The Science Data Model and its Standardization (OBSOLETE)

OBSOLETE

This is now in DMTN-153

1. DPDD

The DPDD (LSE-163) represents an "idealized" data model that cannot directly be used as a recipe for the construction of a physical database schema and is not a full, precise definition of a data model. It should be treated as a requirements document for the data model and schema rather than itself defining them.

We will move toward deriving the document form of the DPDD - at least the tabular portions that actually define the data products to be produced - from an authoritative input that is more formally structured. We expect this to be a YAML document.

What this formal source document will provide beyond the existing LaTeX content is, at a minimum:

- a formal string identifier for each data item ("DPDD Identifier"), most likely in the form of a hierarchical identifier, which can be used to refer to it in multiple contexts; and
- a program-friendly structure that facilitates the transformation of the DPDD content into a variety of formats, now and in the future.

This will permit it to be used, for instance, to generate a deep-linkable HTML form of the DPDD, or to deliver per-data-item documentation in some other form that facilitates linking to it from the components of the Science Platform.

As the architecture and implementation work described in this note proceeds, we will look at whether the DPDD YAML source should also use a more formally defined vocabulary for the definition of data types and units, similarly to what will be used in the Science Data Model (see below).

As will be seen elsewhere in this document, we expect to link the DPDD with additional metadata defining the physical data model for the released data products. However, to facilitate configuration management, the DPDD source document will contain no back-links to the data model definition, so that the DPDD repo in Github can be placed under configuration control on its own. We expect the DPDD itself to be changed infrequently, and to continue to be under the higher, project-wide level of change control that it currently is, while the associated data model may evolve more often as construction proceeds and algorithmic choices are further resolved.

2. The Science Data Model

The Science Data Model (SDM) specification is a machine-readable specification for the physical data model for the publicly released science data products of the LSST Project.

The SDM is a realization of the DPDD; thus, viewing the DPDD as a requirements specification, the SDM is to be viewed as a design specification that must contain elements that formally satisfy every data item called for in the DPDD. The SDM may contain additional elements beyond those required by the DPDD. These may arise from more detailed algorithmic choices made in considering how to meet the DPDD's requirements, from considerations of meeting external standards such as IVOA data models or the CAOM2 data model, or from engineering considerations such as choices of how to map the DPDD onto a formal relational data model and whether it should be normalized or not.

The SDM defines the data products to be served to science users through the LSST Science Platform (LSP), and it is therefore a requirement that the data model be realizable in the underlying data storage systems (e.g., in Qserv, where applicable) and handled by the LSP's data services (e.g., TAP/ADQL and the Butler) and by its user-facing components such as the Portal Aspect.

The SDM must contain sufficient information for a physical SQL schema definition to be derived from it, given a choice of SQL flavor (e.g., MariaDB, Oracle, PostgreSQL).

The SDM specification will be written in YAML. The SDM must contain information that itemizes how it satisfies the DPDD requirements for the content of the data model. For example, each SDM element that realizes a data item from the DPDD might contain a field that references the appropriate DPDD Identifier.

Each element of the SDM must be described by a unique identifier ("SDM Identifier") that can be used programmatically in applications that consume the SDM YAML definition. We expect that the "leaf nodes" in the name space of these identifiers will correspond directly to column names in generated database tables; it seems unnecessary to have yet another layer of indirection at this level. Higher levels in the name space may not correspond exactly to database and/or table names, however; this has yet to be determined precisely.

Software support will be provided for verifying that the SDM provides coverage for all the data items defined in the DPDD. This should ultimately be subject to verification as part of a CI process. In order to facilitate the introduction of the SDM language and software into DM, a transitional period should be supported during which a partially complete SDM can be used without triggering constant CI failures.

The design of the SDM and its specification language should address the need to map the physical data model that derives from it to the *catalog.schema.table* name space of the ADQL 2.0 and TAP standards. (Bear in mind that the way the term "catalog" is used in this context **does not** correspond to the intuitive sense of "astronomical catalog".)

In addition to being able to be used to construct a database schema, the SDM specification must also include the information required to provide IVOA-oriented table and column metadata in query responses. The system must support:

- the assignment of UCDs from the UCD1+ standard to all column-like fields in the SDM;

- the assignment of IVOA "utypes", where they are useful and taken from either a external standard vocabulary or an LSST-provided vocabulary, to column-like fields as well as, potentially, to tables; and
- the definition of "field groups" in the VOTable sense for related data items, such as a quantity and its uncertainty(ies);

and it should also support:

- mapping of the Science Data Model to externally provided or LSST-defined VO-DML for part or all of its content (assuming that the VO-DML effort remains on track to produce something useful to us).

The SDM specification must include a data type definition for each column-like field. This data type is intended to be used to derive several downstream data types involved in the physical instantiation and service of the data:

- SQL database types consistent with the variety of actual database software used in LSST, which will include at least MariaDB and Oracle, may include PostgreSQL depending on the progress of PPDB development, and should also include the HyperSQL in-memory database used in the Portal Aspect of the LSP (in Firefly);
- SQL92-based database types for use in the ADQL context;
- VOTable data types;
- text-based data formats for use when data model elements are represented as text, e.g., in the VOTable TABLEDATA format or in CSV;
- Python data types; and
- data types usable in the Parquet and Apache Arrow cross-platform ecosystem, as well as in Apache Avro.

Many of these mappings are already externally defined, of course. It may be necessary for the SDM specification to allow for an override of the "natural" mappings between these target types, so this should be kept in mind in the design, but no specific instance of a need for this has yet been documented.

Brian Van Klaveren has been working on a platform-independent vocabulary for types that should be very useful in this role.

The SDM specification must contain sufficient information to be able to derive the foreign-key relationships between tables. It should also allow:

- the definition of specific columns as required to be indexed (though downstream database implementations may be permitted to add indexes to additional columns for implementation-time performance optimization); and
- the definition of the columns and key relationships required to support the Qserv architecture, e.g., the designation of which ra/decl values in a table are the "primary" ones to be used for the spatial partitioning of the table.

The SDM specification should be usable to support the mapping of sectors of the Science Data Model to external data models such as ObsCore and CAOM2. In many cases it may be possible for the SDM itself to include data elements that directly correspond to required elements of these data models; in other cases some conversion may be required. Both scenarios should be supportable.

The actual mapping to external data models may be part of the SDM specification itself or it may be deferred to additional specification file(s).

The SDM specification "technology" may be used not only to define **the** LSST data model (for a Data Release, or for the instantiation of the Prompt Processing system and database), but also in narrower contexts to define smaller data models for use in science validation and to support the use of the Science Platform tools during commissioning, for instance.

Concrete use cases that must be supported by the SDM specification and associated software:

- generation of executable physical SQL schema, comparable to what is now in the "cat" repository on Github;
- definition of the alert data model and its associated Apache Avro `.avsc` schema;
- population of the TAP_SCHEMA tables required by the TAP 1.1 standard with the information needed to describe the LSST data model, including foreign-key relationships;
- generation of the VOTable headers for query results from the LSST TAP service;
- population of the tables or, where appropriate, definition of the views mandated by the external standards LSST supports, e.g., the "ivoa.ObsCore" table/view required by the ObsTAP portion of the ObsCore standard, or the standard tables of the CAOM2 data model; and
- creation of Parquet files representing the Science Data Model and usable for ingest into the databases.

The SDM-to-DPDD mapping information may be useful as part of the documentation of the Science Data Model that LSST exposes to users, to facilitate their understanding of how the SDM corresponds to the DPDD. To this end, for instance, it may be appropriate to include in the TAP_SCHEMA tables an additional column (something permitted by the standard) that provides, where appropriate, the DPDD Identifier for a data item.

3. Mapping from Science Pipelines outputs to the Science Data Model - "Standardization"

The SDM specification does not directly constrain the data models produced by the Science Pipelines code's Tasks and PipelineTasks. In some cases, for instance, the Science Data Model will envision a table composed of information assembled from the outputs of several pipeline stages.

We therefore envision a "standardization" component of both the Data Release Production system and the Prompt Processing system that takes the scientific outputs of the pipelines and assembles the data required to load the DRP and/or PP databases.

The mapping may be trivial in most cases, simply involving copying over the content of an attribute from an afw_table object to a data element in the SDM, or in some cases a conversion may be involved, such as a change of units or coordinate system.

The extent to which this process can be implemented in a table-driven way as a mapping from designated afw_table Butler dataset types and attributes to designate SDM Identifiers is still under investigation.

The current plan envisions implementing the standardization as processing steps that read afw_table-formatted FITS files and generate Parquet files in the SDM form. As a large-scale automated process required as part of DRP, this might naturally be implemented as a set of PipelineTask components in order to allow the workflow tooling of the production system to be applied.

The system should provide for a verification step that compares the Parquet outputs to the SDM specification and verifies that they are compatible. This verification could then be part of the CI system for the standardization code as well as being run as a final stage of the actual production system.

Colin Slater and Yusra AlSayyad have in recent months been working on prototyping elements of the above system.

4. Data model metadata for Science Pipelines outputs

The SDM specification will fully enumerate the metadata that LSST provides to its users describing the Science Data Model (e.g., its UCDs). This metadata is used, in part, to enable intelligent behavior in the Science Platform components when serving data to users.

However, it may be useful in a number of cases to provide some of this metadata upstream, at the point of creation of the Science Pipelines outputs. There are two key motivations for this. First, in many cases the authors of the Tasks and PipelineTasks used in production will be the best placed to define the semantics of a particular value produced by the science code.

Second, there are numerous use cases where enhancing the usability of the direct Science Pipelines outputs in the Science Platform will be valuable, e.g., in the use of afw.display to visualize images and catalog data. This arises in the context of internal project activities during construction, commissioning, science validation, and the Operations-era maintenance of the DM software, and it also arises in supporting external science users in developing their own science workflows built out of Science Platform components. While we aspire to make the SDM-standardization system **available** for science users to define and create User Generated (Level 3) data products, we do not wish to **require** the use of this extra layer before useful behaviors from the Science Platform components can be obtained.

By way of example, the ability to tag a table of image metadata from the pipeline software with a minimal set of ObsCore information enables a substantially richer experience when browsing that table with the Firefly components. Similarly, tagging values representing times as such permits them to be displayed easily either as floating-point MJDs or as human-readable times; tagging data uncertainties as such allows them to be used to automatically plot error bars.

We therefore recommend that afw_table be extended to support adding certain types of column metadata such as UCDs, utypes, and field groups at the point of creation of the table. For data elements in the Science Pipelines output for which this information is provided, the SDM Standardization process can validate that the mapping it performs reflects the compatibility of this information between its inputs and outputs.

The Standardization process can add additional metadata. For instance, ra/decl columns in the Science Pipelines outputs may be tagged with "pos.eq.ra" and "pos.eq.dec" UCDs. For any Object there may be many such measurements, e.g., in different bands, for model fits and for simple centroids, etc.; however, in the Science Data Model for an Object, one particular set of coordinates will be specified as "primary" for use in Qserv partitioning as well as for default visualizations. In this case, the output of the SDM Standardization process would have the more qualified, but compatible, UCDs "pos.eq.ra;meta.main" and "pos.eq.dec;meta.main".