation. When BYTES describes an object with variable length (e.g., FIELD), BYTES gives the maximum number of bytes allowed.

c_axis_radius The c_axis_radius element provides the value of the semiminor axis of the ellipsoid that defines the approximate shape of a target body. 'C' is normal to the plane defined by 'A' and 'B'.

center_latitude The center_latitude element provides a reference latitude for certain map projections. For example, in an Orthographic projection, the center_latitude along with the center_longitude defines the point or tangency between the sphere of the planet and the plane of the projection. The map_scale (or map_resolution) is typically defined at the center_latitude and center_longitude. In unprojected images, center_latitude represents the latitude at the center of the image frame.

center_longitude The center_longitude element provides a reference longitude for certain map projections. For example, in an Orthographic projection, the center_longitude along with the center_latitude defines the point or tangency between the sphere of the planet and the plane of the projection. The map_scale (or map_resolution) is typically defined at the center_latitude and center_longitude. In unprojected images, center_longitude represents the longitude at the center of the image frame.

checksum The checksum element represents an unsigned 32-bit sum of all data values in a data object.

citation_desc The CITATION_DESC contains a citation for the product or DATA_SET_INFORMATION object in which it appears. It provides a string that may be used to cite the product (data set) in a publication. It should follow the standard citation order as outlined in Appendix B, Section 31.5.5.3.1 of the PDS Standards reference, which in turn follows established practice for scientific journals that cite electronic publications (e.g., AGU Reference citation format). The CITATION_DESC must contain sufficient information to locate the product or data set in the PDS archives. For example, the CITATION_DESC in a DATA_SET_INFORMATION object must contain the DATA_SET_ID; it will also likely contain VOLUME_ID information for the archive volumes, an author list, a release date, and so on as appropriate. Note that if CITATION_DESC is used within any product label within a data set, all product labels within that data set must also have a CITATION_DESC, even if they are only filled with 'N/A'. DATA_SET Example: CITATION_DESC = 'Levin, G.V., P.A. Strat, E.A. Guinness, P.G. Valko, J.H. King, and D.R. Williams, VL1/VL2 MARS LCS EXPERIMENT DATA RECORD V1.0, VL1/VL2-M-LCS-2-EDR-V1.0, NASA Planetary Data System, 2000.' Data Product Example: CITATION_DESC = 'Cunningham, C., MINOR PLANET INDEX TO SCIENTIFIC PAPERS, EAR-A-5-

DDR-BIBLIOGRAPHY-V1.0:REFS-REFS-199409, NASA Planetary Data System, 1994.'

column_number The column_number element identifies the location of a specific column within a larger data object, such as a table. For tables consisting of rows (i = 1, N) and columns (j = 1,M), the column_number is the j-th index of any row.

columns The columns element represents the number of columns in each row of a data object. Note: In the PDS, the term 'columns' is synonymous with 'fields'.

confidence_level_note The confidence_level_note element is a text field which characterizes the reliability of data within a data set or the reliability of a particular programming algorithm or software component. Essentially, this note discusses the level of confidence in the accuracy of the data or in the ability of the software to produce accurate results.

coordinate_system_name The coordinate_system_name element provides the full name of the coordinate system to which the state vectors are referenced. PDS has currently defined body-fixed rotating coordinate systems. The Planetocentric system has an origin at the center of mass of the body. The planetocentric latitude is the angle between the equatorial plane and a vector connecting the point of interest and the origin of the coordinate system. Latitudes are defined to be positive in the northern hemisphere of the body, where north is in the direction of Earth's angular momentum vector, i.e., pointing toward the hemisphere north of the solar system invariant plane. Longitudes increase toward the east, making the Planetocentric system right-handed. The Planetographic system has an origin at the center of mass of the body. The planetographic latitude is the angle between the equatorial plane and a vector through the point of interest, where the vector is normal to a biaxial ellipsoid reference surface. Planetographic longitude is defined to increase with time to an observer fixed in space above the object of interest. Thus, for prograde rotators (rotating counter clockwise as seen from a fixed observer located in the hemisphere to the north of the solar system invariant plane), planetographic longitude increases toward the west. For a retrograde rotator, planetographic longitude increases toward the east. Note: If this data element is not present in the PDS Image Map Projection Object (for pre-V3.1 PDS Standards), the default coordinate system is assumed to body-fixed rotating Planetographic.

coordinate_system_type There are three basic types of coordinate systems: body-fixed rotating, body-fixed non-rotating and inertial. A

body-fixed coordinate system is one associated with a body (e.g., planetary body or satellite). In contrast to inertial coordinate systems, a body-fixed coordinate system is centered on the body and rotates with the body (unless it is a non-rotating type). For the inertial coordinate system type, the coordinate system is fixed at some point in space. Note: If this data element is not present in the PDS Image Map Projection Object (for pre-V3.1 PDS Standards), the default coordinate system is assumed to be body-fixed rotating Planetographic.

data_format The data_format element supplies the name of the data format or language that was used to archive the science data that this software accesses.

data_object_type The data_object_type element identifies the data object type of a given set of data. Example values: IMAGE, MAP, SPECTRUM Note: Within the PDS, data object types are assigned according to the standards outlined in the PDS Standards Reference. Note: within AMMOS and only for the Magellan catalog, this element is used as an alias for data_set_id. The use of data_object_type as such provides backward compatibility with earlier AMMOS conventions. The use of this element as an alias for data_set_id is not recommended for any new tables. See data_set_id.

data_set_desc The data_set_desc element describes the content and type of a data set and provides information required to use the data (such as binning information).

data_set_id The data_set_id element is a unique alphanumeric identifier for a data set or a data product. The data_set_id value for a given data set or product is constructed according to flight project naming conventions. In most cases the data_set_id is an abbreviation of the data_set_name. Example value: MR9/VO1/VO2-M-ISS/VIS-5-CLOUD-V1.0. Note: In the PDS, the values for both data_set_id and data_set_name are constructed according to standards outlined in the Standards Reference.

data_set_name The data_set_name element provides the full name given to a data set or a data product. The data_set_name typically identifies the instrument that acquired the data, the target of that instrument, and the processing level of the data. Example value: MR9/VO1/VO2 MARS IMAGING SCIENCE SUBSYSTEM/VIS 5 CLOUD V1.0. See also: data_set_id. Note: In PDS, the data_set_name is constructed according to standards outlined in the Standards Reference. Note: This element is defined in the AMMOS Magellan catalog as an alias for file_name to provide backward compatibility

data_set_release_date The data_set_release_date element provides the date when a data set is released by the data producer for archive or publication. In many systems this represents the end of a proprietary or validation period. Formation rule: YYYY-MM-DD Note: In AMMOS, the data_set_release_date element is used to identify the date at which a product may be released to the general public from proprietary access. AMMOS-related systems should apply this element only to proprietary data.

data_set_terse_desc A brief description of the data set

data_type The data_type element supplies the internal representation and/or mathematical properties of a value being stored. When DATA_TYPE is used within a FIELD object definition, its value applies only when the field is populated. Note: In the PDS, users may find a bit-level description of each data type in the Standards Reference document.

dd_version_id This element identifies the version of a PDS dictionary. Current PDS practice is to identify a data dictionary with the identifier used for the PDS Catalog build in which it resides, e.g., pdscat1r47, pdscat1r48, and so on. This keyword will use the upper case representation of the catalog identifier, e.g., PDSCAT1R47, PDSCAT1R48, etc.

description The description element provides a free-form, unlimited-length character string that represents or gives an account of something.

discipline_desc The discipline_desc element describes the discipline identified by the discipline_name element.

discipline_name The discipline_name element identifies the major academic or scientific domain or specialty of interest to an individual or to a PDS Node.

document_format The document_format element represents the manner in which documents are stored, such as TEX, POSTSCRIPT, TIFF, etc. Version numbers for these formats should be included when appropriate, such as 'WORDPERFECT 5.0'.

document_name The document_name element provides the name of a document.

document_topic_type The document_topic_type element is a keyword which identifies the major topic of a reference document.

electronic_mail_id The electronic_mail_id element provides an individual's mailbox name on the electronic mail system identified by the electronic_mail_type element.

electronic_mail_type The electronic_mail_type element identifies an electronic mail system by name. Example values: TELEMAIL, NSI/DECNET.

encoding_type The ENCODING_TYPE element indicates the type of compression or encryption used for data storage. cf. inst_cmprs_name.

fax_number The fax_number data element provides the area code and telephone number needed to transmit data to an individual or a node via facsimile machine.

field_delimiter The FIELD_DELIMITER indicates the single character used to separate variable-width FIELDs in a SPREADSHEET object. The field delimiter must be chosen from the set of standard values.

field_number The FIELD_NUMBER is the sequential number of the enclosing FIELD object within the current SPREADSHEET definition. FIELD objects should be numbered from the beginning of the record to the end.

fields The FIELDS element is the number of FIELD objects defined within the enclosing SPREADSHEET object.

file_name The file_name element provides the location independent name of a file. It excludes node or volume location, directory path names, and version specification. To promote portability across multiple platforms, PDS requires the file_name to be limited to an 27-character basename, a full stop (. period), and a 3-character extension. Valid characters include capital letters A - Z, numerals 0 - 9, and the underscore character (_).

file_records The file_records element indicates the number of physical file records, including both label records and data records. Note: In the PDS the use of file_records along with other file-related data elements is fully described in the Standards Reference.

first_standard_parallel The first_standard_parallel element is used in Conic projections. If a Conic projection has a single standard parallel, then the first_standard_parallel is the point of tangency between the sphere of the planet and the cone of the projection. If there are two standard parallels (first_standard_parallel, second_standard_parallel), these parallel are the intersection lines between the sphere of the planet and the cone of the projection. The map_scale is defined at the standard parallels.

format A specified or predetermined arrangement of data within a file or on a storage medium. Note: In the PDS, the format element indicates the display specification for a collection of data. It is equivalent to the FORTRAN language format specification. Example values: 'Ew.deEXP', A6, I5.

full_name The full_name element provides the complete name or identifier for a person or object. For an individual, full name includes the name as well as titles and suffixes. For an object, full name provides the spelled-out name that in some cases corresponds to an 'id'.

header_type The HEADER_TYPE element identifies a specific type of header data structure. For example: FITS, VICAR. Note: In the PDS, HEADER_TYPE is used to indicate non-PDS headers.

horizontal_fov The horizontal_field_of_view element provides the angular measure of the horizontal field of view of an instrument.

horizontal_pixel_fov The horizontal_pixel_field_of_view element provides the angular measure of the horizontal field of view of a single pixel.

institution_name The institution_name element identifies a university, research center, or NASA center.

instrument_desc The instrument_desc element describes a given instrument.

instrument_host_desc The instrument_host_desc data element describes the spacecraft or earthbase from which particular instrument measurements were taken. For spacecraft, this description addresses the complement of instruments carried, the on-board communications and data processing equipment, the method of stabilization, the source of power and the capabilities or limitations of the spacecraft design which are related to data-taking activities. The description may be a synopsis of available mission documentation.

instrument_host_id The instrument_host_id element provides a unique identifier for the host where an instrument is located. This host can be either a spacecraft or an earth base (e.g., and observatory or laboratory on the earth). Thus, the instrument_host_id element can contain values which are either spacecraft_id values or earth_base_id values.

instrument_host_name The instrument_host_name element provides the full name of the host on which an instrument is based. This host can be either a spacecraft or an earth base. Thus, the instrument_host_name element can contain values which are either spacecraft_name values or earth_base_name values.

instrument_host_type The instrument_host_type element provides the type of host on which an instrument is based. For example, if the instrument is located on a spacecraft, the instrument_host_type element would have the value SPACECRAFT.

instrument_id The instrument_id element provides an abbreviated name or acronym which identifies an instrument. Note: The instrument_id is not a unique identifier for a given instrument. Note also that the associated instrument_name element provides the full name of the instrument. Example values: IRTM (for Viking Infrared Thermal Mapper), PWS (for plasma wave spectrometer).

instrument_name The instrument_name element provides the full name of an instrument. Note: that the associated instrument_id element provides an abbreviated name or acronym for the instrument. Example values: FLUXGATE MAGNETOMETER, NEAR_INFRARED MAPPING SPECTROMETER.

instrument_type The instrument_type element identifies the type of an instrument. Example values: POLARIMETER, RADIOMETER, REFLECTANCE SPECTROMETER, VIDICON CAMERA.

interchange_format The interchange_format element represents the manner in which data items are stored. Example values: BINARY, ASCII.

item_bytes The item_bytes data element represents the size in bytes of an item within a data object such as a column. Notes: (1) In the PDS, the term item_bytes is distinguished from the term bytes because both elements may appear in a single data object definition (e.g., a label) and refer to different parts of the data object. In an object such as a column, bytes represents the size of the column. Should the column be split into equal items, item_bytes would represent the size of each item. (2) In a field object, item_bytes specifies the maximum size of each item.

item_offset The item_offset data element indicates the number of bytes from the start of one item to the start of the next item in any ASCII column or array.

items The items element defines the number of identical parts into which a single object, such as a column or field, has been divided. See also: repetitions. Note: In the PDS, the data element ITEMS is used for subdivision of a single object, such as a column or a field. REPETITIONS is used for multiple occurrences of objects, such as in a container. For a fuller description of the use of these data elements, please refer to the Standards Reference.

label_records The label_records element indicates the number of physical file records that contain only label information. The number of data records in a file is determined by subtracting the value of label_records from the value of file_records. Note: In the PDS, the use of label_records along with other file-related data elements is fully described in the Standards Reference.

label_revision_note The LABEL_REVISION_NOTE element is a free-form unlimited length character string providing information regarding the revision status and authorship of a PDS label. This should include the latest revision date and author of the current version, but may include a more complete history. This element is required in all Catalog labels and should be the second element in the label. Example: '1999-06-07 SBN:raugh Auto-generated, 1999-07-08 CN:JSH Updated;'

last_name The last_name element provides the last name (surname) of an individual.

line_display_direction The line_display_direction element is the preferred orientation of lines within an image for viewing on a display device. The default value is down, meaning lines are viewed top to bottom on the display. See also SAMPLE_DISPLAY_DIRECTION. Note: The image rotation elements such as TWIST_ANGLE, CELESTIAL_NORTH_CLOCK_ANGLE, and BODY_POLE_CLOCK_ANGLE are all defined under the assumption that the image is displayed in its preferred orientation.

line_first_pixel The line_first_pixel element provides the line index for the first pixel that was physically recorded at the beginning of the image array. Note: In the PDS, for a fuller explanation on the use of this data element in the Image Map Projection Object, please refer to the PDS Standards Reference.

line_last_pixel The line_last_pixel element provides the line index for the last pixel that was physically recorded at the end of the image array. Note: In the PDS, for a fuller explanation on the use of this data element in the Image Map Projection Object, please refer to the PDS Standards Reference.

line_projection_offset The line_projection_offset element provides the line offset value of the map projection origin position from the line and sample 1,1 (line and sample 1,1 is considered the upper left corner of the digital array). Note: that the positive direction is to the right and down.

line_samples The line_samples element indicates the total number of data instances along the horizontal axis of an image.

lines The lines element indicates the total number of data instances along the vertical axis of an image. Note: In PDS label convention, the number of lines is stored in a 32-bit integer field. The minimum value of 0 indicates no data received.

map_projection_desc The map_projection_desc element describes the map_projection_type unambiguously. It shall contain the mathematical expressions (it may even contain the source code or pseudo code, with comments) and any assumptions (e.g. the planet is assumed spherical). Additionally it shall describe the planet eccentricity, the treatment of the a_axis_radius, b_axis_radius, and c_axis_radius when the projection was created, and where the map_scale (or map_resolution) is defined.

map_projection_rotation The map_projection_rotation element provides the clockwise rotation, in degrees, of the line and sample coordinates with respect to the map projection origin (line_projection_offset, line_projection_offset) This parameter is used to indicate where 'up' is in the projection. For example, in a polar stereographic projection does the zero meridian go center to bottom, center to top, center to left, or center to right? The polar projection is defined such that the zero meridian goes center to bottom. However, by rotating the map projection, the zero meridian can go in any direction. Note: 180 degrees is at the top of the North Pole and 0 degrees is at the top of the South Pole. For example, if 0 degrees is at the top of the North Pole than the map_projection_rotation would be 180 degrees.

map_projection_type The map_projection_type element identifies the type of projection characteristic of a given map. Example value: ORTHOGRAPHIC.

map_resolution The map_resolution element identifies the scale of a given map. Please refer to the definition for map_scale for a more complete definition. Note: map_resolution and map_scale both define the scale of a map except that they are expressed in different units: map_resolution is in PIXEL/DEGREE and map_scale is in KM/PIXEL.

map_scale The map_scale element identifies the scale of a given map. The scale is defined as the ratio of the actual distance between two points on the surface of the target body to the distance between the corresponding points on the map. The map_scale references the scale of a map at a certain reference point or line. Certain map projections vary

in scale throughout the map. For example, in a Mercator projection, the map_scale refers to the scale of the map at the equator. For Conic projections, the map_scale refers to the scale at the standard parallels. For an Orthographic point, the map_scale refers to the scale at the center latitude and longitude. The relationship between map_scale and the map_resolution element is that they both define the scale of a given map, except they are expressed in different units: map_scale is in KM/PIXEL and map_resolution is in PIXEL/DEGREE. Also note that one is inversely proportional to the other and that kilometers and degrees can be related given the radius of the planet: 1 degree = (2 * RADIUS * PI) / 360 kilometers.

maximum The maximum element indicates the largest value occurring in a given instance of the data object. Note: For PDS and Mars Observer applications – because of the unconventional data type of this data element, the element should appear in labels only within an explicit object, i.e. anywhere between an 'OBJECT =' and an 'END_OBJECT'.

maximum_latitude The maximum_latitude element specifies the northernmost latitude of a spatial area, such as a map, mosaic, bin, feature, or region. See latitude.

maximum_sampling_parameter The maximum_sampling_parameter element identifies the maximum value at which a given data item was sampled. For example, a spectrum that was measured in the 0.4 to 3.5 micrometer spectral region would have a maximum_sampling_parameter value of 3.5. The sampling parameter constrained by this value is identified by the sampling_parameter_name element. Note: The unit of measure for the sampling parameter is provided by the unit element.

mean The mean element provides the average of the DN values in the image array. Note: For the Mars Pathfinder IMP camera, this was the average of only those pixels within the valid DN range of 0 to 4095.

median The median element provides the median value (middle value) occurring in a given instance of the data object. Because of the unconventional data type of this data element, the element should appear in labels only within an explicit object, i.e. anywhere between an 'OBJECT =' and an 'END OBJECT'. Note: For the Mars Pathfinder IMP camera, this was the median value of only those pixels within the valid DN range of 0 to 4095. Note: For Mars Pathfinder, refers specifically to the median DN value in the image array.

medium_type The medium_type element identifies the physical storage medium for a data volume. Examples: CD-ROM, CARTRIDGE

TAPE.

minimum The minimum element indicates the smallest value occurring in a given instance of the data object. Note: For PDS and Mars Observer applications – because of the unconventional data type of this data element, the element should appear in labels only within an explicit object, i.e. anywhere between an 'OBJECT =' and an 'END_OBJECT'.

minimum_latitude The minimum_latitude element specifies the southernmost latitude of a spatial area, such as a map, mosaic, bin, feature, or region. See latitude.

minimum_sampling_parameter The minimum_sampling_parameter element identifies the minimum value at which a given data item was sampled. For example, a spectrum that was measured in the 0.4 to 3.5 micrometer spectral region would have a minimum_sampling_parameter value of 0.4. The sampling parameter constrained by this value is identified by the sampling_parameter_name element. Note: The unit of measure for the sampling parameter is provided by the unit element.

missing_constant The missing_constant element supplies the value used to indicate that no data were available. Note: The MISSING_CONSTANT element should appear only within an explicit object definition – i.e. anywhere between an 'OBJECT =' and an 'END_OBJECT'. MISSING_CONSTANT assumes the data type of its parent object.

mission_alias_name The mission_alias_name element provides an official name of a mission used during the initial design, implementation, or prelaunch phases. Example values: mission_name:MAGELLAN, mission_alias_name:VENUS RADAR MAPPER. The mission_alias_name element accepts set notation for multiple values.

mission_desc The mission_desc element summarizes major aspects of a planetary mission or project, including the number and type of spacecraft, the target body or bodies and major accomplishments.

mission_name The mission_name element identifies a major planetary mission or project. A given planetary mission may be associated with one or more spacecraft.

mission_objectives_summary The mission_objectives_summary element describes the major scientific objectives of a planetary mission or project.

mission_start_date The mission_start_date element provides the date of the beginning of a mission in UTC system format. Formation rule: YYYY-MM-DDThh:mm:ss[.fff]

mission_stop_date The mission_stop_date element provides the date of the end of a mission in UTC system format. Formation rule: YYYY-MM-DDThh:mm:ss[.fff]

node_id The node_id element provides the node id assigned to a science community node.

node_name The node_name element provides the officially recognized name of a PDS Node.

not_applicable_constant The not_applicable_constant element supplies the numeric value used to represent the figurative constant 'N/A'. 'N/A' (Not Applicable) is defined as indicating when values within the domain of a particular data element do not apply in a specific instance.

note The note element is a text field which provides miscellaneous notes or comments (for example, concerning a given data set or a given data processing program).

offset The offset element indicates a shift or displacement of a data value. See also: scaling_factor. Note: Expressed as an equation: true value = offset value + (scaling factor x stored value).

on_line_identification The on_line_identification element is a unique identifier for product resources which are on-line. It may be a URL to a home page, an e-mail address, an ftp site or a jukebox. An on_line_identification element may be associated with a data set, data set collection, mission, instrument, host, target or volume.

on_line_name The on_line_name element is a unique name which corresponds to a given on_line_identification element. It is used to create HTML links to appropriate home pages.

orbit_direction The orbit_direction element provides the direction of movement along the orbit about the primary as seen from the north pole of the 'invariable plane of the solar system', which is the plane passing through the center of mass of the solar system and perpendicular to the angular momentum vect or of the solar system orbit motion. PROGRADE for positive rotation according to the right-hand rule, RETROGRADE for ne gative rotation. See also: orbital_inclination

pds_address_book_flag The pds_address_book_flag data element indicates whether or not a registered PDS user will have an entry in the PDS telephone directory.

pds_affiliation The pds_affiliation data element describes the type of relationship an individual has with a PDS node. (e.g., staff, advisory group, etc..)

pds_user_id The pds_user_id element provides a unique identifier for each individual who is allowed access to the PDS. The system manager at the Central Node assigns this identifier at the time of user registration.

pds_version_id The PDS_version_id data element represents the version number of the PDS standards documents that is valid when a data product label is created. Values for the PDS_version_id are formed by appending the integer for the latest version number to the letters 'PDS'. Examples: PDS3, PDS4.

platform The platform element describes the available platforms which the software supports.

positive_longitude_direction The positive_longitude_direction element identifies the direction of longitude (e.g. EAST, WEST) for a planet. The IAU definition for direction of positive longitude is adopted. Typically, for planets with prograde rotations, positive longitude direction is to the WEST. For planets with retrograde rotations, positive longitude direction is to the EAST. Note: The positive_longitude_direction keyword should be used for planetographic systems, but not for planetocentric.

preference_id The preference_id element indicates a user's degree of preference for one of a set of alternatives (for example, preference for a particular electronic mail system such as Internet). Values range from 1 to 4, with 1 indicating the highest preference.

primary_body_name The primary_body_name element identifies the primary body with which a given target body is associated as a secondary body.

primary_key In a TABLE object, the PRIMARY_KEY ELEMENT indicates the name(s) of one or more columns in the table that may be used to uniquely identify each row in the table.

producer_full_name The producer_full_name element provides the full_name of the individual mainly responsible for the production of a data set. See also: full_name. Note: This individual does not have to be registered with the PDS.

product_creation_time The product_creation_time element defines the UTC system format time when a product was created. Formation rule: YYYY-MM-DDThh:mm:ss[.fff]

product_id The product_id data element represents a permanent, unique identifier assigned to a data product by its producer. See also: source_product_id. Note: In the PDS, the value assigned to product_id must be unique within its data set. Additional note: The product_id can describe the lowest-level data object that has a PDS label.

protocol_type The protocol_type element identifies the protocol type for the on_line_identification element. Example value: URL, FTP, E-MAIL.

publication_date The publication_date element provides the date when a published item, such as a document or a compact disc, was issued. Formation rule: YYYY-MM-DD

record_bytes The record_bytes element indicates the number of bytes in a physical file record, including record terminators and separators. When RECORD_BYTES describes a file with RECORD_TYPE = STREAM (e.g. a SPREADSHEET), its value is set to the length of the longest record in the file. Note: In the PDS, the use of record_bytes, along with other file-related data elements is fully described in the Standards Reference.

record_type The record_type element indicates the record format of a file. Note: In the PDS, when record_type is used in a detached label file it always describes its corresponding detached data file, not the label file itself. The use of record_type along with other file-related data elements is fully described in the PDS Standards Reference.

records The records data element identifies the number of physical records in a file or other data object.

reference_desc The reference_desc element provides a complete bibliographic citation for a published work. The format for such citations is that employed by the Journal of Geophysical Research (JGR). This format is described in the JGR, Volume 98, No. A5, Pages 7849-7850, May 1, 1993 under 'References'. Data suppliers may also refer to recent issues of the Journal for examples of citations. Elements of a complete bibliographic citation must include, wherever applicable, author(s) or editor(s), title, journal name, volume number, page range and publication date (for journal article citations), or page range, publisher, place of publication, and publication date (for book citations).

reference_key_id The reference_key_id element provides the catalog with an identifier for a reference document. Additionally, it may be used in various catalog descriptions, for example in data_set_desc, as a shorthand notation of a document reference. The reference_key_id element is composed according to the following guidelines: 1. if there is an author for the publication, the general rule is: REFERENCE_KEY_ID = <author's last name><year><letter>, where <author's last name> is a maximum of 15 characters, and may need to be truncated. <year> is 4 characters for the year published. <letter> is optional but consists of one character used to distinguish multiple papers by the same author(s) in the same year. The following variations apply: a. If there is one author: <author's last name><year> Example value: SCARF1980 b. If there are two authors: <first author's last name>&<second author's last name> <year> Example value: SCARF&GURNETT1977 c . If there are three or more authors: <first author's last name>ETAL<year> Example value: GURNETTETAL1979 d. If one author has the same last name as another: <author's last name>,<author's first initial> <year published> Example value: FREUD,A1935 e. If the same author(s) published more than one paper in the same year: <author's last name><year><letter> or <first author's last name>&<second author's last name> <year><letter> or <first author's last name>ETAL<year><letter> Example values: SCARF1980A SCARF&GURNETT1977B f. In cases where an initial reference has been catalogued and published on an Archive medium and subsequent references for the same author and same year are needed at a later date, the following rule applies: Leave the original reference as is, and add a letter to the subsequent references starting with the letter 'B' since the original reference will now be assumed to have an implicit 'A'. For example: PFORD1991, PFORD1991B. Note that if the initial reference has only been catalogued and not yet published, then it can be modified such that the 'A' is explicit, i.e. PFORD1991A. 2. If there is no author for the publication, the general rule is: REFERENCE_KEY_ID = <journal name><document identification> where <journal name> is a maximum of 10 characters, and may need to be abbreviated <document identification> is a maximum of 10 characters. This id may consist of a volume number, and/or document or issue number, and/or year of publication. Example values: SCIENCEV215N4532 JGRV88 JPLD-2468

reference_latitude The reference_latitude element provides the new zero

latitude in a rotated spherical coordinate system that was used in a given map_projection_type.

reference_longitude The reference_longitude element defines the zero longitude in a rotated spherical coordinate system that was used in a given map_projection_type.

reflectance_scaling_factor The reflectance_scaling_factor element identifies the conversion factor from DN to reflectance.

registration_date The registration_date element provides the date as of which an individual is registered as an authorized user of the PDS system. Formation rule: YYYY-MM-DD

repetitions The repetitions data element within a data object such as a container, indicates the number of times that data object recurs. See also: items. Note: In the PDS, the data element ITEMS is used for multiple occruuences of a single object, such as a column. REPETITIONS is used for multiple occurrences of a repeating group of objects, such as a container. For fuller explanation of the use of these data elements, please refer to the PDS Standards Reference.

required_storage_bytes The required_storage_bytes element provides the number of bytes required to store an uncompressed file. This value may be an approximation and is used to ensure enough disk space is available for the resultant file. Note: For Zip file labels, this keyword provides the total size of all the data files in the Zip file after being uncompressed. For the software inventory template, this is often the size of the uncompressed distribution tar file.

resource_class The RESOURCE_CLASS element indicates the type of resource associated with the dataset. For the primary browser, the value should always be set to: application.dataSetBrowserP

resource_id The resource_id element provides an unique indentifier for the resource.

resource_link The RESOURCE_LINK element provides the url of a data set browser that allows searching for particular data products or other ancillary files.

resource_name The Resource_Name element provides the descriptive name of a resource url as it should appear in the Data Set Search results page.

rotation_direction The rotation_direction element provides the direction of rotation as viewed from the north pole of the 'invariable plane of

the solar system', which is the plane passing through the center of mass of the solar system and perpendicular to the angular momentum vector of the solar system. The value for this element is PROGRADE for counter -clockwise rotation, RETROGRADE for clockwise rotation and SYNCHRONOUS for satellites which are tidally locked with the primary. Sidereal_rotation_period and rotation_directio n_type are unknown for a number of satellites, and are not applicable (N/A) for satellites which are tumbling.

rotational_element_desc The rotational_element_desc element describes the standard used for the definition of a planet's pole orientation and prime meridian. The description defines the right ascension and the declination values used to define the planet pole, and the spin angle value of the planet referenced to a standard time (typically EME1950 or J2000 time is used). Periodically, the right ascension, declination, and spin values of the planets are updated by the IAU/IAG/COOSPAR Working Group On Cartographic Coordinates and Rotational Elements because an unambiguous definition of a planet's coordinate system requires these values.

row_bytes The row_bytes element represents the maximum number of bytes in each data object row. Notes: (1) In the PDS, in object definitions for tables, the value of row_bytes includes terminators, separators, and delimiters unless row padding is used. For padding at the beginning of a row, the keyword row_prefix_bytes may be used. For padding at the end of a row, row_suffix_bytes may be used. (2) In object definitions for spreadsheets, the value of row_bytes is the maximum number of bytes possible in the row if each field uses its maximum allocation of bytes and including all delimiters. (3) See the Standards Reference, TABLE and SPREADSHEET objects for more information.

row_prefix_bytes The row_prefix_bytes element indicates the number of bytes prior to the start of the data content of each row of a table. The value must represent an integral number of bytes.

row_suffix_bytes The row_suffix_bytes element indicates the number of bytes following the data at the end of each row. The value must be an integral number of bytes.

rows The rows element represents the number of rows in a data object. Note: In PDS, the term 'rows' is synonymous with 'records'. In PDS attached labels, the number of rows is equivalent to the number of file_records minus the number of label_records, as indicated in the file_object definition.

sample_bit_mask The sample_bit_mask element identifies the active bits in a sample. Note: In the PDS, the domain of sample_bit_mask is dependent upon the currently-described value in the sample_bits element and only applies to integer values. For an 8-bit sample where all bits are active the sample_bit_mask would be 2#11111111#.

sample_bits The sample_bits element indicates the stored number of bits, or units of binary information, contained in a line_sample value.

sample_display_direction The SAMPLE_DISPLAY_DIRECTION element is the preferred orientation of samples within a line for viewing on a display device. The default is right, meaning samples are viewed from left to right on the display. See also LINE_DISPLAY_DIRECTION. Note: The image rotation elements such as TWIST_ANGLE, CELESTIAL_NORTH_CLOCK_ANGLE, and BODY_POLE_CLOCK_ANGLE are all defined under the assumption that the image is displayed in its preferred orientation.

sample_first_pixel The sample_first_pixel element provides the sample index for the first pixel that was physically recorded at the beginning of the image array. Note: In the PDS, for a fuller explanation on the use of this data element in the Image Map Projection Object, please refer to the PDS Standards Reference.

sample_last_pixel The sample_last_pixel element provides the sample index for the last pixel that was physically recorded at the end of the image array. Note: In the PDS, for a fuller explanation on the use of this data element in the Image Map Projection Object, please refer to the PDS Standards Reference.

sample_projection_offset The sample_projection_offset element provides the sample offset value of the map projection origin position from line and sample 1,1 (line and sample 1,1 is considered the upper left corner of the digital array). Note: that the positive direction is to the right and down.

sample_type The sample_type element indicates the data storage representation of sample value.

sampling_parameter_interval The sampling_parameter_interval element identifies the spacing of points at which data are sampled and at which a value for an instrument or dataset parameter is available. This sampling interval can be either the original (raw) sampling or the result of some resampling process. For example, in 48-second magnetometer data the sampling interval is 48. The sampling parameter (time, in the example) is identified by the sampling_parameter_name element.

sampling_parameter_name The sampling_parameter_name element provides the name of the parameter which determines the sampling interval of a particular instrument or dataset parameter. For example, magnetic field intensity is sampled in time increments, and a spectrum is sampled in wavelength or frequency.

sampling_parameter_unit The sampling_parameter_unit element specifies the unit of measure of associated data sampling parameters.

scaling_factor The scaling factor element provides the constant value by which the stored value is multiplied. See also: offset. Note: Expressed as an equation: true value = offset value + (scaling factor x stored value). In PDS Magellan altimetry and radiometry labels, the scaling_factor data element is defined as the value of the conversion factor for the best_non_range_sharp_model_tpt and the non_range_sharp_echo_prof element that multiplies the integer array elements of the best_non_range_sharp_model_tpt and the non_range_sharp_echo_prof to yield their physical values, expressed as equivalent radar cross-sections in units of km**2.

second_standard_parallel Please refer to the definition for first_standard_parallel element to see how second_standard_parallel is defined.

software_desc The software_desc element describes the functions performed by the data processing software. If the subject software is a program library, this element may provide a list of the contents of the library.

software_id The software_id element is a short-hand notation for the software name, typically sixteen characters in length or less (e.g., tbtool,lablib3).

software_license_type The software_license_type element indicates the licensing category under which this software falls.

software_name The software_name element identifies data processing software such as a program or a program library.

software_purpose The software_purpose element describes the intended use of the software.

software_version_id The software_version_id element indicates the version (development level) of a program or a program library.

standard_deviation The standard_deviation element provides the standard deviation of the DN values in the image array. Note: For the

Mars Pathfinder image data, the standard deviation was calculated using only those pixels within the valid DN range of 0 to 4095.

start_bit The start_bit element identifies the location of the first bit of a bit field data object such as a BIT_COLUMN or BIT_ ELEMENT. Bits are numbered from left to right, counting fro 1. The start_bit value assumes that any necessary byte re-ordering has already been performed.

start_byte The start_byte element in a data object identifies the location of the first byte of the object, counting from 1. For nested objects, the start_byte value is relative to the start of the enclosing object.

start_primary_key In a TABLE object, the START_PRIMARY_KEY element indicates the beginning of the range of values for the PRIMARY_KEY column in the table. If PRIMARY_KEY consists of multiple column names, then START_PRIMARY_KEY is a sequence of values, one for each column. The data type of this keyword is determined by the data type of the column of interest.

start_time The start_time element provides the date and time of the beginning of an event or observation (whether it be a spacecraft, ground-based, or system event) in UTC. Formation rule: YYYY-MM-DDThh:mm:ss[.fff].

stop_primary_key In a TABLE object, the STOP_PRIMARY_KEY element indicates the end of the range of values for the PRIMARY_KEY column in the table. If PRIMARY_KEY consists of multiple column names, then STOP_PRIMARY_KEY is a sequence of values, one for each column. The data type of this keyword is determined by the data type of the column of interest.

stop_time The stop_time element provides the date and time of the end of an observation or event (whether it be a spacecraft, ground-based, or system event) in UTC. Formation rule: YYYY-MM-DDThh:mm:ss[.fff].

target_desc The target_desc element describes the characteristics of a particular target.

target_name The target_name element identifies a target. The target may be a planet, satellite,ring,region, feature, asteroid or comet. See target_type.

target_type The target_type element identifies the type of a named target. Example values: PLANET, SATELLITE, RING, REGION, FEATURE, ASTEROID, COMET.

technical_support_type The technical_support_type element indicates the type of support provided for a piece of software. SOURCE_NAME = PDS CN/S. Hughes.

telephone_number The telephone_number element provides the area code, telephone number and extension (if any) of an individual or node. See also: fts_number.

unit The unit element provides the full name or standard abbreviation of a unit of measurement in which a value is expressed. Example values: square meter, meter per second. Note: A table of standard units representing those published by the Systeme Internationale appears in the 'Units of Measurement' section of the PSDD. (Please refer to the table of contents for its location.) The values in this table's 'Unit Name' column constitute the standard values for the data element UNIT.

usage_note The usage_note element provides the information about the use of a particular data element or object within a particular context.

valid_maximum The valid_maximum data element represents the maximum value that is valid for a data object. Valid_minimum and valid_maximum define the valid range of values for a data object, such as -90 to 90 for a column object containing latitude values. Note: this element should appear in labels only between the 'OBJECT =' and 'END_OBJECT=' lines of an object with a specific data type.

valid_minimum The valid_minimum data element represents the minimum value that is valid for a data object. Valid_minimum and valid_maximum define the valid range of values for a data object, such as -90 to 90 for a column object containing latitude values. Note: this element should appear in labels only between the 'OBJECT =' and 'END_OBJECT=' lines of an object with a specific data type.

vertical_fov The vertical_field_of_view element provides the angular measure of the vertical field of view of an instrument.

vertical_pixel_fov The vertical_pixel_field_of_view element provides the angular measure of the vertical field of view of a single pixel.

volume_desc The volume_desc element describes the content and type of data contained in the volume.

volume_format The volume_format element identifies the logical format used in writing a data volume, such as ANSI, TAR, or BACKUP for tape volumes and ISO-9660, HIGH-SIERRA, for CD-ROM volumes.

volume_id The volume_id element provides a unique identifier for a data volume. Example: MG_1001.

volume_name The volume_name element contains the name of a data volume. In most cases the volume_name is more specific than the volume_set_name. For example, the volume_name for the first volume in the VOYAGER IMAGES OF URANUS volume set is: Volume 1: Compressed Images 24476.54 - 26439.58

volume_set_id The volume_set_id element identifies a data volume or a set of volumes. Volume sets are normally considered as a single orderable entity. Examples: USA_NASA_PDS_MG_1001, USA_NASA_PDS_GR_0001_TO_GR_0009

volume_version_id The volume_version_id element indentifies the version of a data volume. All original volumes should use a volume_version_id of 'Version 1'. Versions are used when data products are remade due to errors or limitations in the original volumes (test volumes, for example), and the new version makes the previous volume obsolete. Enhancements or revisions to data products which constitute alternate data products should be assigned a unique volume id, not a new version id. Examples: Version 1, Version 2.

9 Glossary

The following glossary contains a list of terms used within this specification and the definitions for those terms.

Aggregation Aggregation is the act of gathering something together

Associations Associations are the relationships to indicate that the two variables are related.

Attributes Attributes control the context in which elements are defined. Attributes can be related to element identification, or identify one or more context in which the element applies, or one or more options to be used, etc.

Cardinality Cardinality defines the relationship between the entities in terms of numbers. In mathematics, the cardinality of a set is a measure of the 'number of elements of the set'.

Class Class or object class is a template definition of the methods and variables in a particular kind of object. An object is a specific instance of a class. Object contains real values instead of variables. A class can have subclasses that can inherit all or some of the characteristics of the class.

Class_Hierarchy Class hierarchy is a classification of object types, denoting objects as the instantiations of classes inter-relating the various classes by relationships such as 'inherits', 'extends', 'is an abstraction of', and 'an interface definition'

Concept Concept is an abstract idea associated with a corresponding representation that denotes all of the objects in a given class of entities, interactions, or relationships between them.

Conceptualization Conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose.

Context Context is the background and specific circumstances of a subject.

Core_Metadata Core metadata is the metadata common across all domains. The purpose of core metadata is for information discovery across the shared space, to support, at a minimum, the initial data discovery.

Data_Element_Attributes Data element attributes are descriptors of a descriptor. They are the meta-metadata.

Data_Elements Data elements are discrete units of data or metadata. They are an elementary piece of information in a data dictionary. A data element is a data descriptor for which the definition, identification, representation and permissible values are specified by means of a set of data element attributes.

Data_Model Data model is the analysis of data objects that are used in a context and the identification of the relationships among these data objects. The modeling of representations of the data objects in an information system

Data_Object_Description Data Object Description - A Data Object Description is an object class that is used to provide a description for a data object.

Data_Types Data types indicate the types of data that are either number types, composite types (integer numbers, real numbers, strings), numeric data type ranges, abstract data type (associative array, complex number, container, deque, list, multimap, priority queue, queue, set, stack, string, tree).

Data_object Data object is anything that exists in storage and on which operations can be performed. A sequence of digital bits.

Domain_Metadata Domain metadata is the metadata that is specific within a given domain.

Entity Entity is a person, object, place or event for which information is collected.

Information_Model Information model is typically a data model that has a richer set of relationships that add 'meaning' and 'change data into information', for example, the relationship of associating an image data structure with an image map projection. The model uses classifiers to collect properties.

Information_Object Information Object comprises two parts, an internal data object and the metadata that describes the structure and prescribes the nature of the internal data object with semantic information.

Knowledge_Representation Knowledge representation is a method used to code knowledge in an expert system, typically a series of IF-THEN rules.

Labeled_Data_Object A labeled data object consists of a data object in association with a data dbject description and optionally a pointer to the data object.

Meta-Metadata Meta-Metadata is the information about the metadata.

Metadata Metadata is a structured data that contains a definition or description about an object or resource, whether it is physical or electronic. It adds layers of meaning to data or information, it is data about data, or data about information, is sometimes referred to as pure semantics.

Model Model is a formalized description.

Object-Oriented_Methodology Object-Oriented methodology centers on the concept of classes. The methodology organized around objects rather than actions and data rather than logic rather than actions

Object_Class Class or object class is a template definition of the methods and variables in a particular kind of object. An object is a specific instance of a class. Object contains real values instead of variables. A class can have subclasses that can inherit all or some of the characteristics of the class.

Object_Model Object model is an abstract model that describes how data is represented in object-oriented concepts.

Objects Objects are the units that are derived from the process.

Ontology Ontology defines the common words and concepts used to describe and represent an area of knowledge. Taxonomy can portray the equivalent of two-dimensional space, while ontology can portray the equivalent of three or more dimensions, since ontology adds semantics, at varying level s of complexity.

Relationship Relationship is a particular type of interaction existing between classes/entities, It involves one or more classes/entities Relations have direction.

Schema A schema designates a set of semantic units along with their attributes, such as name, identifier, definition, or relationship to other semantic units. Metadata element sets are called schema.

Semantics Semantics is the study of meaning.

Taxonomy Taxonomy is a tree structure that represents information entities in a hierarchical manner. It is a classification system or an organization of related keywords or concepts that is of great explanatory value and meets the established rules of existing model(s). Creating taxonomy requires considerable domain expertise.