Multimission Labels in PDS4, Part 2: Property Maps 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 second of a set of three abstracts [1,2] describing the authors' experience with creating multimission dictionaries, labels, and tools for PDS4. See part 1 [1] for a full introduction.

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

Property Maps: In one of the examples presented part 1 of this series [1], it was stated that the actual range for downlink_priority didn't matter in a multimission sense. That's true, but it does matter generally. How can we keep track of the fact that InSight priority is 1-6 but MSL is 0-101?

For that matter, how can we track the association between PDS4 attributes and classes ("items" here), and PDS3 keywords?

These issues and more are solved by a PDS4 concept called Property Maps.

Property maps are additional bits of information that are attached to attributes and classes. They are essentially metadata for the item.

A convenient way to think of these is in terms of a spreadsheet. If there is a row for each attribute or class, the columns represent Property Maps.

Although the governance model for property maps has not been formally defined, it is expected that it will be very different from dictionaries. Although property maps are associated with items across all dictionaries, they should be controlled not by the dictionary curator but rather by the "owner" of that property map. In other words, when an entity defines a "column" in the spreadsheet, they should own that column, and can put whatever information in it they want – across any dictionary in all of PDS4.

Examples of Property Maps: One of the most important property maps are "mission-specific definitional nuances". These provide the details for how a specific item is used for that mission, that augments or enhances the multimission definition.

So for the downlink_priority example, the InSight nuance would say that the range is 1-6 while the MSL nuance would say it is 0-101.

As another example, sample_fov is computed differently on MER than it is on MSL and InSight. The definitional nuances can specify exactly how it is computed for that mission – without negating the value of having the field of view available generically.

As a final example of nuances, both InSight and MSL have exposure_duration_count, which gives the number of clock ticks for an exposure (there's also the more useful exposure_duration in seconds). InSight uses 6.21 msec/DN, while MSL uses 5.12 msec/DN for engineering cameras and 0.1 msec/DN for the MMM cameras. These differences are all specified in the definitional nuances.

Another property map specifies the VICAR or PDS3 keyword associated with the item. This is used for mapping values between VICAR/PDS3 and PDS4.

Additional property maps can be defined for any other perceived need. Several will be defined based on the needs of the table, described below.

Below is a highly simplified example showing a JSON version of the download_priority property map:

```
"PropertyMapDictionary": [ {
  "PropertyMap": {
    "model_object_id": "0001_NASA_PDS_1.
      msn_surface.Telemetry_Information.
      msn_surface.download_priority" ,
    "propertyMapEntryList": [
      {"PropertyMapEntry": {
        "property_name": "VICAR" ,
        "property_value":
          "TELEMETRY.DOWNLOAD_PRIORITY" } },
      {"PropertyMapEntry": {
        "property_name": "MSL" ,
        "property_value":
          "Values are 0-101 for MSL." } },
      {"PropertyMapEntry": {
        "property_name": "NSYT" ,
        "property_value":
          "Values are 1-6 for InSight."}}]}}]
```

Documentation Table for SIS: How do we go about documenting the labels? Science data users cannot and should not be expected to read the data dictionaries directly. Plus, a lot of the descriptions of keywords, especially in the PDS core, are so generic as to lose meaning.

In the Software Interface Specification (SIS) documents for MER, MSL, and Phoenix [3,4,5], the VICAR/PDS3 keywords were listed in a table in an appendix. Every keyword was listed, along with its definition, data type, valid values, etc. This is an invaluable resource for users, as it provides a single, easy to use place to get information about what a keyword means.

However, this table had to be maintained by hand, as a rather unwieldy Microsoft Word table. It takes a lot of effort to create, and to keep it up to date.

We have solved this issue with an automated table creation tool. This tool creates a table similar to the PDS3 SIS tables, listing every class or keyword that is used in the label, and the associated information for that keyword. All information is pulled from the data dictionaries and property maps, so it is guaranteed to be up to date with PDS. The complete XPath to the keyword is shown, and it is completely hyperlinked, so users can easily click on any parent (or child) and see the definition for that item. Finally, because it is automated, the table can be created multiple ways, for example sorted by PDS4 attribute/class or sorted by VICAR/PDS3 keyword.

Importantly, this table contains *only* the items of interest for the mission. In order to do this, the tool accepts a set of sample PDS4 labels for the mission. It uses these to decide which classes and keywords to include. Including the entire dictionary without this culling would be unwieldy for users, as it would not be clear which entries were important and which were not. It would also cause a false sense of "this should be in there", when the item is not in fact relevant. For InSight, we included calibration and ancillary data in this example set, so the table documents any label item that might be seen in those data sets as well.

This is an entirely data-driven process – everything is in the data dictionaries, including the property maps. Obviously the property maps containing the mission's definitional nuance and VICAR/PDS3 keyword mappings are important. Some missions (e.g. Mars 2020) want additional columns in the table, such as where in the telemetry the information comes from. These columns will be stored as additional property maps, so they can also be included in the table.

Basically, property maps provide the flexibility to add ancillary information to the PDS data model that are not properly part of the model per se.

An example of a few entries from this table is below. See the InSight SIS [6] for the full table.

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.

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 #7050
[2] Deen, R.G. et al (2019), 4th PDW, Abstract #7052
[3] Chen, A. et al (2014), MER Camera SIS, PDS
[4] Alexander, D. and Deen, R. (2018), MSL Camera SIS, PDS
[5] Alexander, D. et al (2009), Phoenix Camera SIS, PDS
[6] Deen, R.G. et al (2018), InSight Camera SIS, PDS

Dictionary: **PDS4 Keyword** *VICAR Property.* ***VICAR Keyword***	**General Definition** *InSight-Specific Information*	**Xpath** Valid Values (attribute) *Children (class)*	Data Type *Units*
geom:**Derived_Geometry**	The Derived_Geometry class is a container for surface based observations (lander or rover). It is used to provide some geometric quantities relative to a specific Reference Coordinate Space. **InSight Specific:** *On InSight, a Derived_Geometry class exists for both Site and Lander frame.*	**/Product_Observational/Observation_Area/Discipline_Area/Geometry/Geometry_Lander/Derived_Geometry[*]** 1) *geom:emission_angle* 2) *geom:Coordinate_Space_Reference* 3) *start_azimuth* 4) *stop_azimuth* 5) instrument_azimuth 6) instrument_elevation 7) solar_azimuth 8) solar_elevation	
msn_surface: **download_priority** *TELEMETRY.* ***DOWNLOAD_PRIORITY***	The download_priority attribute specifies which data to downlink/transmit, based on order of importance. The lower numerical priority (higher-ranked number) data products are transmitted before higher numerical priority (lower-ranked number) data products. For example, an image with a downlink priority of 1 will be transmitted before an image with a downlink priority of 6. Value of 0 specifies use of on-board default. **InSight Specific:** *Values are 1-6 for InSight.*	**/Product_Observational/Observation_Area/Mission_Area/Surface_Mission_Parameters/Telemetry_Information/download_priority**	ASCII_NonNegative_Integer
img: **exposure_duration_count** *OBSERVATION_REQUEST_PARMS.* ***EXPOSURE_DURATION_COUNT***	The exposure_duration_count attribute specifies the value, in raw counts, for the amount of time the instrument sensor was gathering light from the scene, such as between opening and closing of a shutter, or between flushing and readout of a CCD. This is the raw count either commanded or taken directly from telemetry as reported by the spacecraft. This attribute is the same as the exposure_duration but in DN counts versus time, and the translation of exposure_duration_count to exposure_duration will differ by mission. The attribute can be specified in the context of both Imaging_Instrument_State_Parameters (actual value) and Command_Parameters (commanded value). Both commanded and actual because it's possible for the actual to not match the commanded. For example the exposure might fault out early, or there might be a deadband (for example, pointing backlash) where changes in the input do not actually affect the output. **InSight Specific:** *For InSight, the factor is 6.21 msec/DN.*	**/Product_Observational/Observation_Area/Discipline_Area/Imaging/Command_Parameters/Exposure_Parameters/exposure_duration_count** **/Product_Observational/Observation_Area/Discipline_Area/Imaging/Image_Product_Information/Exposure_Parameters/exposure_duration_count**	ASCII_NonNegative_Integer
img:**sample_fov** *INSTRUMENT_STATE_PARMS.* ***AZIMUTH_FOV*** *INSTRUMENT_STATE_PARMS.* ***AZIMUTH_FOV__UNIT***	The sample_fov attribute specifies the angular measure of the field of view of an imaged scene, as measured in the image sample direction (generally horizontal). **InSight Specific:** *Computed by projecting rays from the left and right edges of the image at the center through the camera model, and computing the angle subtended by those rays.*	**/Product_Observational/Observation_Area/Discipline_Area/Imaging/Imaging_Instrument_State_Parameters/sample_fov**	ASCII_Real *Units_of_Angle*