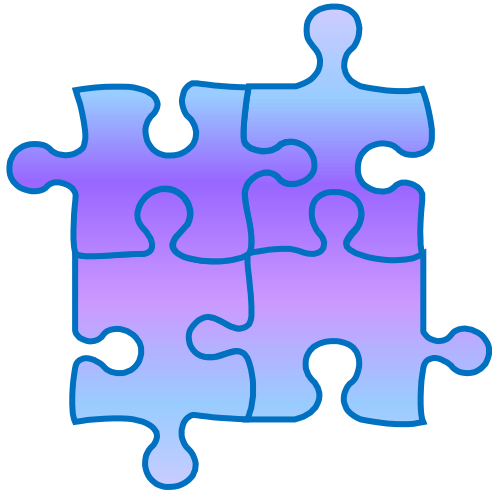


Utypes : WD status

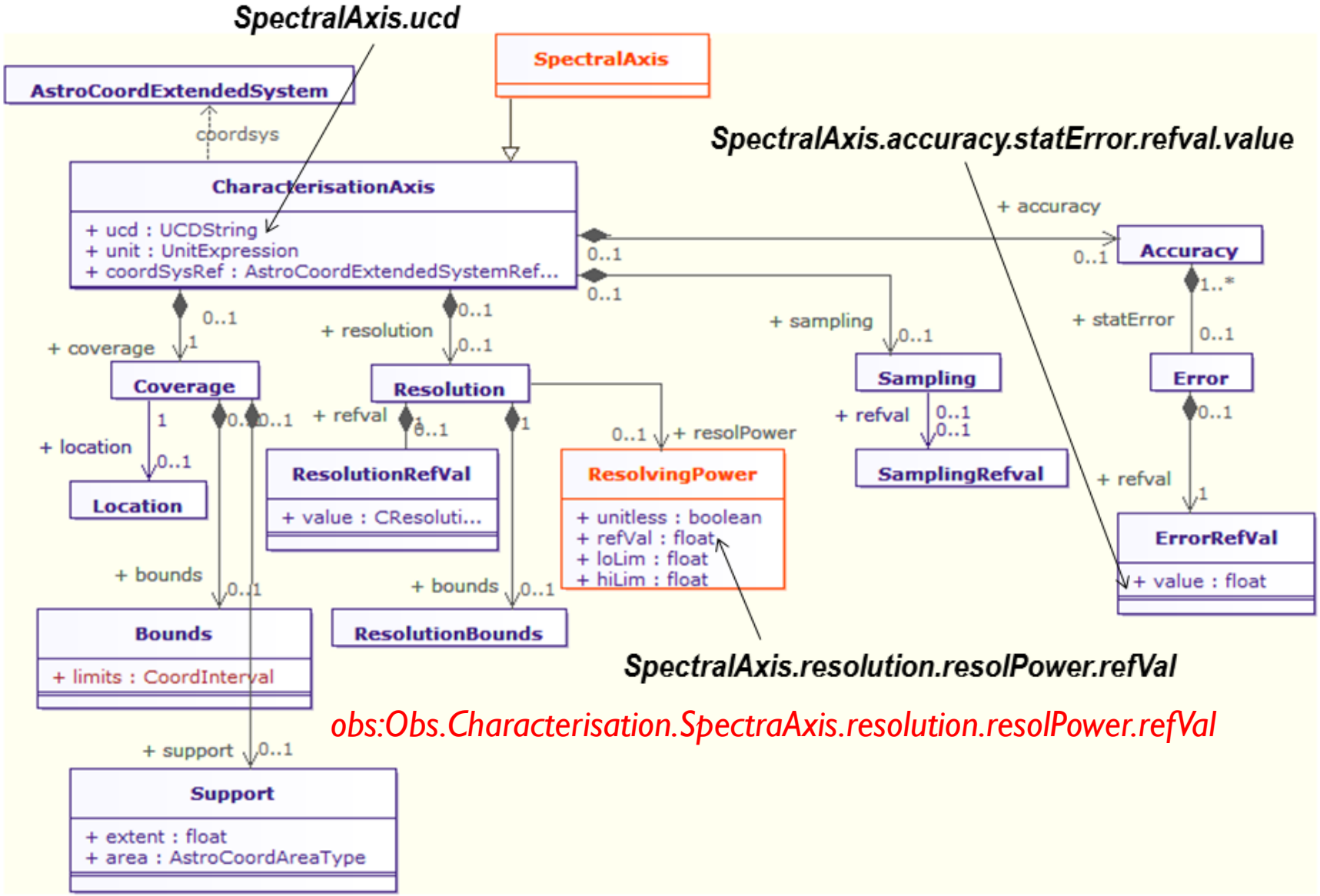
Data Model Working Group , M.Louys,
IVOA Interop Urbana, mai 2012

Status of the IVOA working draft

- New version at:
- http://www.ivoa.net/cgi-bin/twiki/bin/view/IVOA/Utypes
- Tried to re-organize the view :
  - Context detailed, especially for data model serializations
  - Use-cases and requirements gathered at various meetings
  - Stress the role as logical path to a data model item bringing in semantics and data structure.
  - Enhance data models documentation and implementation
    - Publish Utypes list (machine readable)
    - Access documentation via a utype–formed URI
    - Centralize all about one data model in a published record
  - Data model extension

→ TBC : need for description of implementation examples

From Obscore DM

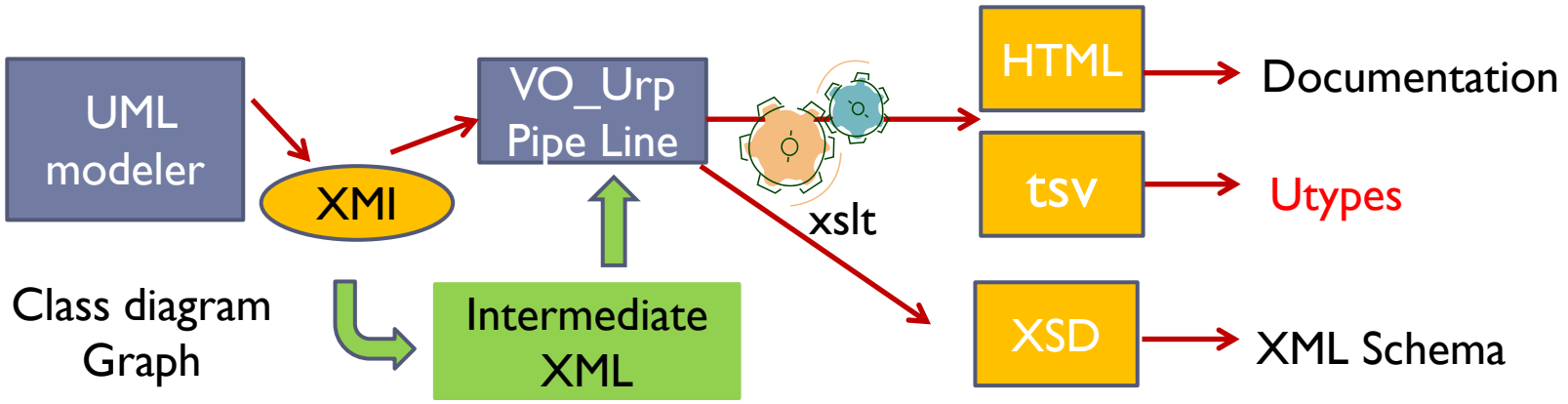


UML rules

- Design
  - No cycle in the UML diagram
  - Define root elements
  - Unique names for: attributes within a class, class within a package, package within a DM

  → warrants a unique path from root to finest grains elements in the DM
- Syntax
  - Attribute name start with lower case
  - Reference (association) names start with lower case
  - Classes and attribute names may use CamelCase

Utype automatic generation



- Use some reasonable design rules for UML representation

  UML Graph→ XML Tree → unique Utype path
- XMI not standard between modelers.
- Need for adapting before appling the VO_URP extraction pipeline (G.Lemson et coll.)

Data model extension (1)

Re-use class definitions from an existing IVOA DM

- Include the reusable parts of existing DMs in the model
- For instance ObsTAP re-uses classes from Spectrum DM: Curation, DataID, Target, and from SSA : Access class
- These objects are directly included in the model and reused in a specific way defined in the context of the UML model
- Utypes
  - Follow down the encapsulated attributes within reused classes
  - Use only one data model name space

This provides an explicit and static binding of the two data models with explicit versions.

Data model extension (2)

Define new data model fields for a specific use-case

- If a data model does not cover sufficiently the needs of a specific service or data collection
- Define a new data model name (name space)
  - Define new classes by derivation of existing classes
  - Addition of new classes
  - Provide documentation and utypes for the extented data model fields
- Example
  - SED data model defines a data model item of utype ***sed:Data.FluxAxis.Value***, a measured value in an observation
  - The NED service defines a computed Flux value when flux conversion are applied following its own calibration method.

Re-use the concept , but specify its particular application

Define a new data model item and new utype ***sedNed:Data.FluxAxis.Published Value***

Utype publishing

- Allow applications to access machine-readable datamodel summary

For each dm element: utype, datatype, units, ucd, status

- Allow users to access on-line documentation for each data model item via a utype based URI on the documentation part

- Provide a unified access to a standard data model and its side-products :
  - IVOA standard document
  - Utype list
  - XML schema
  - Instances of serialisations

Example of a DM registration/StandardRegExt

Conclusion and plans

- Utype are meant for various purposes by various VO actors
- Protocol, data base tags → labels
- Application : pointers to documentation and instanciation of a data model class.
- This is bound to some data organisation and to semantics via the UML /modeling point of view : not universal

- Focused on flat serialisation instances of approved data models

- Meta model programming may need a different partition, naming representation of metadata applied to concepts for the whole astronomical domain.