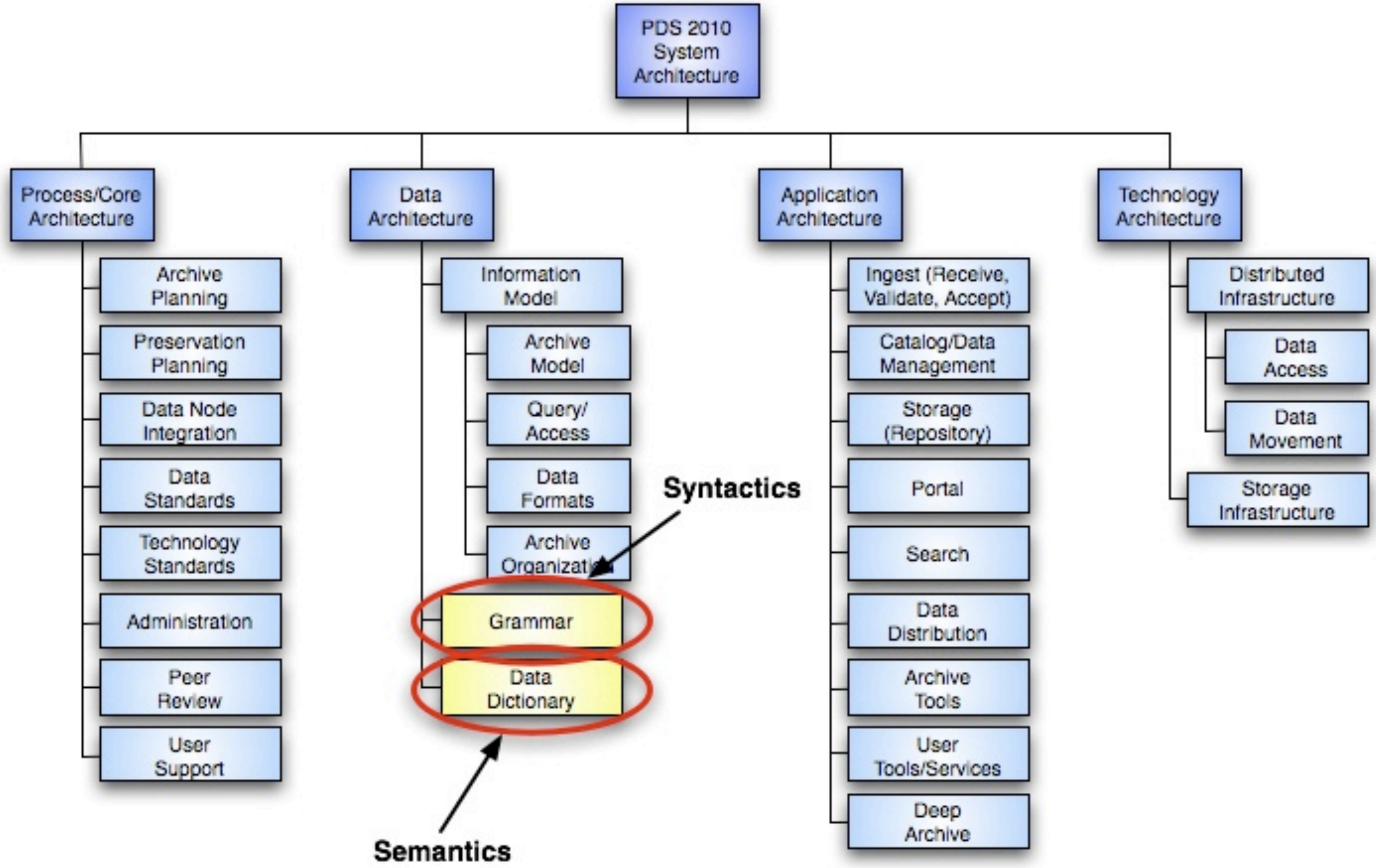
PDS Grammar in ODL, PVL, and XML

Paul Ramirez
Sean Hardman


PDS 2010 System Architecture
Process/Core Architecture
Archive Planning
Preservation Planning
Data Node Integration
Data Standards
Technology Standards
Administration
Peer Review
User Support
Data Architecture
Information Model
Archive Model
Query/ Access
Data Formats
Archive Organization
Grammar
Data Dictionary
Syntactics
Semantics
Application Architecture
Ingest (Receive, Validate, Accept)
Catalog/Data Management
Storage (Repository)
Portal
Search
Data Distribution
Archive Tools
User Tools/Services
Deep Archive
Technology Architecture
Distributed Infrastructure
Data Access
Data Movement
Storage Infrastructure

What is the PDS Grammar

- The PDS Grammar provides a standard syntax for capturing PDS Labels
  - It is a critical part of the data standards and architecture as it defines the format to exchange information
- We use the term “grammar” because in computer science it describes *formal* language for defining the syntactic rules and constraints for capturing and organizing data from computer languages to metadata
- Grammars generally provide syntactic rules, not semantic rules
  - PDS uses the data dictionary as a necessary component to validate the semantics
- There are implications of choosing a grammar beyond just syntax

Goals in Selecting the PDS Grammar

- Standards-based; wide adoption
- Support a syntax, that is both human and computer readable, which can “express” all aspects of the PDS data object description
- Common software libraries in a wide variety of languages available for PDS use; implementations that yield consistent results
- Supports consistent syntactic validation as well as support to perform semantic validation
- Simplify integration into pipelines for data providers
- Simplify data sharing and transport by being hardware and software agnostic
- Usable by science users

Who are the Contenders

- ODL – Object Description Language
  - Attribute, object, and group statements
  - Boils down to a key equals value with nesting
  - Governed by PDS
- PVL – Parameter Value Language
  - Parameter statements and aggregation blocks
  - Think ODL + semicolons
  - Governed by CCSDS
- XML – EXtensible Markup Language
  - Looks like HTML but its not
  - Tags with nesting, you must define your own tags
  - Meaning of the tags is left up to the designer
  - Many accompanying standards to address different needs XML Schema, XSL, XSLT, XPath, etc.
  - Governed by W3C (World Wide Web Consortium)

What is XML Schema?

- Constrains content in an XML document describing things such as elements, attributes, nesting, ordering, cardinality, data types, default values, and fixed values, etc.
- Supports extension and restriction of complex and simple types
- Supports and encourages reuse and extensions
- Supports namespaces
- Enables content generation and validation
- Became a W3C Recommendation May 2nd 2001

Understanding of …

	ODL	PVL	XML
Syntax • Typically captured as a grammar • Implies a format for the label • Enables one to read and write the format	PDS ODL Grammar	CCSDS PVL Grammar	W3C XML Specification
Semantic • Typically captured as a content constraint language • Provides meaning to the label • Enables one to express the model in terms of the format	PDS Data Dictionary Specification	CCSDS DEDSL Specification	W3C XML Schema Specification
Mapping • Typically captured in documents and examples • Describes how the model was expressed in the format	Standards Reference	Standards Reference	Standards Reference

Standards and Community

- ODL
  - We are building our own standards and community
  - Large maintenance costs for standards
  - Our community is relatively small so it can be hard to keep up with evolution and maintenance
- PVL
  - We are adopting standards whose community is slightly bigger than ours
  - No maintenance costs for standards
  - Semantics limited by DEDSL specification
- XML
  - We are adopting standards whose community is huge
  - No maintenance costs for standards
  - Need to understand best practices
  - Not all the accompanying standards are always needed or applicable

Support to …

	ODL	PVL	XML
Read and Write	Library we write and share	Existing codebase? What languages	Third party libraries. Many languages
Edit	Plain text editor	Plain text editor	Plain text editor and XML aware editors
Generate	PDS data dictionary + Code we write	DEDSL + code we write	XML Schema + Third party libraries
Transform	Code for each type of transformation	Code for each type of transformation	XSLT for each transformation. Portable and standard
Design	Application we write (LTDTool)	Application we write?	XML aware editor
Style	As is	As is	XSLT and CSS
Diffs and Merges	Code we write	Code we write	Third party tools
Validate Syntactically	PDS ODL Grammar + code we write	CCSDS PVL Grammar + code that is written?	W3C XML Specification + Third party libraries
Validate Semantically	PDS Data Dictionary Format + code we write	DEDSL + code that is written?	W3C Schema Specification + Third party libraries

Tools, Software, and Libraries

- ODL
  - We write a lot of code from scratch
  - Support in a limited number of languages
  - We have to coordinate to build a supporting libraries for consistent results
  - Long term higher development costs
- PVL
  - There are a some libraries to read and write PVL
  - Software support for validation of PVL against a data entity dictionary unknown
  - We will still need to write a lot of code from scratch
  - Long term higher development costs
- XML
  - Any code we write will like start from another library or framework
  - Many third party tools, software, and libraries
  - Long term lower development costs

User Impacts

	ODL or PVL	XML
Data Providers	Medium impact on upgrading existing software; long-term impact on maintaining skill set and tools for ODL/PVL	High short-term impact for existing software; long-term reduction in level of effort and recruiting individuals with knowledge of XML
Discipline Nodes	Medium impact on upgrading existing software; long-term impact on maintaining skill set and tools for ODL/PVL	High short-term impact in node tools; long-term reduction in level of effort to maintain tools
Data Users	Low impact except for data users who have tools that parse PDS labels	Low technical impact, particularly if PDS can display labels in multiple structures (PVL could be one of them)
Engineering Node	Medium impact on software development; long term sustaining costs in maintaining grammar libraries	High impact to software development; Medium impact to DEs; Long term reduction in standards management due to more stable model and data dictionary

Other Considerations

- Usability
  - The community is familiar with ODL
  - Training would be a requirement for XML
- Cost
  - Initial cost to translate software to XML “might” be higher, however, we believe keeping PDS4 data in the current ODL grammar is not an option either
  - Tool upgrades under a “fix ODL” plan would potentially be less
  - Long term costs to develop software and maintain standard would decrease under XML
- Integration …

Summarization of Analysis

- Build versus adopt standards
- Build versus adopt/adapt tools and libraries
- Buy in to a larger community versus create our own community
- Near term versus long term benefits and drawbacks
- Other design choices build off this decision so the sooner the consensus the better
- Make labels more accessible by everyone

Demo

- Sample ODL, PVL, and XML Label
- Sample PDS Dictionary, Data Entity Dictionary, and XML Schema
- XML Validation
- XML Label Generation
- XML Diffs and Merge
- XML to ODL Transformation

Options

- Option 1: Keep ODL “as is”
  - ODL is one of the major short comings in the current implementation of PDS that is driving inconsistency
  - We believe this is a non-starter option
- Option 2: PVL
  - PVL is managed by CCSDS
  - It is very close to ODL so the community would be familiar with it
  - Minimal library and application support
- Option 3: XML
  - A more modern approach, but different for the community
  - Would affect software and tools
  - Significant software support available

Recommendation

- Recommendation #1: Upgrade or replace PDS grammar in PDS4

- Recommendation #2: Perform PDS tech group survey to determine whether there is a "near" consensus on how to proceed

NOTE: From a "pure" engineering perspective, EN believes XML will improve PDS software development long term (cost, functionality, consistency). The major question is impact and usability. Ultimately, EN believes either ODL/PVL or XML will work.

Next Steps

- System Design group will perform a survey to capture node input on XML vs ODL/PVL to see if a proposal can be formulated for August MC

Backup

Terminology Primer

- What is XML?
  - Stands for EXtensible Markup Language
  - Is a markup language much like HTML
  - Was designed to carry data, not to display data
  - Tags are not predefined. You must define your own tags
  - Is designed to be self-descriptive
  - Is a W3C Recommendation February 10th, 1998
- What is XPath?
  - Is a syntax for defining parts of an XML document
  - Uses path expressions to navigate in XML documents
  - Contains a library of standard functions
  - Is a major element in XSLT

Source: http://www.w3schools.com/

```
<?xml version="1.0” encoding=“UTF-8”?>
<label urn=“dataset-product-1234”>
  <target_name>Mars</target_name>
  <data_set_id>test-1234</data_set_id>
  <table>
    <table_name>test table</table_name>
    <field>
      <field_number>1</field_number>
      <field_name>latitude</field_name>
    </field>
    <field>
      <field_number>2</field_number>
      <field_name>longitude</field_name>
    </field>
  </table>
</label>
```

```
/label/target_name/text()
/label/table/field[ field_number > 1]/field_name
//target_name
/label/table[ last()]
```

Terminology Primer (con’t)

- What is XSL?
  - Stands for EXtensible Stylesheet Language
  - Describes how the XML document should be displayed
  - Consists of XSLT, XPath, XSL-FO
  - Is a W3C Recommendation November 16th 1999
- What is XSLT?
  - Stands for XSL Transformations
  - Is the most important part of XSL
  - Transforms an XML document into another document
  - Uses XPath to navigate in XML documents
  - Is a W3C Recommendation November 16th 1999

Source: http://www.w3schools.com/

```
URN = dataset-product-1234
TARGET_NAME = Mars
DATA_SET_ID = test-1234

OBJECT = TABLE
  TABLE_NAME =  “test table”

  OBJECT = FIELD
    FIELD_NUMBER = 1
    FIELD_NAME = “latitude”
  END_OBJECT = FIELD

  OBJECT = FIELD
    FIELD_NUMBER = 2
    FIELD_NAME = “longitude”
  END_OBJECT = FIELD

END_OBJECT = TABLE
```

Terminology Primer (con’t)

- What is XML Schema?
  - Constrains content in an XML document describing things such as elements, attributes, nesting, ordering, cardinality, data types, default values, and fixed values (possibly missed something but you get the idea)
  - Supports extension and restriction of complex and simple types
  - Supports and encourages reuse and extensions
  - Supports namespaces
  - Became a W3C Recommendation May 2nd 2001

Source: http://www.w3schools.com/

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="label">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="target_name" type="xs:string"/>
      <xs:element name="data_set_id" type="xs:string"/>
      <xs:element name="table" type="table"/>
    </xs:sequence>
    <xs:attribute name="urn" type="xs:string"/>
  </xs:complexType>
</xs:element>

</xs:schema>
```