Planetary Data System

Preparation Tools

Software Requirements and Design Document (SRD/SDD)

Sean Hardman

June 15, 2011
Version 0.2

Jet Propulsion Laboratory
Pasadena, California

CHANGE LOG

Revision	Date	Description	Author
0.1	2010-10-06	Initial draft focusing in the Design and Validate tools.	S. Hardman
0.2	2011-06-15	Updated controlling document references.	S. Hardman

TABLE OF CONTENTS

1.0 INTRODUCTION

The PDS 2010 effort will overhaul the PDS data architecture (e.g., data model, data structures, data dictionary, etc) and deploy a software system (online data services, distributed data catalog, etc) that fully embraces the PDS federation as an integrated system while leveraging modern information technology.

This suite of tools provides functionality for preparing data for ingestion into PDS. This includes product label design and generation along with product validation, transformation and visualization.

1.1 Document Scope and Purpose

This document addresses the use cases, requirements and software design of the Preparation Tools within the PDS 2010 data system. This document is intended for the reviewer of the tools as well as the developer and tester of the tools.

1.2 Method

This combined Software Requirements and Software Design Document (SRD/SDD) represents the software by defining use cases and requirements and by using architecture diagrams, functional descriptions, context diagrams and data flow diagrams for the high-level design. UML diagrams will illustrate the detailed design.

1.3 Notation

The numbering of the requirements in this document will be formatted as **LX.PRP.AA.X**, where:

- **LX** represents the requirements level where X is a number.
- **PRP** is an abbreviation representing the tool requirements section for the specified level.
- **AA** is a two-letter abbreviation representing the requirement sub-category (optional).
- **X** is a unique number within the section and optional sub-category for the requirement.

Following the text of a requirement may be a reference to the requirement or use case from which it was derived. The reference will be in parenthesis. A paragraph following a requirement, which is indented and has a reduced font size, represents a comment providing additional insight for the requirement that it follows. This comment is not part of the requirement for development or testing purposes.

1.4 Controlling Documents

[1] Planetary Data System (PDS) Level 1, 2 and 3 Requirements, March 26, 2010.

[2] Planetary Data System (PDS) 2010 Project Plan, February 2010.

[3] Planetary Data System (PDS) 2010 System Architecture Specification, Version 1.2, May 25, 2011.

[4] Planetary Data System (PDS) 2010 Operations Concept, February 2010.

[5] Planetary Data System (PDS) General System Software Requirements Document (SRD), Version 1.0, June 11, 2011.

1.5 Applicable Documents

[6] Format Transformation Survey, March 15, 2010.

[7] Extensible Markup Language (XML) Specification, 1.0 (Fifth Edition), November 26, 2008.

1.6 Document Maintenance

The component design will evolve over time and this document should reflect that evolution. This document is limited to design content because the specification content will be captured in separate documentation (e.g., Installation Guide, Operation Guide, etc.). This document is under configuration control.

2.0 COMPONENT DESCRIPTION

The Preparation Tools component is a suite of tools providing functionality for preparing data for ingestion into PDS. The following diagram details the context of the Preparation Tools component within the system:

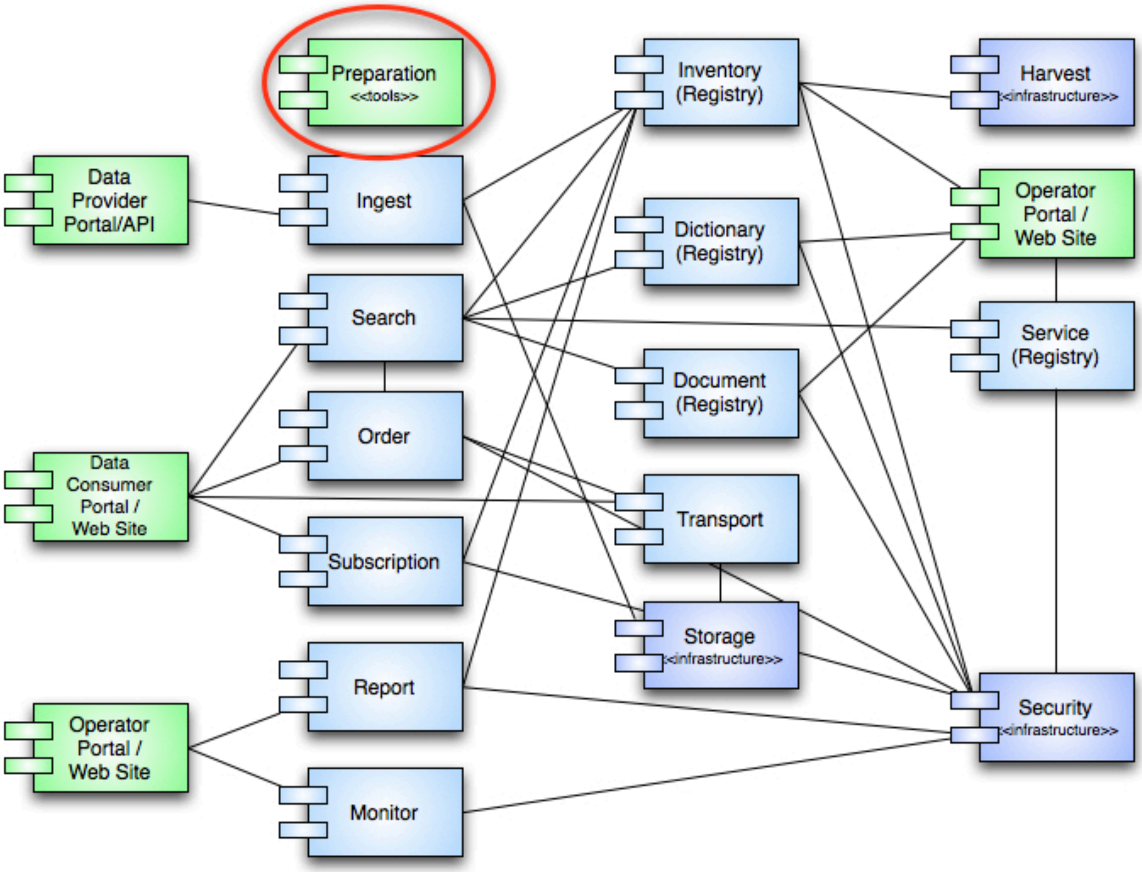

Figure 1: Preparation Tools Context

There are several functions within the PDS system that are well suited for tool-based interfaces. Tool-based interfaces for this component include command-line and GUI applications executed on a user's desktop machine. This document identifies the following suite of tools:

Design Tool
This tool provides functionality for designing product label schemas using the XML Schema standard. Although a PDS-specific tool may be developed in the future, Commercial Off-The-Shelf (COTS) and Open Source tools have been identified to satisfy this functional need. Along with facilitating manipulation of a generic schema into a specific schema, the tool also generates sample product labels to aide users in schema comprehension.

Generate Tool

This tool provides functionality for generating product labels from and conforming to the specific product schema. The tool offers a command-line interface but the intent is to provide software that can be incorporated into a mission pipeline for producing product labels.

Validate Tool

This tool provides functionality for validating product labels and product data. The associated specific schema for the product label specifies syntactic and semantic constraints. The product label itself specifies the constraints for the data.

Transform Tool

This tool provides functionality for transforming product labels and data to and from supported PDS4 formats. The plan is to design the tool with a plug-in framework allowing the PDS community to develop plug-ins for various formats.

Visualize Tool

This tool provides functionality for inspecting and visualizing PDS products. Includes support for products beyond images including tables, etc. Although this is considered a useful tool for the PDS community, its predecessor (NASAView) is commonly used during peer reviews for product verification.

3.0 USE CASES

A use case represents a capability of the component and why the user (actor) interacts with the component. It should be at a high enough level so as not to reveal or imply the internal structure of the system. An actor is an object (e.g., person, application, etc.) outside the scope of the component but interacts with the component. This section captures the use cases for the Preparation Tools based on the description of the component from the previous section. These use cases will be used in the derivation of requirements for the component. The following diagram details the use cases:

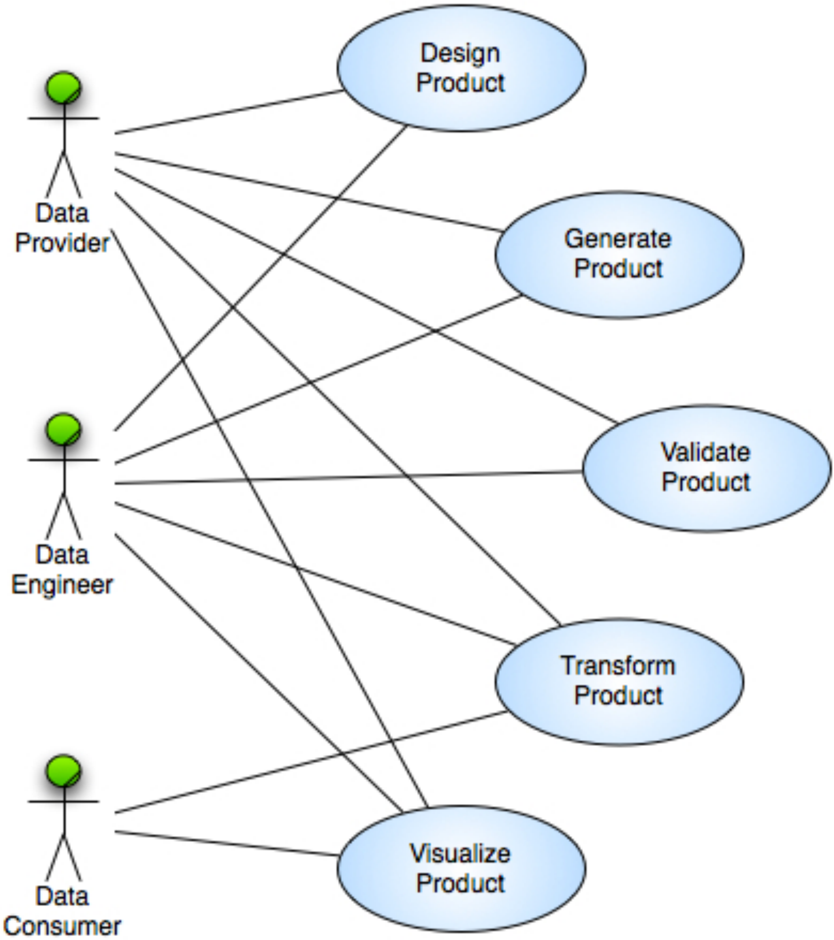

Figure 2: Preparation Tools Use Cases

The above diagram identifies the following actors (represented as stick figures):

Data Consumer
This actor represents the Planetary Scientist, which includes those experienced with solar system exploration missions and those who are mission-naïve. They include graduate students.

Data Engineer
This actor represents a portion of the PDS Technical group that curates the data before and after it enters the PDS system.

Data Provider
This actor represents the mission, instrument team and NASA-funded researcher who are involved with delivering data to the PDS.

The following sections detail the use cases identified in the above diagram.

3.1 Design Product

The procedure for designing a product normally starts with taking a generic product schema and modifying it to become a specific product schema. See the Operations Concept document [4] for more information on product design. This use case pertains to the Data Provider and Data Engineer actors represented as the “User” in the use case.

1. User saves a copy of the generic product schema as the initial cut at generating the specific product schema.
2. User executes the Design Tool and opens the newly saved specific product schema.
3. Design Tool displays the specific product schema and indicates any errors found in the file.
4. User tailors the specific product schema to properly describe the target product. This includes, but is not limited to:
   a. Restricting the value set for appropriate keywords.
   b. Restricting ranges for appropriate keywords.
   c. Removing unnecessary optional blocks and keywords.
   d. Incorporating Node-specific and Mission-specific content where appropriate.
5. Design Tool indicates any errors introduced by the User.
6. User reviews for desired content and structure.
7. User saves the specific product schema.

<u>Alternative: Schema Contains Errors</u>
At step 5, the tool indicates that the specific product schema contains errors.
   a. User corrects the errors.
   b. Return to primary scenario at step 6.

<u>Alternative: Example Label Generated</u>
At step 6, one or more example labels are generated to aide in the review.
   a. User requests generation of one or more example labels.
   b. Design Tool generates one or more example labels based on the specific product schema.
   c. Return to primary scenario at step 6.

3.2 Generate Product

TBD

3.3 Validate Product

Validation of products occurs in two passes. The first pass involves validating the product label against its associated specific product schema. The second pass involves validating the product data against the description in the associated product label. This use case pertains to the Data Provider and Data Engineer actors represented as the “User” in the use case.

1. User executes the Validate Tool specifying one or more products to validate.
2. Validate Tool performs validation of product label(s) based on the appropriate specific product schema(s).
3. Validate Tool performs validation of product data based on the description of the product in the associated product label.
4. Validate Tool generates a report detailing the status of the validation run.

<u>Alternative: User Specifies Schema(s)</u>
At step 1, the user wants to validate the product(s) against a different set of schemas than the ones provided with the tool.

a. User specifies one or more specific product schemas to for validation.
b. Return to primary scenario at step 2.

<u>Alternative: User Specifies an Aggregate Product</u>
At step 1, the user wants to validate a set of products as listed in an aggregate product.

a. User specifies the aggregate product to validate.
b. Validate Tool examines the aggregate product for a list of products to validate.
c. Return to primary scenario at step 2.

3.4 Transform Product

TBD

3.5 Visualize Product

TBD

4.0 REQUIREMENTS

The architecture definition phase of the PDS 2010 project resulted in the decomposition of the system into several elements [3]. The Preparation Tools derive from the Archive Tools and User Tools/Services elements, which were derived from requirements 1.5 and 3.3 of the PDS Level 1, 2, and 3 Requirements document [1], respectively. The following level 3 requirements are relevant to this component:

1.5.1 PDS will provide tools to assist data producers in generating PDS compliant products
1.5.2 PDS will provide tools to assist data producers in validating products against PDS standards
3.3.2 PDS will provide a capability for opening and inspecting the contents (e.g. label, objects, groups) of any PDS compliant archival product
3.3.3 PDS will provide tools for translating archival products between selected formats
3.3.4 PDS will provide tools for translating archival products between selected coordinate systems
3.3.5 PDS will provide tools for visualizing selected archival products

In addition to the level 4 and 5 requirements specified below, the Preparation Tools must also comply with the general tool-based requirements found in the General System SRD document [5].

4.1 Level 4 Requirements

The level four requirements in PDS represent subsystem or component requirements at a high level. The following requirements pertain to the Preparation Tools:

L4.PRP.1 - The system shall provide a tool that assists users in the design of PDS product labels. (1.5.1)

L4.PRP.2 - The system shall provide a tool that assists users in the generation of PDS product labels. (1.5.1)

L4.PRP.3 - The system shall provide a tool that assists users in the validation of PDS products. (1.5.2)

L4.PRP.4 - The system shall provide a tool for transforming PDS products as follows: (3.3.3, 3.3.4)

a) From one supported data format to another
b) From one supported coordinate system to another

L4.PRP.5 - The system shall provide a tool for visualizing PDS products as follows: (3.3.2, 3.3.5)

a) Label content in text form
b) Data content in text or binary form
c) Data content in image form

4.2 Level 5 Requirements

The level five requirements in PDS represent subsystem or component requirements at a detailed level.

4.2.1 Design Tool

These requirements define the methods by which the Design Tool will assist the user in designing a schema for a PDS product label that is conformant with PDS standards. The following level 5 requirements pertain to the Design Tool:

L5.PRP.DE.1 - The tool shall initiate a design session as follows: (L4.PRP.1, UC 3.1)

a) From a blank schema
b) From an existing schema

L5.PRP.DE.2 - The tool shall accept the following as input for specifying a schema file: (L4.PRP.1, UC 3.1)

a) File specification
b) Uniform Resource Locator (URL)

L5.PRP.DE.3 - The tool shall facilitate modification of a schema file as follows: (L4.PRP.1, UC 3.1)

a) Remove optional content
b) Incorporate Node or Mission-specific content
c) Restrict the value set for a defined element
d) Restrict the range for a defined element

L5.PRP.DE.4 - The tool shall provide standard editing features as follows: (L4.PRP.1, UC 3.1)

a) Cut
b) Copy
c) Paste
d) Undo

L5.PRP.DE.5 - The tool shall indicate when a schema is not valid. (L4.PRP.1, UC 3.1)

> In the case of XML Schema, this includes a check for well-formed XML and conformance with its own schema. The definition of well-formed XML is detailed in the XML specification [7].

L5.PRP.DE.6 - The tool shall generate an XML instance file from a schema. (L4.PRP.1, UC 3.1)

The XML instance file represents an example product label conforming to the schema.

L5.PRP.DE.7 - The tool shall export the schema for use outside the tool. (L4.PRP.1, UC 3.1)

4.2.2 Generate Tool

TBD

4.2.3 Validate Tool

These requirements define the methods by which the Validate Tool will programmatically ascertain if a given product is PDS compliant (or "valid"). Typically, this means the product is well formed, complete, syntactically and semantically correct, and that it conforms to all applicable PDS standards. Validation of product labels in the PDS 2010 system relies heavily on the constraints of the XML grammar and the use of XML Schema to further constrain that grammar and the label content.

Also note that validation does not ensure that a product, specifically a data product, is scientifically accurate or useful to the planetary science community. Those issues are decided through the peer review process. The following level 5 requirements pertain to the Validate Tool:

L5.PRP.VA.1 - The tool shall accept the following as input for specifying the product(s) to be validated: (L4.PRP.3, UC 3.3)

a) File specification(s)
b) Directory specification(s)

Specifying a product is accomplished by providing a path to the product's label file. In PDS3 terminology, all products are detached meaning the label and data are contained in separate files.

L5.PRP.VA.2 - The tool shall traverse a directory tree and validate products discovered within that tree. (L4.PRP.3, UC 3.3)

L5.PRP.VA.3 - The tool shall validate aggregate products and all products referenced by such products. (L4.PRP.3, UC 3.3)

In PDS4 terminology, aggregate products are referred to as bundles and collections. Since bundles reference collections and collections reference individual products, specifying a bundle or collection product is another method for traversing a directory tree.

L5.PRP.VA.4 - The tool shall merge the contents of label fragments referenced by include elements with the contents of the parent label when validating a product. (L4.PRP.3, UC 3.3)

L5.PRP.VA.5 - The tool shall verify that a product label is well-formed XML. (L4.PRP.3, UC 3.3)

> The definition of well-formed XML is detailed in the XML specification [7] and is a standard check in open source and commercial parsers.

L5.PRP.VA.6 - The tool shall verify that a product label conforms to its associated schema file(s). (L4.PRP.3, UC 3.3)

> The schema defines the structure, content and, to some extent, the semantics of the product labels. The initial implementation of PDS4 utilizes XML Schema and XML Schema Definition (XSD) documents for this purpose.

L5.PRP.VA.7 - The tool shall accept the following as input for specifying the associated schema file(s): (L4.PRP.3, UC 3.3)

a) File specification(s)
b) Directory specification(s)

L5.PRP.VA.8 - The tool shall verify that a schema file is valid. (L4.PRP.3, UC 3.3)

> In the case of XML Schema, this includes a check for well-formed XML and conformance with its own schema. The definition of well-formed XML is detailed in the XML specification [7].

L5.PRP.VA.9 - The tool shall indicate the schema(s) utilized during validation. (L4.PRP.3, UC 3.3)

L5.PRP.VA.10 - The tool shall verify that a file exists when referenced from a product label. (L4.PRP.3, UC 3.3)

4.2.4 Transform Tool

TBD

4.2.5 Visualize Tool

TBD

5.0 DESIGN PHILOSOPHY, ASSUMPTIONS, AND CONSTRAINTS

The intent for this suite of tools is to develop them on top of a common set of functions and make them available in a Java-based library. These functions are then available to not only the tools but also to the applications and services within the system. The following diagram details where this common library resides in the layered architecture of the system:

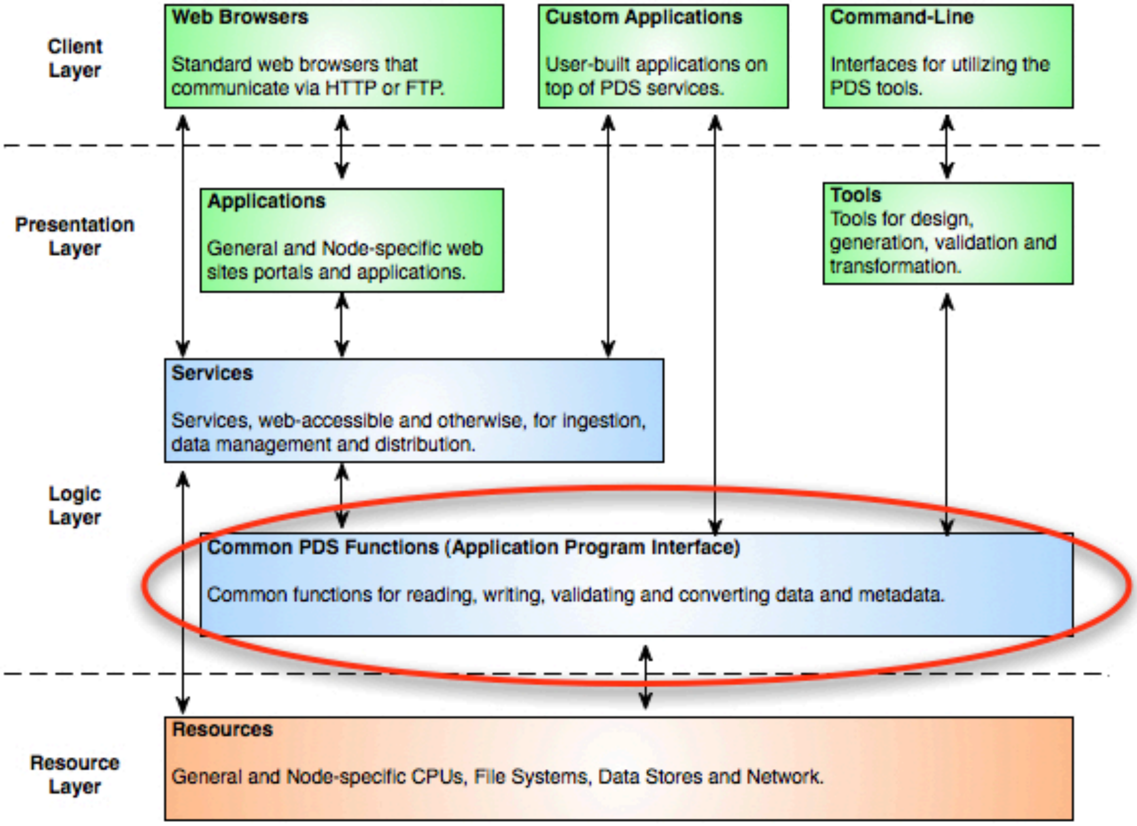
Client Layer
Web Browsers
Standard web browsers that communicate via HTTP or FTP.
Custom Applications
User-built applications on top of PDS services.
Command-Line
Interfaces for utilizing the PDS tools.
Presentation Layer
Applications
General and Node-specific web sites portals and applications.
Tools
Tools for design, generation, validation and transformation.
Logic Layer
Services
Services, web-accessible and otherwise, for ingestion, data management and distribution.
Common PDS Functions (Application Program Interface)
Common functions for reading, writing, validating and converting data and metadata.
Resource Layer
Resources
General and Node-specific CPUs, File Systems, Data Stores and Network.

Figure 3: Preparation Tools Approach

Although the majority of this library development will occur at the Engineering Node, portions of the library will be open for contributions from the PDS community. This is especially true when it comes to the transformation function. A few dozen possible transformations were identified in a PDS Node survey [6]. The Engineering Node will look to implement some of the highest priority transformations and invite the PDS community to contribute others.

6.0 ARCHITECTURAL DESIGN

The architectural design covers the component breakdown within the tools, external/internal interfaces and the associated data model.

6.1 Component Architecture

The following diagram details the architecture for the common function library:

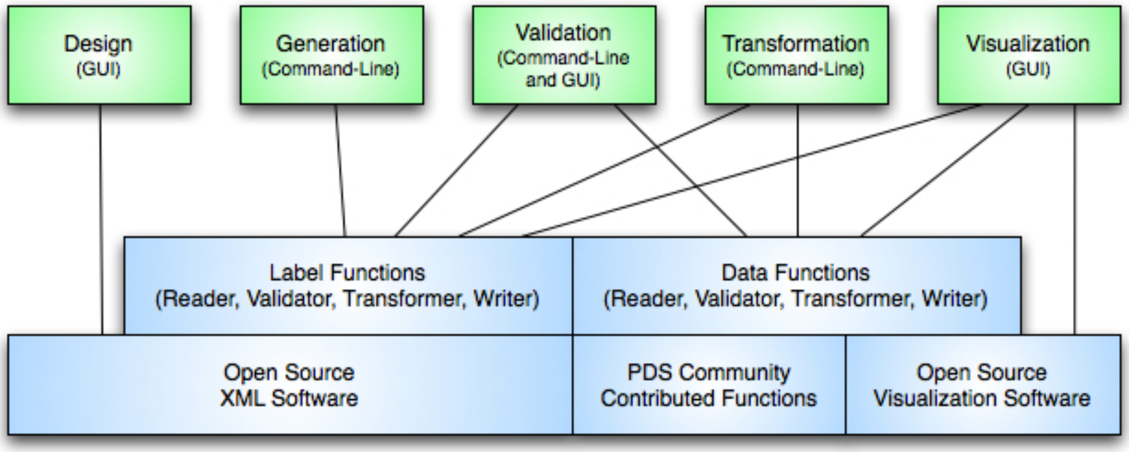

Figure 4: Preparation Tools Library Architecture

6.2 Data Model

The Preparation tools do not have an associated data model.

7.0 ANALYSIS

More information on this topic will be forthcoming in a subsequent version of this document.

8.0 IMPLEMENTATION

The PDS 2010 system is a phased implementation with increasing capabilities delivered in three planned builds. The builds are as follows:

- **Build 1** – This build consists of the Ingestion subsystem including the Security, Harvest, Registry (Inventory, Dictionary, Document, Service) and Report components along with the Data Provider tool suite.
- **Build 2** – This build consists of the Distribution subsystem including the Search and Monitor components along with a revised web site and general portal applications.
- **Build 3** – This build consists of enhanced user capabilities include the Order and Subscription components along with integration of Discipline Node applications and science services.

More information on this topic will be forthcoming in a subsequent version of this document.

9.0 DETAILED DESIGN

More information on this topic will be forthcoming in a subsequent version of this document.

APPENDIX A ACRONYMS

The following acronyms pertain to this document:

API	Application Programming Interface
COTS	Commercial Off-The-Shelf
GUI	Graphical User Interface
JPL	Jet Propulsion Laboratory
NASA	National Aeronautics and Space Administration
PDS	Planetary Data System
PDS3	Version 3.8 of the PDS Standards
PDS4	Version 4 of the PDS Standards
SDD	Software Design Document
SRD	Software Requirements Document
UC	Use Case
URL	Uniform Resource Locator
XML	Extensible Markup Language
XSD	XML Schema Definition