

Collecting Utype requirements - Splinter meeting during ADASS conference - 2011, November 8

Participants: D.Tody, F.Bonnarel, L.Michel, O.Laurino, M.Louys, A.Micol

Goal : start gathering of requirements for Utypes IVOA definition as indicated on the utype list recently.

Reminder:

The Utype is a string defined following some rules in the context of a data model published in the IVOA.

Utype definitions are issued together with a data model and published in the IVOA.

We identified several requirements the Utypes should comply for:

Requirements:

1. The Utype prefix (also named "namespace") should be defined for each data model
2. The Utype definition should support data model extensibility

We identified several cases where some metadata handled in an archive or an application are partly covered by an IVOA data model and some other metadata are not covered at all by any IVOA model. Then the data model could be extended for a specific purpose, on a case to case basis. The data provider or application developer should find a possibility to reuse the data model parts and add new features. These customized features should be added in a customized version of the model , by inheritance as much as possible , and not overlap with existing properties.

The developer could define new Utypes for the customized data model fields. New utypes should be built with the same procedure as for recommended data model (syntax, object inheritance, symbols).

They can be distinguished from the original model by a customized data model prefix (namespace).

f.i.: sed : Data.FluxAxis.value derived in sedNED : Data.FluxAxis.publishedValue

3. Utypes can be used as parameters in a query language like PQL
4. Utypes should be compatible with data model re-usability.

This may imply some properties on the string syntax and on the meaning interpretation:

- Unique meaning in the IVOA domain
- Do we allow for derived meanings?

The data model re-use should be explicit in the IVOA documentation, using the inheritance property.

Tools understanding the parent model should be able to manage their metadata without knowing more about the children model and added features.

Need to describe inheritance of models in a computer readable way.

- re-use when defining a new model :

e.g: TimeseriesDM inherits SpectralDM

- re_use when gathering metadata covered by 2 models

e.g.: source catalogs with photometry capabilities

see <http://www.ivoa.net/internal/IVOA/PhotometryDataModel/NOTE-PPDMDesc-0.2-20110512.pdf>

use

photdm:PhotometryFilter.SpectralAxis.Coverage.Location.Value

and spec:Data.FluxAxis.value