25 Years of Planetary Data Archiving with NASA's PDS: Lessons Learned the Hard Way

Anne C. Raugh[1] and John S. Hughes[2]

[1]University of Maryland, College Park, MD [2]Jet Propulsion Laboratory, Pasadena, CA

When the Planetary Data System undertook a redesign of its data format and metadata standards, it reviewed decisions made over two decades of archiving, keeping the successes and re-thinking some things that seemd like good ideas at the time, but proved problematic in the long term...

Good Idea

External Peer Review of All Archive Data

- Ensures quality and completeness
- Demonstrates usability
- Tests documentation

For PDS4

Rigorous, external peer-review will continue to be a major part of archive ingest. In addition, the PDS4 services will be tying the data into search and citation services like the ADS and DOI registries.

Good Idea

Manage Metadata Definitions in a Data Dictionary

- Organizes and formalizes definitions
- Provides a single official reference

For PDS4

Metadata has been divided into a namespace hierarchy, with stewardship in the hands of experts at the corresponding levels.

All namespace definitions are created using a canonical tool, and integrated into the complete Information Model to ensure interoperability.

Not-So-Good Idea

A Single, Centrally-Controlled Metadata Dictionary will Ensure Interoperability

- Institutional inertia makes modification onerous
- Discipline and mission metadata compromised by pressure to be broadly applicable
- Undermines interoperability by inviting local "interpretation" of existing definitions to avoid conflict and delay

For more about the PDS4 Standards:

https://pds.nasa.gov/pds4

Not-So-Good Idea

Any Sufficiently Well-Described Data Format Will Do

- Undermines long-term usability
- Impedes interoperability
- Obsolescence requires migration

For PDS4

Format standards optimized for long-term archiving were developed to ensure that archived data will never have to be "rescued" from becoming unreadable.

These standards support code sharing and third-party development, and minimize the cost of porting from the archive format to contemporary formats. The archive can remain stable and viable as processing formats and user needs evolve.

Good Idea

Identify and Support Recurring Data Structures

- Enables common processing
- Supports interoperability

Not-So-Good Idea

No Need for an Archiving-Specific Data Model

- Leads to unconstrained data structures
- No basis for code sharing or tool development
- Constant demand to migrate existing formats to new ones