The PDS Information Model Specification

PDS Engineering Node

23 January 2008

Why We are here today

- Explain how to read the specification document
  - Understand the PDS Model as presented in the specification
- Explain the importance of a formal specification
  - Why this is the right thing to do for the PDS
- Prepare the PDS Tech staff to be able to review and comment on the specification document
- Prepare the PDS Tech staff for explaining the concepts to Node Managers
- Respond to MC directives

• EN-09 (Hughes): Take the PDS3 formal specification to the Technical Group for evaluation.

• EN-10 (Technical Group/Crichton; by TBD): Review the PDS3 specification, and report to MC whether they understand it; if so, explain the specification to the MC

Advantages of Model-Based PDS

- More consistency across the system
  - Concepts are simpler, broader, and more logical than rule books
  - Fewer special conditions, exceptions
  - Easier to understand for
    - Newbies (especially novice data providers)
    - PDS veteran users
    - Staff
    - International partners
- Simpler management
  - Standards and implementations which follow are also simpler and more logical
  - Less work required
  - Faster updates
  - Allows evolution based on requirements
- Model is separated from implementation
  - Definitions are explicit and in commonly accepted data engineering notation
  - Model remains valid across diverse domains
    - institutional, geographical / international, and political boundaries
  - Agnostic to PDS3, PDS4, IPDA

What we hope to accomplish today

1. Provide a simple explanation of a very technical issue:
   - What the Model is
   - How the Model was derived
   - What the Model represents
   - How the Model was captured
   - How will PDS will utilize the Model
   - Why the Model is captured as a Specification

2. “Read” the Specification

3. To introduce the specification document and argue that it can become the formal definition of the PDS model.

What the Model Is

- A Data Model is a formal representation of the entities within a domain.
    - Entities are named and defined using attributes.
    - Entities are associated using prescribed relationships.

- Data Modeling is the process of creating a data model by applying data model theory to create a data model instance.

We are here. →

- A conceptual data model formalizes the description of an "entity" without constraining how that description is implemented

- A logical data model is derived from conceptual model by constraining the conceptual model for a particular data manipulation technology. (e.g. Relational)

- A physical data model is the physical implementation of the logical data model. (e.g. SQL Create statements)

How the Model was derived

- The Model was derived from information in:
  - PDS 3 Standards Reference
  - PDS Data Dictionary
  - Process documents (*e.g.,* APG, PAG)
  - Data products in the PDS Archive (as submitted by the DNs)
  - Catalog Design Document (circa 1900+)
- The Model reflects the operational system, PDS 3 standards, and node input

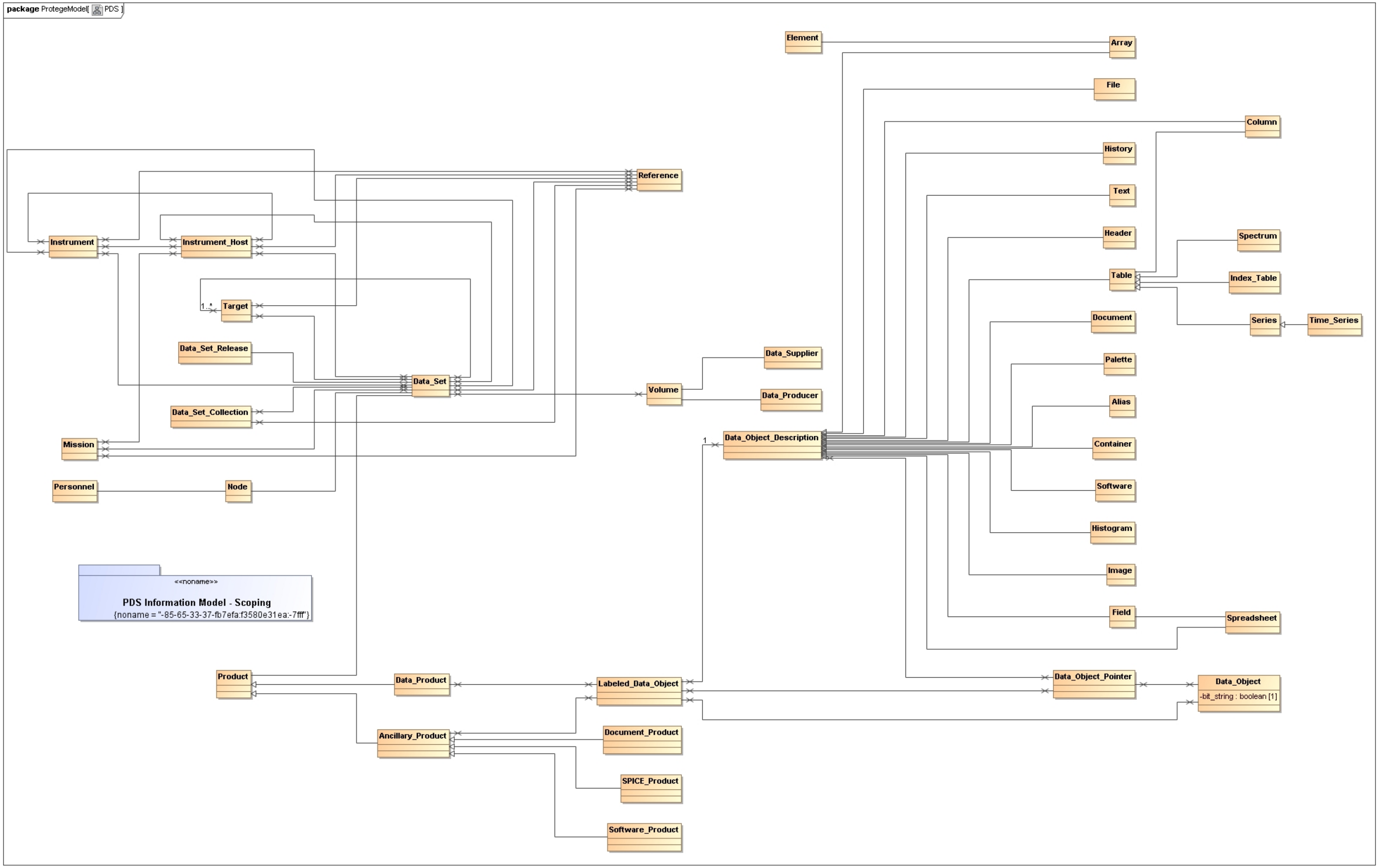
From mostly “unstructured narrative” to a precisely structured domain

What the Model is representative of

- The Model represents the PDS3 system (inclusive of known "problems"):
  - PDS Catalog
    - Mission, Instrument Host, Instrument, Data Set…
  - Data Dictionary
    - PSDD (database schema)
  - Data Products
    - As generically defined in the PSDD and refined by input from DNs
  - Data Formats
    - As defined in the Standards Reference
  - Ancillary Entities
    - Map Projection
  - Operational Entities
    - Repository, Registry, Resource, Release

Reflects the operational system, PDS 3 standards, and node input

A view of the model from 30K feet



How the Model is captured

- The Model was captured into a data engineering tool specifically developed for data model management:
  1. For diverse and complex science domains
  2. Management of evolutionary changes and additions to the information model
     - Easily modifiable to extend existing elements, deprecate obsolete elements, and add new elements
  3. Maintain the information model independent of system implementation choices
  4. Can export the model to commonly accepted data engineering languages.

Represents an initial attempt to get to a PDS 3 core

How PDS will utilize the Model

- **The Model will be utilized by the PDS in various ways:**
  - **Archivists**: an explicit "how to" for creating PDS compliant data products and archive volumes
  - **PDS Newbies**: the model defines the PDS archive
  - **Validation Tools**: an explicit set of rules the tools can use to validate catalog files, data products, and archive volumes
  - **Label Template Design Tools**: an explicit set of rules the tools can use to assist users in creating PDS compliant labels (templates)
  - **EN**: improve maintenance (ease of change, rapid change, automate) of the PDS Standards; Use the information model to drive implementations and documentation

To facilitate movement from PDS3 to PDS4

Why the Model is expressed as a Specification

- The data engineering tool captures the model using a language that is complex and not easy to understood.
  - An analogy is a document prepared in MS WORD. Internally it is stored in a complex formal language but is “rendered” to the screen and documents for end users.

- For the model to be human or machine readable it is exported from the data engineering tool into commonly accepted data engineering languages and notations.

.

- A specification is a set of requirements.
  - The formal data engineering language can be used by developers.
  - Additional text is provided to “tie” the information together and explain the diagrams and tables.
  - Other components of a formal specification are added.

- Once cleaned up and adopted, the specification document becomes the official description of PDS3.

“Reading” the Specification

- **Site Login**
  http://pds-engineering.jpl.nasa.gov/index.cfm?pid=5&cid=26
- **Document URL**
  http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html
- **Introduction**
  http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#1%A0%A0INTRODUCTION
- **Object Models** (e.g.Object Class Definitions)
  - Upper Level Model
    http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#5%A0%A0UPPER%20LEVEL%20OBJECT%20CLASSES
  - Data Formats (Core, Base and Generic)
    http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#6%A0%A0DATA%20FORMAT%20OBJECT%20CLASSES
  - Label Model (e.g. Data Object, Pointer, …)
    http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#7%A0%A0LABEL%20OBJECT%20CLASSES
  - Data Product
    http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#8%A0%A0DATA%20PRODUCT%20OBJECT%20CLASSES
  - Ancillary Object Classes (Image Map Projection)
    http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#9%A0%A0ANCILLARY%20OBJECT%20CLASSES
- **Data Dictionary** (Referenced Data Elements)
  http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#11%A0%A0DATA%20DICTIONARY

Summary

- The PDS3 data model was captured in a data engineering tool to explicitly define the model and to make model management more efficient.

- The conceptual model has been expressed in the specification document using commonly accepted data engineering notation and language.

- The model was written to a formal specification document to be used as an explicit set of requirements to be satisfied by the information system.

  - Once cleaned up and adopted, the specification document becomes the official description of PDS3.

Next Steps

- Tech group reviews the specification document and provides comments
- Tech reps go over the document and concepts with their managers
- Reps report manager’s comments and concerns to tech session
- Tech group endorse the specification document

Time Line

Wed	1/23	TS - introduce document and tasks
Wed– Tue	1/23 – 1/29	Tec reps - review document
		– propose changes
		– discuss with managers
Wed	1/30	TS – report manager's comments/concerns
		– endorse document in principle
Thu – Wed	1/31 – 2/6	WG – final revisions, MC versions
Thu	2/7	WG – distribute MC versions for review to MC & TS
Mon	2/11	MC Telecon

Back Up Slides

Information Model Specification

- **Site Login**
  http://pds-engineering.jpl.nasa.gov/index.cfm?pid=5&cid=26
- **Document URL**
  http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html
- **Introduction**
  http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#1%A0%A0INTRODUCTION
- **Object Models** (e.g.Object Class Definitions)
  - Upper Level Model
    http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#5%A0%A0UPPER%20LEVEL%20OBJECT%20CLASSES
  - Data Formats (Core, Base and Generic)
    http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#6%A0%A0DATA%20FORMAT%20OBJECT%20CLASSES
  - Label Model (e.g. Data Object, Pointer, …)
    http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#7%A0%A0LABEL%20OBJECT%20CLASSES
  - Data Product
    http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#8%A0%A0DATA%20PRODUCT%20OBJECT%20CLASSES
  - Ancillary Object Classes (Image Map Projection)
    http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#9%A0%A0ANCILLARY%20OBJECT%20CLASSES
- **Data Dictionary** (Referenced Data Elements)
  http://pds-engineering.jpl.nasa.gov/system_eng/InfoModel/index.html#11%A0%A0DATA%20DICTIONARY

Evolving the Specification

- **The very act of capturing the PDS data models has brought out their strengths, forced out inconsistencies, and made existing problems an order of magnitude easier to identify and fix.**

- **The formal specification is implementation independent.**
  - Object Description Language (ODL) with its limitations and quirks was ignored.

- **A cursory glance over the compiled list of existing problems and issues associated with the data model indicate that a significant number are addressed by the above two points.**

- **The information model can be easily modified (evolved) to fix problems, extend existing elements, deprecate obsolete elements, and add new elements.**

PDS Data Models

1. **PDS Catalog** – Also known as the Upper Level Model and includes Mission, Instrument, etc
2. **Data Product** - As defined in the Standards Reference
3. **Data Formats** – PDS Objects as defined in Appendix A - PDS Data Object Definitions of the Standards Reference
4. **Ancillary** – Domain specific models such as Map Projection
5. **Data Dictionary** (PSDD) - Data Dictionary database schema
6. **Operational Entities** – Repository, Registry, Resource, Release

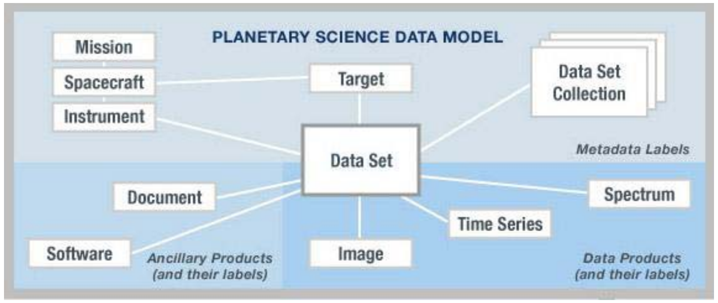
PDS Data Models Mapped to OAIS* Functional Areas

	Preservation	Ingest	Access Query	Access Distribution	Data Management	Administration
Upper Level	x	x	x	x	x	x
Product	x	x	x	x	x	x
Data Format	x	x	x	x	x	
Ancillary	x	x	x		x	
Operational		x	x	x	x	x
Data Dictionary	x	x	x	x	x	x

*Open Archival Information System (OAIS) Reference Model

PDS Catalog Example - Data Set Model

Conceptual Model



Questions

1. How many Targets can be associated with a Data_Set?
2. Is Mission related to Data_Set and is it required or is N/A sufficient?
3. Why does the Data_Set_Information object exist and how many times can it be repeated?
4. What role does Data_Set_Id play.

Specified as

B.1 DATA_SET

The DATA_SET catalog object is used to submit information about a data set to the PDS. The DATA_SET object includes a free-form text description of the data set as well as sub-objects for identifying associated targets, hosts, and references.

B.1.1 Required Keywords

1. DATA_SET_ID

B.1.2 Optional Keywords

None

B.1.3 Required Objects

1. DATA_SET_HOST
2. DATA_SET_INFORMATION
3. DATA_SET_REFERENCE_INFORMATION
4. DATA_SET_TARGET
5. DATA_SET_MISSION

B.1.4 Optional Objects

None

Data Product Example - Label Model

Conceptual Model

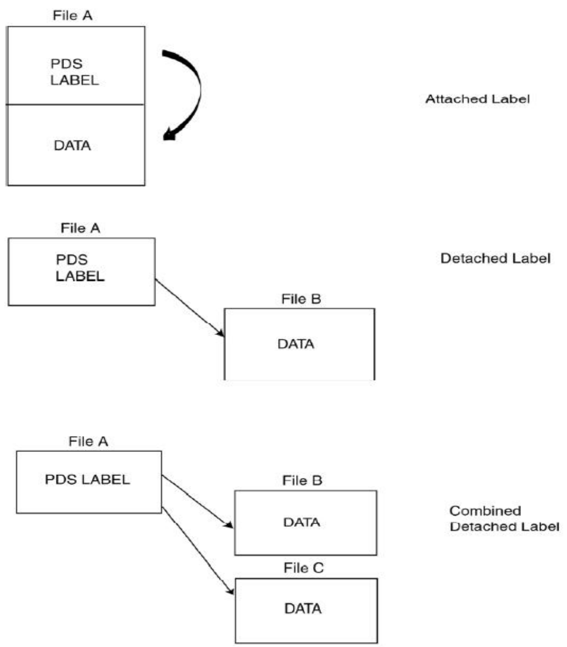
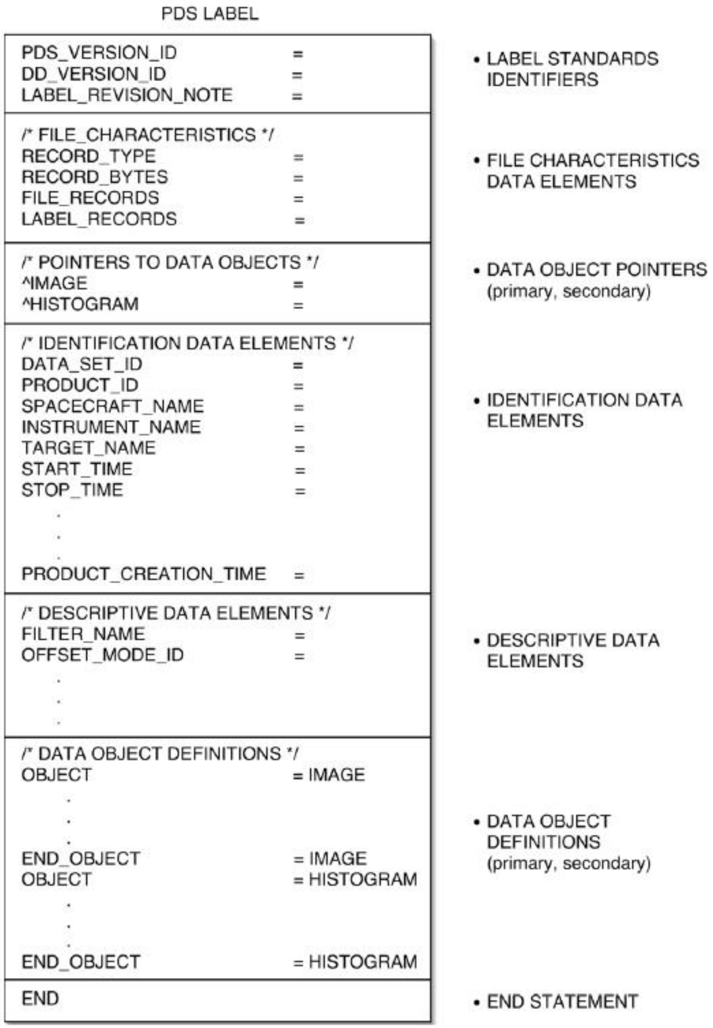


Figure 5.1 Attached, Detached, and Combined Detached PDS Labels

Specified as



Questions

1. What objects are allowed as primary objects?
2. Given a primary object what objects are allowed as secondary objects?
3. What file is being described?
4. Where does map projection information go?

Data Format Example - Image Model

Conceptual Model

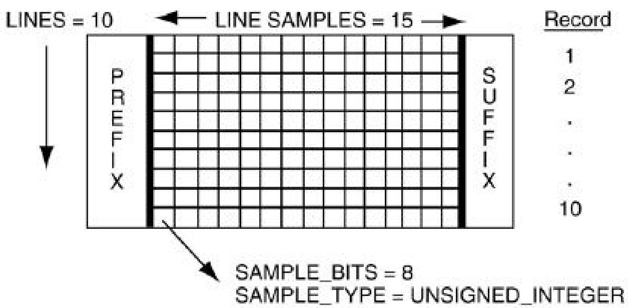


Figure A.1 – Prefix and Suffix Bytes Attached to an Image

Questions

1. How are samples and line samples related?
2. Are all "band" keywords required for banded images.
3. Is "Bands=1" required for a single banded image?
4. Which data elements allow a set of values versus a sequence of values?

Specified as

A.20 IMAGE

An IMAGE object is a two-dimensional array of values, all of the same type, each of which is referred to as a *sample*. IMAGE objects are normally processed with special display tools to produce a visual representation of the samples by assigning brightness levels or display colors to the values. An IMAGE consists of a series of lines, each containing the same number of samples.

The required IMAGE keywords define the parameters for simple IMAGE objects:

- LINES is the number of lines in the image.
- LINE_SAMPLES is the number of samples in each line.
- SAMPLE_BITS is the number of bits in each individual sample.
- SAMPLE_TYPE defines the sample data type.

In more complex images, each individual line may have some attached data which are not part of the image itself (engineering data, checksums, time tags, etc.). In this case the additional, non-image parameters are accounted for as either LINE_PREFIX_BYTES or LINE_SUFFIX_BYTES, depending on whether they occur before or after the image samples in the line. These keywords indicate the total number of bytes used for the additional data, so that software processing the image can clip these bytes before attempting to display or manipulate the image. The structure of the prefix or suffix bytes is most often defined by a TABLE object (in the same label), which will itself have ROW_SUFFIX_BYTES or ROW_PREFIX_BYTES, to allow table-processing software to skip over the image data. Figure A.1 illustrates the layout of prefix and suffix bytes around an image.

A.20.1 Required Keywords

1. LINES
2. LINE_SAMPLES
3. SAMPLE_TYPE
4. SAMPLE_BITS

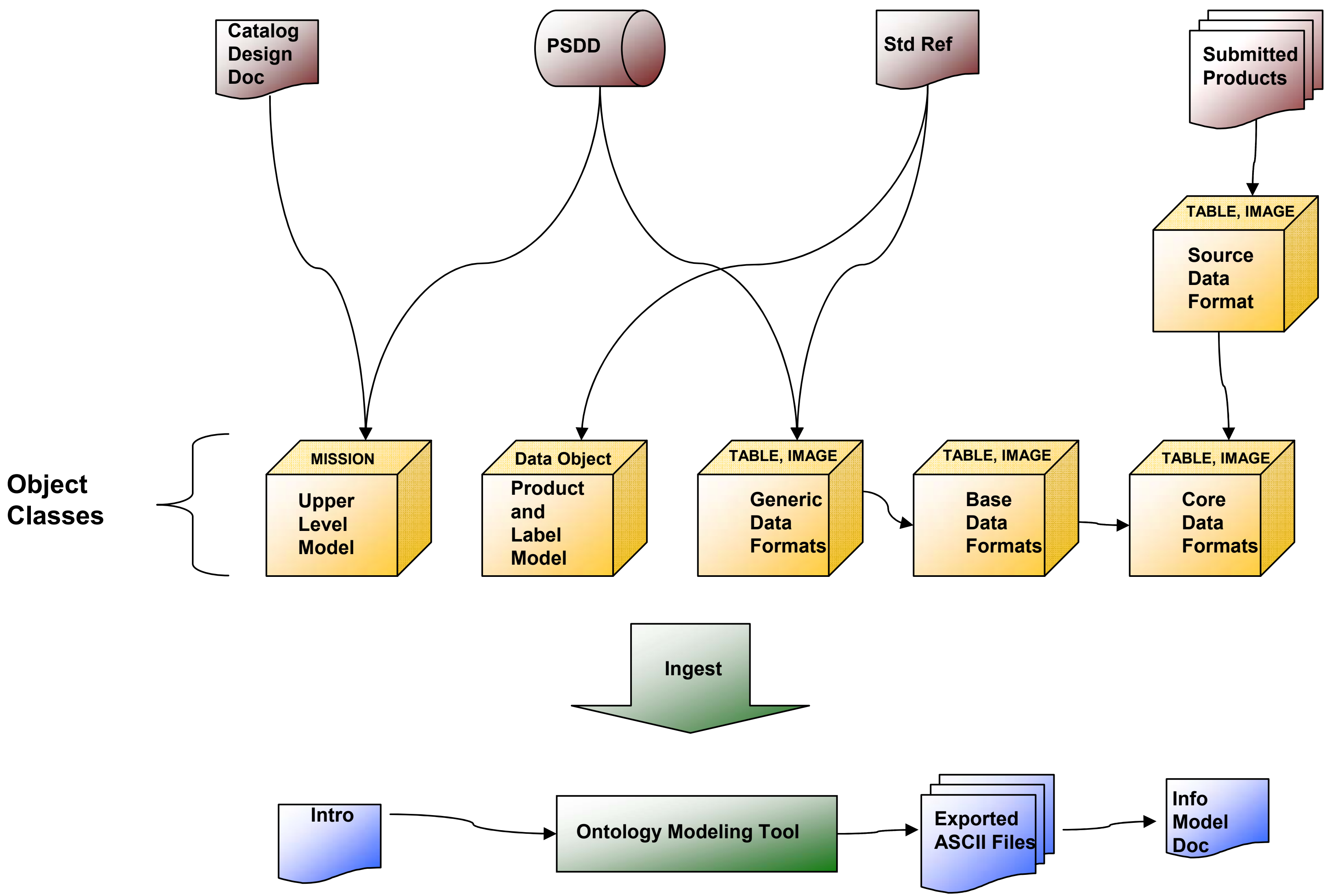
A.20.2 Optional Keywords

1. BAND_SEQUENCE
2. BAND_STORAGE_TYPE
3. BANDS
4. CHECKSUM
5. DERIVED_MAXIMUM

History of the PDS Data Models

- **1988-1989** – PDS Data Modeling team interviews Planetary Scientists at discipline nodes to develop upper level model. ODL developed to capture metadata and load catalog database.
- **1990** – PDS Version 1.0 goes online. ODL used to capture and present metadata on archive volumes. (Tape)
- **1991** – PDS Volume model designed for self-contained archive products on Optical Media. (CD-ROM)
- **1994?** – PDS streamlines the PDS Data Model.
- **2001** – PDS Resource and Release models developed to distribute data through PDS-D. (Distribution System)
- **2004** – PDS Upper Level model captured in modeling tool for prototype “Semantic PDS”. (outside funding)
- **January 2007** – PDS Data Models captured in modeling tool. “Core” Information Model specification was generated for IPDA core requirements project.
- **August – September 2007** – Several “Core” PDS3+ Information Model specifications generated for PDS4 Data Model and IPDA Assessment teams.

Process Graph



Understanding the Specification

- **Current notation* is the Object-Oriented concept of Class Definition**
  - PDS data objects are captured as "classes"
  - PDS data elements are captured as "attributes" of a class
  - Relationships are identified for PDS data elements which are used in multiple classes for linking objects
  - "Subclasses" are identified when one class is a type of another class (e.g., Series is a type of Table)

- **All classes are ultimately derived from the CCSDS Open Archive Information System (OAIS) Reference Model**
  - Unifies catalog files and product labels

* Suggestions are welcome as to how to depict the modeling information for better understanding

Data Product Example

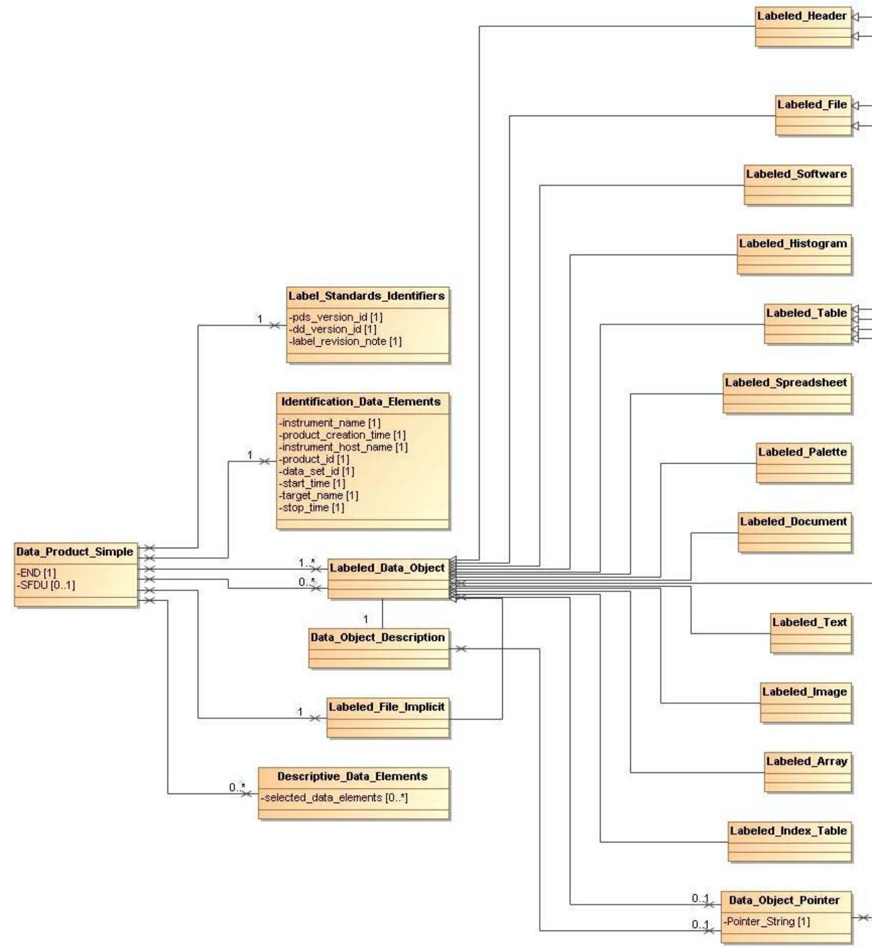
Standards Reference

PDS LABEL

```
PDS_VERSION_ID                    =
DD_VERSION_ID                     =
LABEL_REVISION_NOTE               =

/* FILE_CHARACTERISTICS */
RECORD_TYPE                       =
RECORD_BYTES                      =
FILE_RECORDS                      =
LABEL_RECORDS                     =

/* POINTERS TO DATA OBJECTS */
^IMAGE                            =
^HISTOGRAM                        =

/* IDENTIFICATION DATA ELEMENTS */
DATA_SET_ID                       =
PRODUCT_ID                        =
SPACECRAFT_NAME                   =
INSTRUMENT_NAME                   =
TARGET_NAME                       =
START_TIME                        =
STOP_TIME                         =
      .
      .
      .
PRODUCT_CREATION_TIME             =

/* DESCRIPTIVE DATA ELEMENTS */
FILTER_NAME                       =
OFFSET_MODE_ID                    =
      .
      .
      .

/* DATA OBJECT DEFINITIONS */
OBJECT                            = IMAGE
      .
      .
      .
END_OBJECT                        = IMAGE
OBJECT                            = HISTOGRAM
      .
      .
      .
END_OBJECT                        = HISTOGRAM

END
```

- LABEL STANDARDS IDENTIFIERS
- FILE CHARACTERISTICS DATA ELEMENTS
- DATA OBJECT POINTERS (primary, secondary)
- IDENTIFICATION DATA ELEMENTS
- DESCRIPTIVE DATA ELEMENTS
- DATA OBJECT DEFINITIONS (primary, secondary)
- END STATEMENT

Specification Notation 1



Data Product Example

Standards Reference

PDS LABEL

```
PDS_VERSION_ID              =
DD_VERSION_ID               =
LABEL_REVISION_NOTE         =

/* FILE_CHARACTERISTICS */
RECORD_TYPE                 =
RECORD_BYTES                =
FILE_RECORDS                =
LABEL_RECORDS               =

/* POINTERS TO DATA OBJECTS */
^IMAGE                      =
^HISTOGRAM                  =

/* IDENTIFICATION DATA ELEMENTS */
DATA_SET_ID                 =
PRODUCT_ID                  =
SPACECRAFT_NAME             =
INSTRUMENT_NAME             =
TARGET_NAME                 =
START_TIME                  =
STOP_TIME                   =
      .
      .
      .
PRODUCT_CREATION_TIME       =

/* DESCRIPTIVE DATA ELEMENTS */
FILTER_NAME                 =
OFFSET_MODE_ID              =
      .
      .
      .

/* DATA OBJECT DEFINITIONS */
OBJECT                      = IMAGE
      .
      .
      .
END_OBJECT                  = IMAGE
OBJECT                      = HISTOGRAM
      .
      .
      .
END_OBJECT                  = HISTOGRAM

END
```

- LABEL STANDARDS IDENTIFIERS
- FILE CHARACTERISTICS DATA ELEMENTS
- DATA OBJECT POINTERS (primary, secondary)
- IDENTIFICATION DATA ELEMENTS
- DESCRIPTIVE DATA ELEMENTS
- DATA OBJECT DEFINITIONS (primary, secondary)
- END STATEMENT

Specification Notation 2

Relationship	Entity	Card	Value
Hierarchy	Product		
	. Data Product		
Attribute	END	1	END
	SFDU	0..1	SFDU
Inherited Attribute	URI	0..1	
Association	has AOD	0..*	Ancillary Object Description
	has DDE	0..*	Descriptive Data Elements
	has IDE	1	Identification Data Elements
	has ILF	1	Labeled File Implicit
	has LSI	1	Label Standards Identifiers
	has Primary LDO	1..*	Labeled Time Series
			Labeled Table Binary
			Labeled Spectrum
			Labeled Image
			Labeled Series
			Labeled Table ASCII
			Labeled Table
	has Secondary LDO	0..*	Labeled Histogram
			Labeled Palette
			Labeled Header
			Labeled Data Object
Inherited Association	none		

Further Definitions

- Information Model
  - Data modeling theory suggests that information is derived from data by adding meaning.
  - Meaning or semantics are added through relationships.
  - This suggest that an Information Model is derived from a set of Data Models by adding relationships between the Data Models.

- Formal Specification
  - A information model specification is an explicit set of requirements to be satisfied by an information system.
  - A formal specification is a mathematical description of a model that may be used to develop an implementation.
  - A specification can include:
    - Descriptive title and scope of the specification
    - Date of last effective revision and revision designation
    - Responsible team for questions on the specification, updates, and deviations.
    - The significance or importance of the specification and its intended use.
    - Terminology and definitions to clarify the meanings of the specification
    - Test methods for measuring all specified characteristics
    - Requirements (Conceptual Model)