Multimission Labels in PDS4, Part 1: Dictionaries R. G. Deen[1], C. M. De Cesare[1], J. H. Padams[1], S. S. Algermissen[1], N. T. Toole[1], S. R. Levoe[1], J. S. Hughes[1], P. M. Ramirez[1], [1]Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr, Pasadena, CA 91109, Bob.Deen@jpl.nasa.gov

Introduction: This is the first of a set of three abstracts [1,2] describing the authors' experience with creating multimission dictionaries, labels, and tools for PDS4. While the techniques made the initial mission (InSight in our case) a little harder, our results accommodate camera data for all recent Mars surface missions – InSight, Mars 2020, MSL, and MER, as well as two PDART tasks [3,4] that use MSL data. The authors were able to create valid labels for MSAM in just half a day. While the labels are not complete, they are valid and correct – and once the concepts new to MSL are added, future missions will be able to use those new concepts as well.

PDS4 is the Planetary Data System's new standard for archiving data [5]. While minimal PDS4 labels are not terribly complex, they also fail to capture the rich metadata that typically surrounds an observation. This metadata is often critical for understanding the context of an observation, the conditions under which it was acquired, and how the data was acquired.

Properly capturing all this metadata can lead to a very complex label. For example, the labels for raw products (EDRs) on the InSight mission are almost 650 lines long, with some derived products topping 850 lines [6].

One of the best ways to manage this complexity is to reuse dictionaries, concepts, tools, and even the entire label structure across missions. It is possible to create multimission labels with a high degree of commonality across missions, which reduces development cost and complexity significantly.

This abstract discusses the use of dictionaries in ways that support multimission reuse.

Multimission Dictionaries: It may be a surprise to some readers, but the InSight cameras do not use an InSight-specific mission dictionary. All the concepts InSight needs are in multimission, reusable dictionaries [7].

How is this possible? Primarily, by taking a broad view of common concepts. Things do not have to be *exactly* the same to warrant using the same class or attribute. This point can be illustrated with several examples, below. It is important to note that the PDS4 Information Model was designed to allow this – to infinitely extend the information a mission can specify without modifying the pre-existing schemas [8].

Reusable Attributes: Consider the attribute `download_priority`. Pretty much any mission using a file-based telemetry system (like the recent Mars landed missions) will have a priority for downlinking a given product. On InSight, the priorities go from 1-6. On the more complicated MSL, the values go from 0-101. That difference is not meaningful to the general concept of download priority. It would be wasteful to have separate `insight:download_priority` and `msl:download_priority` attributes when they really mean the same thing. The actual data range does not matter (and is thus not defined in the data dictionary); what is important is the concept that there is a priority assigned, and that lower numbers mean higher priority. (See the Property Maps abstract, part 2 of this series [1] regarding documentation of the difference).

Another example of this is in the `Image_Compression_Parameters` class in the Imaging dictionary. This class contains a `compression_type` attribute which says what the specific compression is, and child classes containing parameters specific to that type. However, certain concepts are abstracted out to the main compression class, containing generic information about the compression. For example, `compression_class` says whether it is lossy or lossless, `compression_rate` and `compression_ratio` describe how much compression there is, and `compression_quality` gives an indication (on a 0-1 scale) of how good the compression is.

The quality is worth some discussion. Each compression type measures quality in a different way, and there is for example a `jpeg_quality` under `JPEG_Parameters` specifically stating this. However, the general `compression_quality` field provides a generic abstraction of quality, normalized to the 0-1 scale. This allows programs or users with no knowledge of the specific compression type to make at least a qualitative assessment of which of two images is likely to be better.

Reusable Classes: Most spacecraft have moving parts: arms, masts, antennae, wheels, etc. Rather than defining a separate set of classes for each of these, the Geometry dictionary defines an `Articulation_Device_Parameters` class. This is a completely reusable class that is adaptable to all of those moving parts, defining joint angles, states, temperatures, motor counts, and other common attributes. Rather than "`rocker_bogie_angle`" or "`elbow_joint_angle`", an array of joint angles and corresponding names are used. This allows any angle to be defined by the data provider without having to modify the dictionary. For example:

```
<geom:Device_Angle>
   <geom:Device_Angle_Index>
      <geom:index_name>
         AZIMUTH-MEASURED
      </geom:index_name>
      <geom:index_value_angle>
         3.618562
      </geom:index_value_angle>
   </geom:Device_Angle_Index>
   <geom:Device_Angle_Index>
      <geom:index_name>
         ELEVATION-MEASURED
      </geom:index_name>
      <geom:index_value_angle>
         1.635858
      </geom:index_value_angle>
   </geom:Device_Angle_Index>
</geom:Device_Angle>
```

While this may look harder to read at first glance, the regularity of the array structure actually makes it easier to interpret in the long run. Software need be written only once, and it can understand – at least at a basic level – any articulation device. As an aside, this is similar to how the PDS3 articulation device labels are structured as well.

Another example of this concept is the `Coordinate_Space_Definition` class in Geometry, which can be used to define any coordinate space (such as arm, rover, site, or camera), without needing to create classes specific to each one.

Reusable Dictionaries: PDS4 defines three levels of dictionaries, based primarily on governance: common, discipline, and mission. PDS itself keeps tight rein on common, PDS nodes manage discipline dictionaries, and data providers manage the mission dictionaries.

One of the key concepts here is that mission dictionaries *need not be tied to a mission*! They can in fact be abstracted to cover a class of missions, and thus be reused.

A primary example of this is the `msn_surface` dictionary. All of the recent NASA Mars surface missions have a similar command and telemetry architecture. Commands are combined into sequences, downlink products are identified by APIDs (Application IDs), etc. These concepts are specific to the Mars surface missions – yet they are shared across all such missions. It does not make sense to repeat them over and over again in different mission dictionaries. Instead, a "mission" dictionary called `msn_surface` (mission surface) was created [7]. It has classes and attributes allowing this command and telemetry structure to be described and documented. It is reusable across all of the relevant missions, and also is reusable across all disciplines – not only for imaging but for any instrument data from these missions.

In a similar vein, `img_surface` contains items that are too tied to surface operations to make sense in the main Imaging dictionary, yet are shared across all of these surface missions.

As a final example, consider the set of 5 cameras made for surface missions by Malin Space Science Systems (MSSS): Mastcam (MSL), MAHLI (MSL), MARDI (MSL), Mastcam-Z (M2020), and SHERLOC-WATSON (M2020). All five cameras share a common heritage, and in fact very similar electronics. They all have an identical "mini-header" that is created by the camera itself (rather than the host rover). The information in this header is very specific to the MSSS cameras. Rather than repeating this information in both the MSL and M2020 mission dictionaries, a single dictionary, "`msss_cam_mh`", was created to hold this shared mini-header information.

Conclusion: Contrary to what some may believe, multimission PDS4 labels are well within reach. With a little attention up front, it is entirely possible to write dictionaries, generate labels, and document them in ways that are highly reusable across multiple missions. Reusing dictionaries, classes, and attributes will help future data mining efforts, as similar concepts are called the same thing across multiple missions.

Acknowledgements: This research is being performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA.

References:

[1] Deen, R.G. et al (2019), 4th PDW, Abstract #7051
[2] Deen, R.G. et al (2019), 4th PDW, Abstract #7052
[3] Deen, R.G. et al (2018), 49th LPSC, Abstract #2332
[4] Bell, J.F. III et al (2018), 49th LPSC, Abstract #1068
[5] https://pds.nasa.gov/datastandards/about/
[6] Deen, R.G. et al (2018), InSight Camera SIS, PDS
[7] https://pds.nasa.gov/datastandards/schema/released/
[8] https://pds.nasa.gov/datastandards/documents/sr/v1/StdRef_1.11.0.pdf