The Characterisation DM applied to the Infrared Space Observatory Archive

Alberto Micol

Pedro Osuna, Jesús Salgado, Aurélien Stébé, I. Ortiz

ESAVO

European Space Astronomy Centre (ESAC)

Rationale

- Infrared Space Observatory (ISO)
  - Operational from 1995 to 1998
  - Post operational phase ended in 2006
    - ISO Archive qualifies as a mature and stable archive, a perfect case
- Characterisation Data Model (CharDM)
  - Well developed concepts
  - List of CharDM UTYPEs
  - About to reach IVOA Reccomandation status
  - Tutorial on how to publish image products using CharDM
  - Version 1 in particular covering Data Discovery aspects

Aim: VO-standardised data discovery for ISO archived products using CharDM

ISO Metadata & Products

- Physical Data Model of the ISO Post Mission Archive (ISOPMA)
  - Detailed description of each database table and attribute

- ISO Data Products
  - ISOCAM (imaging), ISOLWS and ISOSWS (spectra), ISOPHOT (photometry)
  - Current effort limited to some modes (aka, AOTs): C01, L01, S01, P40
    - C01 is a multi-wavelength mode in its nature

      each of the binary table cells contains a 1-filter image

      For simple VO protocols, it was decided long ago to offer only 1 of such images -as if it were a normal 2d image taken in 1 filter
    - L01 grating range scan with 5 SW and 5 LW detectors 43-197 um
    - S01 has got 12 detectors which were scanning the spectral range (2-45um) using three “parallel” apertures via specific telescope jitter manouvers
    - P40 will be our next challenge, requires proper DM for photometry

VO perspective

- The ISO Physical Data Model perfectly answers all the ISO requirements

- Going VO, though, is a different story…
  - VO, in its quest for interoperability, establishes new standards for coherent data and metadata access…
  - In so doing, the VO is de facto enforcing a list of (optional) metadata that the original missions/observatories might not have claimed relevant by their own community.

- What is needed to go from an internal model to an IVOA model?
  1) Data Providers might need to find what the missing bits are;
     - *more metadata might be needed*
  2) Mapping from internal model to the IVOA model of interest

Mapping ISO and CharDM

It involves a carthusian work of
going through element by element of the CharDM,
finding the proper correspondence to one or more internal attributes,
as in:

Value(CharDM.utype[n]) = f(ISO.attribute[1], ISO.attribute[2], …)

for each of the instrument modes.

Examples:

time.coverage.bounds.extent => observations.utc_end - observations.utc_start

spatial.coverage.support.area => f(pointing.ra, pointing.dec, pointing.roll, observations.aperture)

Once the data provider delivers the mapping recipes …

Mapping ISO and CharDM

- ESAVO DMMapper
  - maps (both ways) data models to database table columns

```
<?xml version="1.0" encoding="UTF-8"?>
<TablesConfig xmlns="http://www.ivoa.net/wsdl/DMMapper/v0.1">
    <MetaTable>
        <Name>observations</Name>
        <Description></Description>
        <PrimaryKey>obsno</PrimaryKey>
        <Rows></Rows>
        <Rank></Rank>
        <Relations>
            <RelationShip>
                <ForeignKey>obsno</ForeignKey>
                <Table>iso_chardm</Table>
            </RelationShip>
            <RelationShip>
                <ForeignKey>obsno</ForeignKey>
                <Table>obs_pointing</Table>
            </RelationShip>
            <RelationShip>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ColumnsConfig xmlns="http://www.ivoa.net/wsdl/DMMapper/v0.1">
    <Table name="iso_chardm">
        <MetaColumn>
            <Name>err_pos_ref</Name>
            <Unit>deg</Unit>
            <Description></Description>
            <UCD>pos.errorEllipse</UCD>
            <DataType>float</DataType>
            <Precision></Precision>
            <ByteSize></ByteSize>
            <Rank></Rank>
        </MetaColumn>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<XPathQLConfig>
<property name="spatial.sampling.location.refval" value="iso_chardm.pos_sam_ref"/>
<property name="spectral.coverage.bounds.min"     value="iso_chardm.wav_cov_min"/>
<property name="spectral.coverage.bounds.max"     value="iso_chardm.wav_cov_max"/>
<property name="time.coverage.bounds.min"         value="observations;utc_start/86400 + 47527.0"/>
<property name="time.coverage.bounds.max"         value="observations;utc_end/86400 + 47527.0"/>
<property name="time.coverage.bounds.extent"      value="observations;(utc_end-utc_start)/86400."/>
```

DB ⟷ UFI mapping

The DMMapper uses a configuration file where the mapping between the internal ISO database structure and the external IVOA data model occurs:

```
<XPathQLConfig>
<property name=”spatial.sampling.location.refval" value="iso_chardm.pos_sam_ref"/>
<property name=”spatial.sampling.bounds.min”     value="iso_chardm.pos_sam_min"/>
<property name=”spatial.sampling.bounds.max”     value ="iso_chardm.pos_sam_max"/>
<property name=”spectral.coverage.bounds.min"   value="iso_chardm.wav_cov_min"/>
<property name=”spectral.coverage.bounds.max" value="iso_chardm.wav_cov_max"/>
<property name=”time.coverage.bounds.min”        value="observations;utc_start/86400 + 47527.0"/>
<property name=”time.coverage.bounds.max"        value="observations;utc_end/86400 + 47527.0"/>
<property name=”time.coverage.bounds.extent"     value="observations;(utc_endutc_start)/86400."/>
```

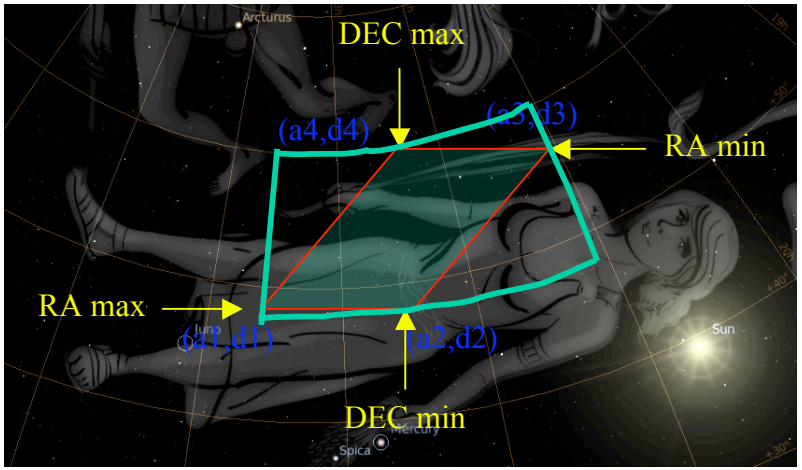
Note: those are not the official CharDM UFIs, because UFIs are being discussed.
As soon as the official UFIs will become available, we will make use of them.

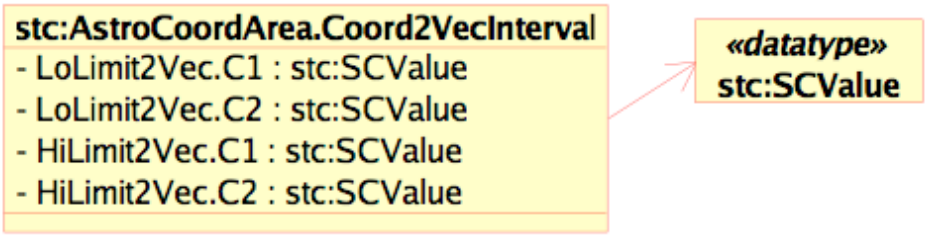
For example, instead of spectral.coverage.bounds.min I should have written something like…

Spatial bounds explained

CharDM likes to define a bounding box:
spatial.coverage.bounds.limits
(ra min, ra max, dec min, dec max)

where spatial.coverage.bounds.limits
is of type:
stc:Coord2VecInterval type





Food for thoughts: UFI discussion

RA Min:
spatialAxis.coverage.bounds.limits[stc:AstroCoordSystem.ID="UTC_ICRS_TOPO"]\
.stc:Coord2VecInterval.LoLimit2Vec.C1 ?

spatialAxis.coverage.bounds.limits[stc:AstroCoordSystem.ID="UTC_ICRS_TOPO"]\
.CharBox.stc:Value2.C1 ?

ISO and CharDM mapped!

- Once the mapping is engineered…

```
<?xml version="1.0" encoding="UTF-8"?>
<TablesConfig xmlns="http://www.ivoa.net/wsdl/DMMapper/v0.1">
    <MetaTable>
        <Name>observations</Name>
        <Description></Description>
        <PrimaryKey>obsno</PrimaryKey>
        <Rows></Rows>
        <Rank></Rank>
        <Relations>
            <RelationShip>
                <ForeignKey>obsno</ForeignKey>
                <Table>iso_chardm</Table>
            </RelationShip>
            <RelationShip>
                <ForeignKey>obsno</ForeignKey>
                <Table>obs_pointing</Table>
            </RelationShip>
            <RelationShip>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<ColumnsConfig xmlns="http://www.ivoa.net/wsdl/DMMapper/v0.1">
    <Table name="iso_chardm">
        <MetaColumn>
            <Name>err_pos_ref</Name>
            <Unit>deg</Unit>
            <Description></Description>
            <UCD>pos.errorEllipse</UCD>
            <DataType>float</DataType>
            <Precision></Precision>
            <ByteSize></ByteSize>
            <Rank></Rank>
        </MetaColumn>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<XPathQLConfig>
<property name="spatial.sampling.location.refval" value="iso_chardm.pos_sam_ref"/>
<property name="spectral.coverage.bounds.min"     value="iso_chardm.wav_cov_min"/>
<property name="spectral.coverage.bounds.max"     value="iso_chardm.wav_cov_max"/>
<property name="time.coverage.bounds.min"         value="observations;utc_start/86400 + 47527.0"/>
<property name="time.coverage.bounds.max"         value="observations;utc_end/86400 + 47527.0"/>
<property name="time.coverage.bounds.extent"      value="observations;(utc_end-utc_start)/86400."/>
```

Data Discovery: ADQL on CharDM

- ESAVO DMMapper

  It allows queries without* prior knowledge of db structure by

  - Resolving an ADQL query containing Unique Field IDs
    [mapping back UFIs to database columns]
  - Performing the database query
  - Returning VOTable with the UTYPEs (or UFIs?)
    [mapping database columns to UTYPEs]

  All one needs to know is the CharDM UTYPEs/UFIs

* almost… keep listening…

Simple query (astronomer viewpoint)

- “Please return (spatial, spectral, and time) CharDM information
  for all ISO observations
  which have an area greater than 0.5 sq. deg.”

Data Discovery: ADQL on CharDM

- Example:

```
<?xml version="1.0" encoding="utf-16"?><Select xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.ivoa.net/xml/ADQL/v0.7.4">  <Restrict Top="40" />
<SelectionList>
   <Item xsi:type="columnReferenceType" xpathName=”char.observation_id" />
   <Item xsi:type="columnReferenceType" xpathName="time.coverage.location.refval" />
   <Item xsi:type="columnReferenceType" xpathName="spectral.coverage.location.refval" />
   <Item xsi:type="columnReferenceType" xpathName="spatial.sampling.location.refval" />
   <Item xsi:type="columnReferenceType” xpathName=“spatial.coverage.bounds.ra.min” />
   <Item xsi:type="columnReferenceType" xpathName="spatial.coverage.bounds.ra.max" />
   <Item xsi:type="columnReferenceType" xpathName="spatial.coverage.bounds.dec.min" />
   <Item xsi:type="columnReferenceType" xpathName="spatial.coverage.bounds.dec.max" />
   <Item xsi:type="columnReferenceType" xpathName="spatial.coverage.support.area" />
</SelectionList>
<Where>
<Condition xsi:type="comparisonPredType" Comparison="&gt;">
<Arg xsi:type="columnReferenceType" xpathName="spatial.coverage.support.area.extent" />
<Arg xsi:type="atomType">   <Literal xsi:type="integerType" Value="0.5" />   </Arg>
</Condition>  </Where></Select>
```

Data Discovery: ADQL on CharDM

Note 1: CharDM does not supply a UTYPE for the observation ID

```
<Item xsi:type="columnReferenceType" xpathName="char:observation_id" />
<Item xsi:type="columnReferenceType" xpathName="obs:observation_id" />
```

We cannot wait until the obsdm is ready…

Note 2: default vs explicit units?

```
<Condition xsi:type="comparisonPredType" Comparison="&gt;">
<Arg xsi:type="columnReferenceType" xpathName="spatial.coverage.support.area.extent" />
  <Arg xsi:type="atomType">   <Literal xsi:type="integerType" Value="0.5" />   </Arg>
</Condition>
```

```
Spatial.coverage.support.area.extent[unit="deg**2"] < 0.5 ?
```

DMMapper output

```
<VOTABLE version="1.1" xmlns="http://www.ivoa.net/xml/VOTable/v1.1">
<RESOURCE type="results">
<INFO name="QUERY_STATUS" value="OK"/>
<TABLE>
<FIELD … ucd="meta.id;meta.dataset” utype=“char.observation_id” />
<FIELD … ucd=“time.epoch;obs” unit=“MJD” utype=”time.coverage.location.refval” />
<FIELD … ucd="em.wl.central"    unit=“um”   utype="spectral.coverage.location.refval” />
<FIELD … ucd="pos.wcs.scale"   unit="deg”  utype="spatial.sampling.location.refval” />
…
<FIELD … ucd="?" unit="deg" utype="spatial.coverage.support.area” />
<DATA>…
```

Should the output VOTable contain UTYPEs or UFIs?
VOTable standard knows about UTYPEs not UFIs, a problem?

Demo (part 0)

A humble demo could be:

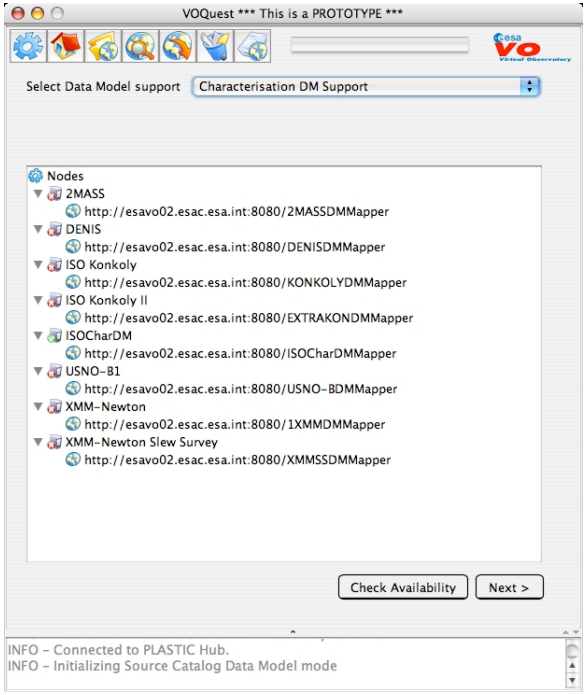
```
wget "http://esavo02.esac.esa.int:8080/ISOCharDMMapper/performQuery' \
    --post-file=your.adql -O votable.xml
```

and that would be it.

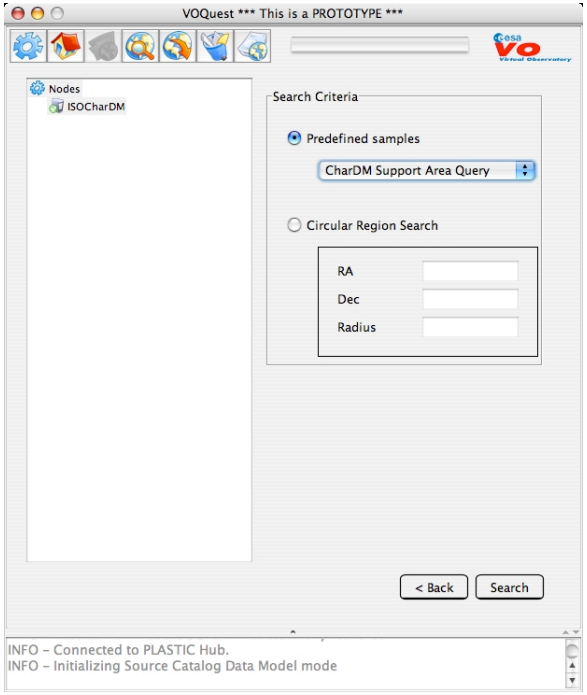
We prefer something fancier…

Demo (part 1)

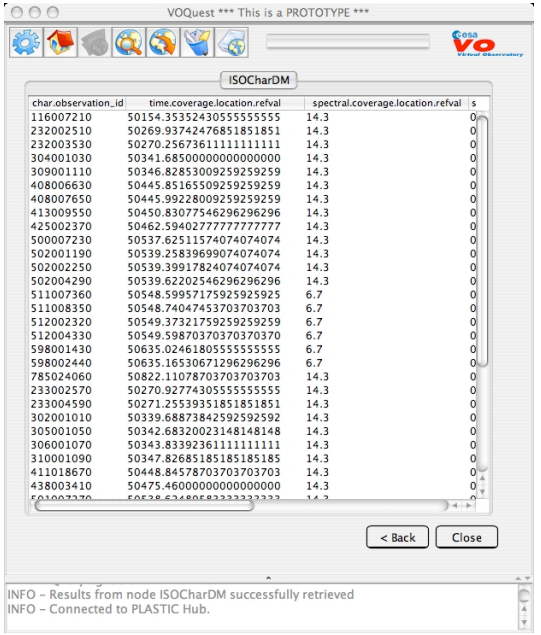
VOQuest prototype



a) Which of the various services does **support CharDM**? (query by utype)



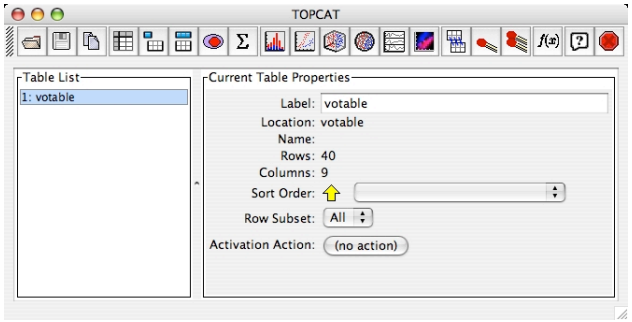
b) The **ADQL** previously shown is hardcoded in the **VOQuest** prototype

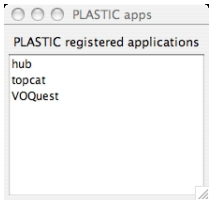


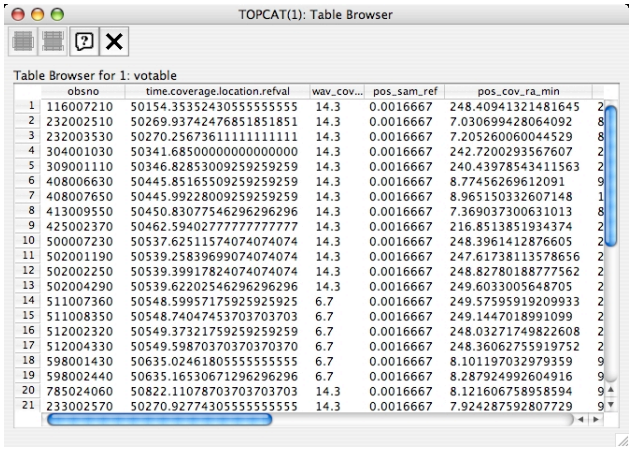
c) VOQuest sends the query to the **DMMapper**, which does the mapping and returns a **VOTable**

Demo (part 2)

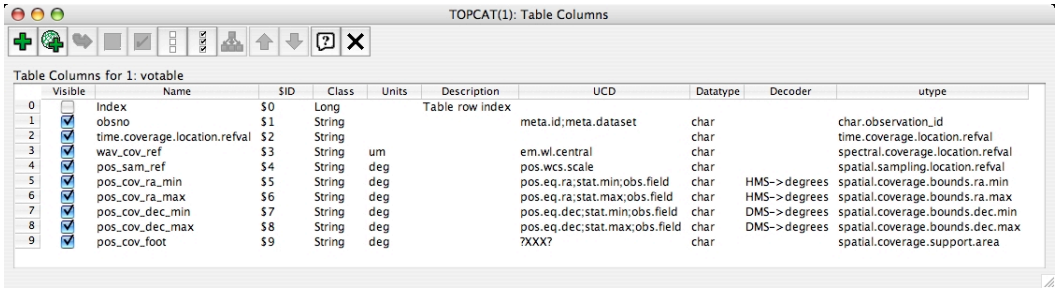
Passing results to TOPCAT







	obsno	time.coverage.location.refval	wav_cov...	pos_sam_ref	pos_cov_ra_min	
1	116007210	50154.35352430555555555	14.3	0.0016667	248.40941321481645	2
2	232002510	50269.93742476851851851	14.3	0.0016667	7.030699428064092	8
3	232003530	50270.25673611111111111	14.3	0.0016667	7.205260060044529	8
4	304001030	50341.68500000000000000	14.3	0.0016667	242.7200293567607	2
5	309001110	50346.82853009259259259	14.3	0.0016667	240.43978543411563	2
6	408006630	50445.85165509259259259	14.3	0.0016667	8.77456269612091	9
7	408007650	50445.99228009259259259	14.3	0.0016667	8.965150332607148	1
8	413009550	50450.83077546296296296	14.3	0.0016667	7.369037300631013	8
9	425002370	50462.59402777777777777	14.3	0.0016667	216.8513851934374	2
10	500007230	50537.62511574074074074	14.3	0.0016667	248.3961412876605	2
11	502001190	50539.25839699074074074	14.3	0.0016667	247.61738113578656	2
12	502002250	50539.39917824074074074	14.3	0.0016667	248.82780188777562	2
13	502004290	50539.62202546296296296	14.3	0.0016667	249.6033005648705	2
14	511007360	50548.59957175925925925	6.7	0.0016667	249.57595919209933	2
15	511008350	50548.74047453703703703	6.7	0.0016667	249.1447018991099	2
16	512002320	50549.37321759259259259	6.7	0.0016667	248.03271749822608	2
17	512004330	50549.59870370370370370	6.7	0.0016667	248.36062755919752	2
18	598001430	50635.02461805555555555	6.7	0.0016667	8.101197032979359	9
19	598002440	50635.16530671296296296	6.7	0.0016667	8.287924992604916	9
20	785024060	50822.11078703703703703	14.3	0.0016667	8.121606758958594	9
21	233002570	50270.92774305555555555	14.3	0.0016667	7.924287592807729	9

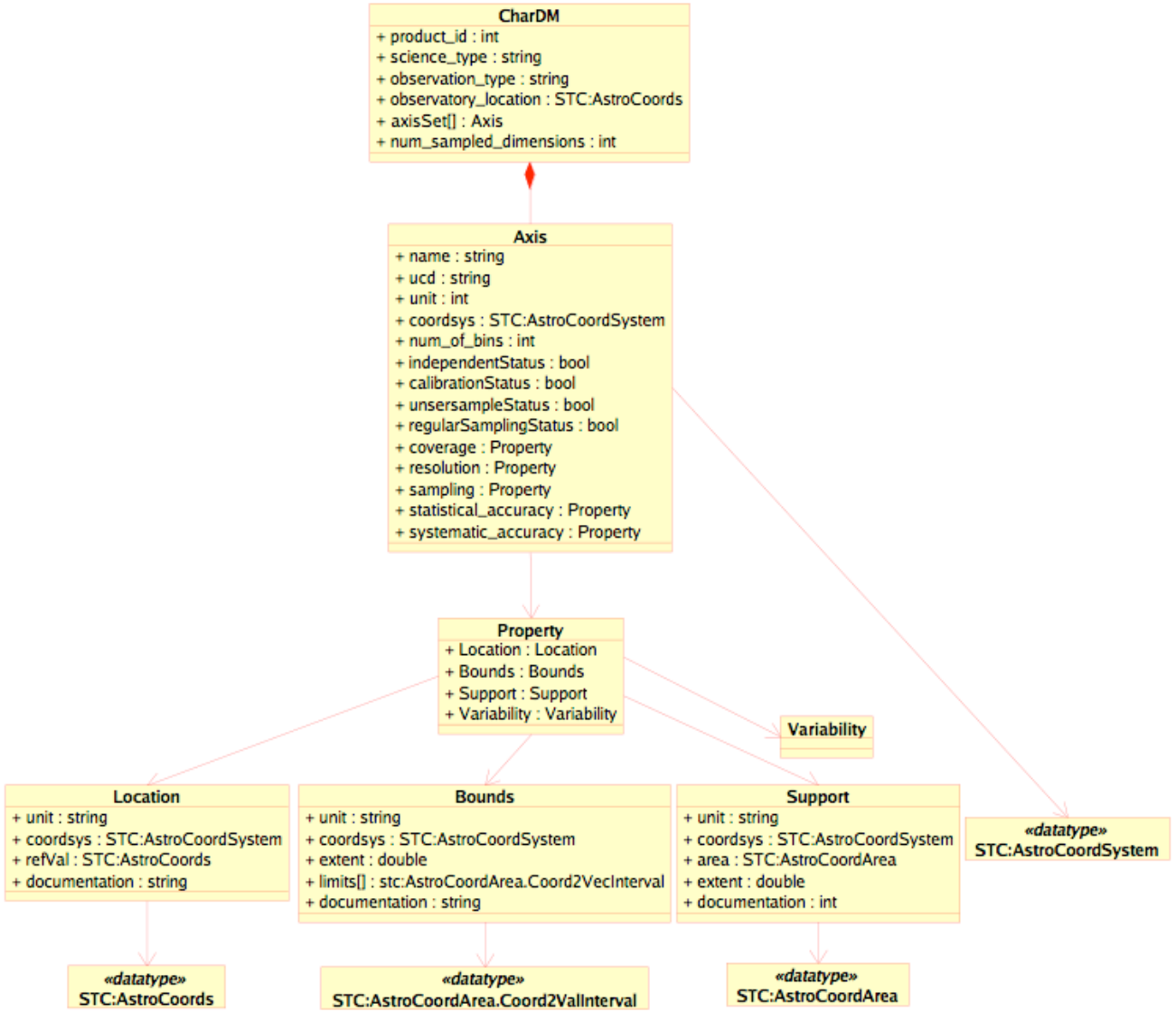


	Visible	Name	$ID	Class	Units	Description	UCD	Datatype	Decoder	utype
0	☐	Index	$0	Long		Table row index				
1	☑	obsno	$1	String			meta.id;meta.dataset	char		char.observation_id
2	☑	time.coverage.location.refval	$2	String				char		time.coverage.location.refval
3	☑	wav_cov_ref	$3	String	um		em.wl.central	char		spectral.coverage.location.refval
4	☑	pos_sam_ref	$4	String	deg		pos.wcs.scale	char		spatial.sampling.location.refval
5	☑	pos_cov_ra_min	$5	String	deg		pos.eq.ra;stat.min;obs.field	char	HMS->degrees	spatial.coverage.bounds.ra.min
6	☑	pos_cov_ra_max	$6	String	deg		pos.eq.ra;stat.max;obs.field	char	HMS->degrees	spatial.coverage.bounds.ra.max
7	☑	pos_cov_dec_min	$7	String	deg		pos.eq.dec;stat.min;obs.field	char	DMS->degrees	spatial.coverage.bounds.dec.min
8	☑	pos_cov_dec_max	$8	String	deg		pos.eq.dec;stat.max;obs.field	char	DMS->degrees	spatial.coverage.bounds.dec.max
9	☑	pos_cov_foot	$9	String	deg		?XXX?	char		spatial.coverage.support.area

Via PLASTIC, the ISO CharDM information is passed to TOPCAT, or any other VO Application, for visualisation, analysis , and interpretation.

Simplified Characterisation DM UML

Internal UML used to engineer our system



Conclusions

Recap

What we did was to:

- map the ISO internal model to the IVOA CharDM
- assign (provisional) UFIs to each attribute
- formulate ADQL queries using only UFIs (no prior DB knowledge)

To demonstrate the concept:

- VOQuest queries are passed to the DMMapper
- DMMapper maps ADQL to the actual ISO internal model
- Returned information is mapped back to (provisional) CharDM UFIs
- VOTable with UFIs is passed via PLASTIC to an application (TOPCAT) for analysis.

Conclusions (2/2)

Main conclusion:

- ✓ No need to know the internal DB structure,
  applications can make use of UFIs,
  Astronomers and developers can concentrate on the science case
  not on specific implementations at different data centres.

Caveats:

- UFIs are now the main item on our table
- VOTable + STC in the making…
- VOTable currently knows about UTYPEs, not UFIs