

Common Interface for Every Kind of Astronomical Data Service

Under Preparation !!

Yuji Shirasaki

National Astronomical Observatory of Japan

JVO

Contents

- Data Service Classification
 - Data model for each service class
- Definition of minimum subset of ADQL
- Update of ADQL syntax
 - "ucd" and "utype" attributes for columnReferenceType
 - Allow for using regionType in comparisonPredType
 - comparisonType: "within", "overlaps"
- Guideline of usage of ADQL
- Access to the Object data type
- Requirement in constructing the output VOTable

Table Data Model for Astronomical Data Service

- Query is described in SQL construct (ADQL).
- Query result is returned as a form of table (VOTable).
- Table Data Models for each data service class are required.

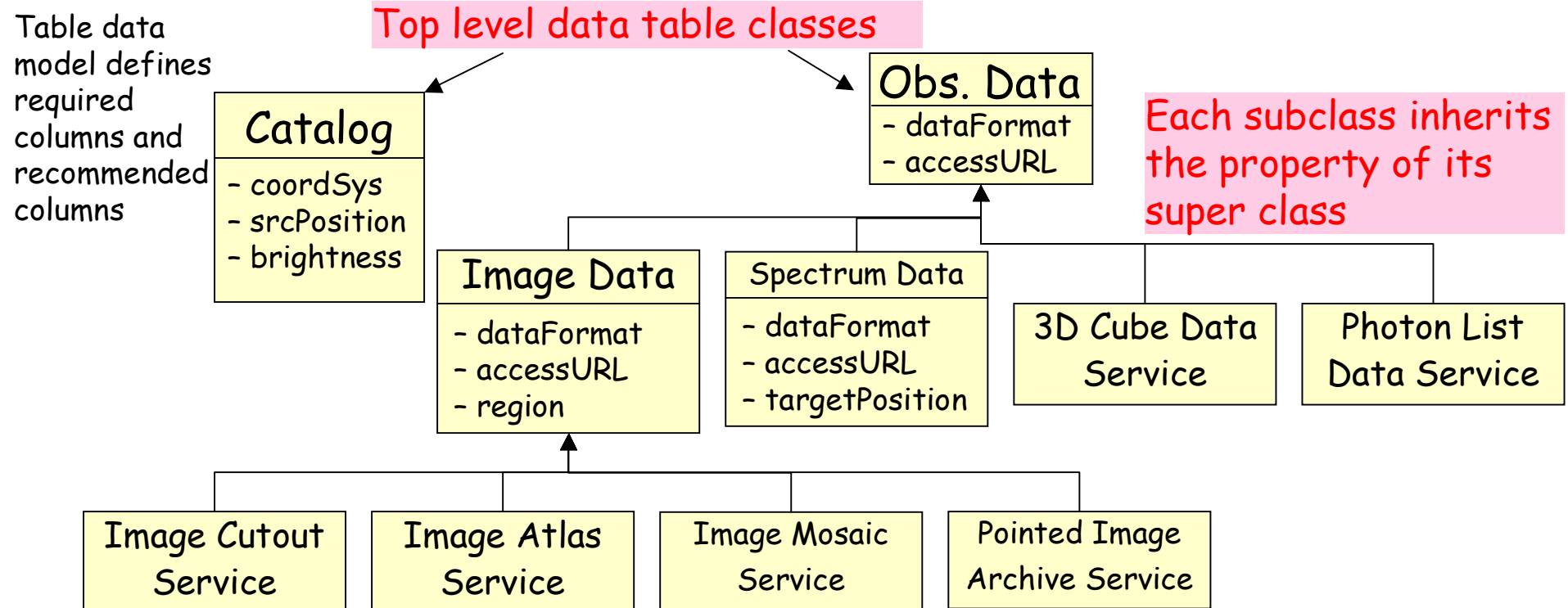


Table Data Model

- All the column must have the following attributes:
 - name, data type, unit, arraysize, width, precision, ucd, utype
 - column name can be specified by a data privder.
 - data type must follows the VOTable specification.
 - expression of unit is ...
 - if "arraysize" is not defined "*" is assumed for "char" data type and "1" for the other data types.
 - utype is supplied by VO data model naming scheme.
- Each table must have columns defined as 'Required' by the table data model.

Data Service Type	Required Columns (utype) (TBR)	Query Spec. of "Where" part
Catalog	Coverage.Location.Sky.Value	<position> within 'STC Region'
Image	Coverage.Region.Sky Data.Format Data.AccessURL	<region> overlaps 'STC Region' and <format> = 'Image/FITS'
Spectrum	Coverage.Location.Sky.Value Data.Format Data.AccessURL	<position> within 'STC Region' and <format> = 'VOTable'
3D Cube (Space x Spectrum)	Coverage.Region.Sky Data.Projection Data.Format Data.AccessURL	<region> overlaps 'STC Region' and <projection> = 'Image' and format = 'Image/JPEG'
Photon list	Coverage.Region.Sky Data.Projection Data.Format Data.AccessURL	<region> overlaps 'STC Region' and <projection> = 'Spectrum' and format = 'Spectrum/FITS'

Minimum Requirement for Query Spec.

Current ADQL can:

- 😊 describe most of the SQL92 syntax and some astro-extension
- 😞 many elements (33) & many data types (69), hard to build a data service. It is helpful to define minimum subset ADQL spec as a base line

- **SelectionItemType** (13 sub types in ADQL 0.8)
 - Only **AliasSelectionItem** and **ColumnReferenceType**
 - All the column must have **alias name**
- **FromTableType** (3 sub types in ADQL 0.8)
 - Only **TableType**
- **SearchType** (16 sub types in ADQL 0.8)
 - Only **IntersectionSearchType** and **ComparisonPredType**
 - Region Search Criterion can be specified by **ComparisonPredType**.

Minimum subset of ADQL for basic Skynode Service

Element: 33 (full) → 12 (basic)

Simple Type: 13 (f) → 4 (b)

Complex Type: 56 (f) → 12 (b)

Fundamental Type

xs:unsignedInt
 xs:string(*)
 xs:double(*)
 xs:long(*)

Simple Type

aggregateFunctionNameType
 allOrDistinctType
 binaryOperatorType
 comparisonType(*)
 jointTableQualifierType
 mathFunctionNameType
 orderDirectionType
 trigonometricFunctionNameType
 unaryOperatorType

Element

Allow	Restrict
Arg(*)	select(*)
Column	selectionList(*)
Condition(*)	Set
EndComment	Sigma
Expression(*)	StartComment
From(*)	Table(*)
GroupBy	TableName
Having	Tables
InTo	Unit(*)
Item(*)	where(*)
Literal(*)	fromTableType
Name	selection
Nature	
Order	
OrderBy	
Params	
Pattern	
Qualifier	
Region(*)	

Complex Type

ArrayOfFromTableType	includeTableType	searchType
ConstantListSet	inclusionSetType	selectType
aggregateFunctionType	inclusiveSearchType	selectionItemType
aliasSelectionItemType(*)	integerType(*)	selectionLimitType
allSelectionItemType	intersectionSearchType(*)	selectionListType
archiveTableType	intoType	selectionOptionType
atomType(*)	inverseSearchType	stringType(*)
betweenPredType	joinTableType	subQuerySet
binaryExprType	likePredType	tableType(*)
closedExprType	literalType(*)	trigonometricFunctionType
closedSearchType	mathFunctionType	unaryExprType
columnReferenceType(*)	notBetweenPredType	unionSearchType
comparisonPredType(*)	notLikePredType	userDefinedFunctionType
dropTableType	numberType	whereType(*)
exclusiveSearchType	orderExpressionType	xMatchTableAliasType
fromTableType	orderOptionType	xMatchType
fromType(*)	orderType	
functionType	realType(*)	
groupByType	regionSearchType	
havingType	scalarExpressionType	

Substitution of Column Name by UCD and UType

- Without knowing the column names, we cannot construct an ADQL...
- Possible Solution → use UCD or Utype for describing the query condition and selection list.
- Introduce "ucd" and "utype" attributes to the `columnReferenceType`

```
<Item xsi:type="columnReferenceType" Name="ra" Table="qso"/>
```

```
<Item xsi:type="columnReferenceType" ucd="pos.eq;src" Table="qso"/>
```

```
<Item xsi:type="columnReferenceType" utype="Target.pos" Table="qso"/>
```


ComparisonPredType vs RegionSearchType

Output

- FIELD id →
- FIELD name →
- The other FIELD attributes
- Some of the columns must always be included in the output VOTable.
These columns are defined in SIAP, SSAP, SXAP.
-