

UTYPES and UFIs

Jonathan McDowell

Data Model fields

- A data model consists of a bunch of information (attributes) and a structure organizing them (classes)
- If we ignore the structure, we get a long checklist of all the information data providers need or might need to describe their data
- Each piece of information and each grouping (class) is called a data model field, and is given a name which we call a utype:
 - Example: `Spectrum.Target.Name`

UTYPE syntax

- Basic syntax is tokens separated by periods as **token.token.token**
- Leftmost is the highest level containing class, rightmost is the actual item in question
- Implies a “has-a” hierarchy: Spectrum.Target has a field whose 'simple utype' is Name and whose 'fully qualified utype' is Spectrum.Target.Name
- Case-insensitive, but recommend CamelCase style

Utypes

- A utype is unique within the MODEL; it defines a class or attribute.
- But, the same utype can appear multiple times in a single INSTANCE document.
- When you find a piece of data in the document, the utype tells you what it is – what role it plays in the structure. For example: “axis units”. There can be several axes, each with axis units.

UFIs

- VOQL team identifies a need to locate information within a document – there must be a unique address for each piece of information
- I call such a unique address a UFI (Unique Field Identifier).
- UFIs cannot be utypes since utypes are not unique. But, we can build UFIs from utypes and ucds and attribute values.

Can UFIs be simple strings?

- A Micol suggested every document be describable by simple strings: a different predefined UFI for each piece of info.
- This limits severely the complexity of documents that you can create – works ok for a document with very fixed structure (but then, why not use FITS?). If document has variable structure (linked lists, arbitrary number of axes, arrays of data structures whose size is not predefined) then this approach just does not work.

Querying Utypes

- Francois Bonnarel has argued for a query language form in which (similar to XPATH?) one could distinguish multiple instances of the same utype within a document by specifying the values of attributes of the relevant element or containing elements, e.g.
 - utype=token1.token2[attribute1=val1].token3.token4