

# The

ModelInstanceInVot

# Alternative or Evolution

## Mapping Syntax

# A Simple Position with a Reference to a Space Frame

GitHub Use-cases -> standard\_properties -> ivoa\_csc2\_example.xml

```
<INSTANCE dmtpe="mango:Parameter">
  <ATTRIBUTE dmrole="mango:Parameter.semantic">
    <INSTANCE dmtpe="mango:VocabularyTerm">
      <ATTRIBUTE dmrole="mango:VocabularyTerm.label">
        <LITERAL value="position" dmtpe="ivoa:string"/>
      </ATTRIBUTE>
    </INSTANCE>
  </ATTRIBUTE>
  <ATTRIBUTE dmrole="mango:Parameter.ucd">
    <LITERAL value="pos" dmtpe="ivoa:string"/>
  </ATTRIBUTE>
  <COMPOSITION dmrole="mango:Parameter.measure">
    <INSTANCE dmtpe="mango:measures.LonLatSkyPosition">
      <ATTRIBUTE dmrole="mango:measures.LonLatSkyPosition.coord">
        <INSTANCE dmtpe="mango:coordinates.LonLatPoint">
          <ATTRIBUTE dmrole="mango:coordinates.LonLatPoint.longitude">
            <COLUMN dmtpe="ivoa:RealQuantity" ref="col9"/>
          </ATTRIBUTE>
          <ATTRIBUTE dmrole="mango:coordinates.LonLatPoint.latitude">
            <COLUMN dmtpe="ivoa:RealQuantity" ref="col10"/>
          </ATTRIBUTE>
          <REFERENCE dmrole="coords:Coordinate.coordSys">
            <IDREF_coosys</IDREF>
          </REFERENCE>
        </INSTANCE>
      </ATTRIBUTE>
    </COMPOSITION>
  </INSTANCE>
</INSTANCE>
```

## VODML Mapping

```
<INSTANCE dmrole="mango:MangoObject.parameters" dmtpe="mango:Parameter">
  <ATTRIBUTE dmrole="mango:Parameter.semantic" dmtpe="ivoa:string" value="http://www.ivoa.net/rdf/ua">
  <ATTRIBUTE dmrole="mango:Parameter.ucd" dmtpe="ivoa:string" value="pos"/>
  <ATTRIBUTE dmrole="mango:Parameter.description" dmtpe="ivoa:string" value="Corrected position"/>
  <ATTRIBUTE dmrole="mango:Parameter.reductionStatus" dmtpe="ivoa:string" value="Corrected"/>
  <INSTANCE dmrole="mango:Parameter.measure" dmtpe="mango:stcextend.LonLatSkyPosition">
    <INSTANCE dmrole="mango:stcextend.LonLatSkyPosition.coord" dmtpe="mango:stcextend.LonLatPoint">
      <ATTRIBUTE dmrole="mango:stcextend.LonLatPoint.longitude" dmtpe="ivoa:real" ref="col9"/>
      <ATTRIBUTE dmrole="mango:stcextend.LonLatPoint.latitude" dmtpe="ivoa:real" ref="col10"/>
      <INSTANCE dmrole="coords:Coordinate.coordSys" dmref="SpaceFrame_Galactic"/>
    </INSTANCE>
  </INSTANCE>
</INSTANCE>
```

## ModelInstanceInVot

- Less XML elements
- More @attributes

# Similar Path for Mapping and Models

**Model** path leading to the space coordinate system



## 2 flaws

- Mapping path differs from the model one
- path depends on the location of the searched object

**VODML mapping:** path leading to the space coordinate system that is located in GLOBALS



... or located here



**ModelInstanceInVot:** path leading to the space coordinate system



- Both mapping and model paths are similar whatever the location of the space coord object is



# Browsing Path Faith to the Model

source	ra	dec	time
MovingStar	12.34	-45.56	Yesterday
MovingStar	14.78	-56.78	Today

The source name is repeated in each row although all are related to the same source

Param source=MovingStar		
ra	dec	time
12.34	-45.56	Yesterday
14.78	-56.78	Today

The source name is given once as a *Param*

ra	dec	time
12.34	-45.56	Yesterday
14.78	-56.78	Today

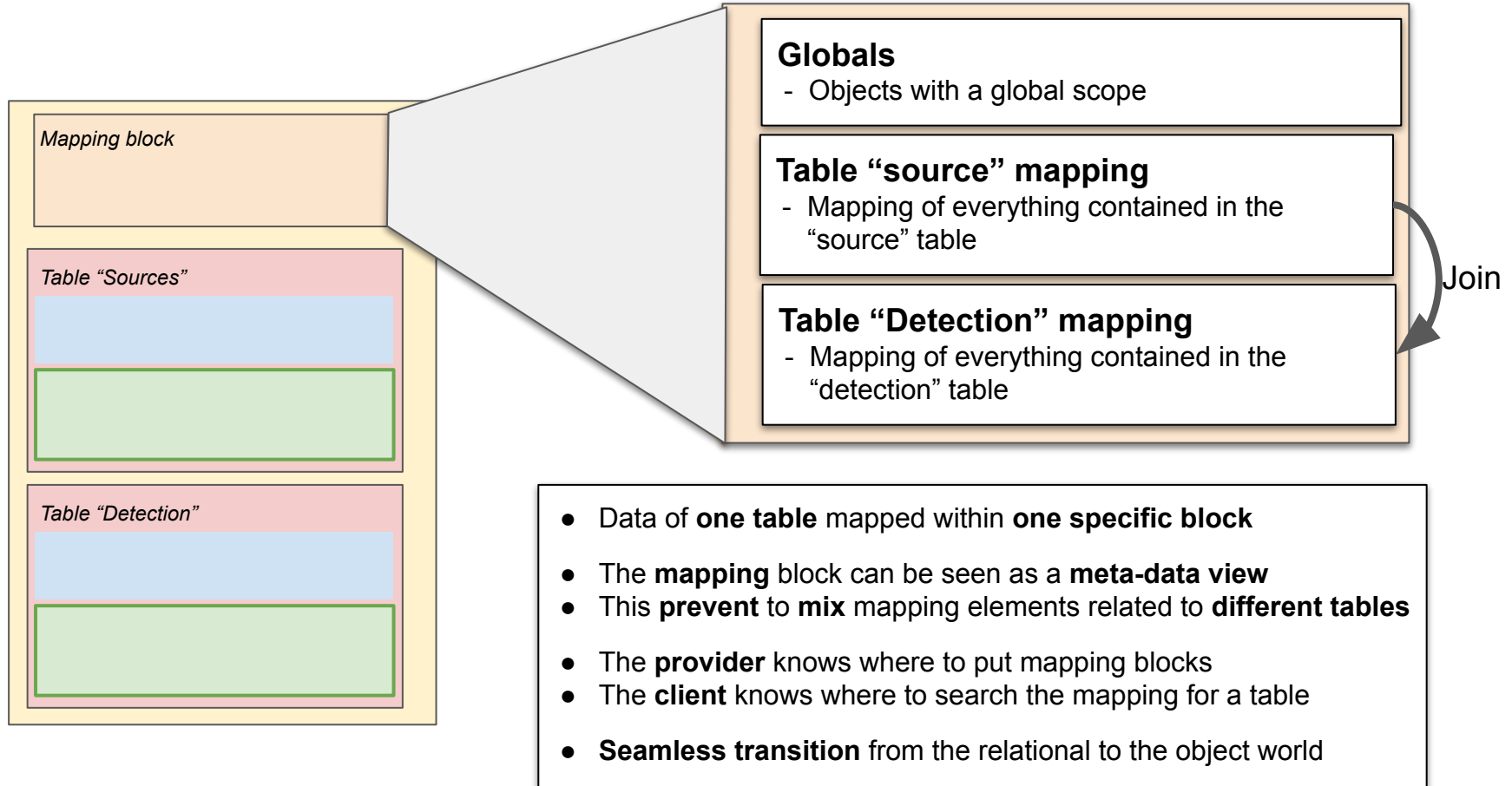
Everyone knows I'm working on MovingStar, no reason to write this in my dataset.

```
<INSTANCE dmttype="Star">
  <ATTRIBUTE dmrole="Star.name" dmttype="String" ref="source" value="MovingStar" />
  <ATTRIBUTE dmrole="Star.longitude" dmttype="real" ref="ra"/>
  <ATTRIBUTE dmrole="Star.latitude" dmttype="real" ref="dec"/>
  <ATTRIBUTE dmrole="Obs.timeStamp" dmttype="real" ref="time"/>
</INSTANCE>
```

Same mapping component, only the attributes do change

- The mapping component keeps the same structure (DOM) in all cases.
- The specificity of the data arrangement is taken into account by the attributes of the **<ATTRIBUTE>** elements
  - **@value** : literal value
  - **@ref** : reference to a **<FIELD>** first and then to an **<PARAM>**
  - Resolve **@ref** first and the **@value**
- XPATH to model leaves independent from the data layout
  - Follow the XPATH to reach the leaf
  - Then checks the leaf element attributes to get the value

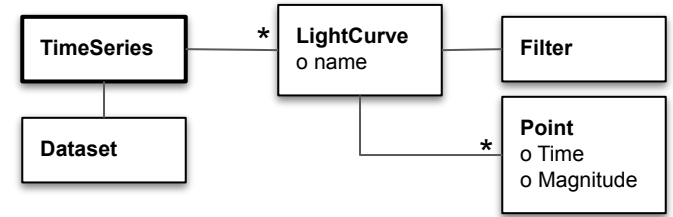
# Tight coupling with the VOTable Structure



```

<TABLE_MAPPING tabref='Results'>
  <!--
  The TABLE contains one time series
  -->
  <INSTANCE dmrole="root" dmtpe="mock_ts:TimeSeries">
    <!--
    TS meta data
    -->
    <INSTANCE dmrole="mock_ts:TimeSeries.dataSet" dmtpe="mock_ts:DataSet">
      <!--
      The TS in a collection of light curves
      -->
      <COLLECTION dmrole="mock_ts:TimeSeries.lightCurves">
        <!--
        Light curve for B band
        -->
        <INSTANCE dmrole="mock_ts:TimeSeries.lightCurves" dmtpe="mock_ts:LightCurve">
          <!--
          Light curve meta data: Name + filter
          -->
          <ATTRIBUTE dmrole="mock_ts:LightCurve.name" dmtpe="ivoa:string" value="Light curve G band" />
          <INSTANCE dmrole="mock_ts:LightCurve.filter" dmref="PhotFrame_gaiaG" />
          <!--
          Light curve points: populated by iterating on the data table with a filter on band G
          -->
          <COLLECTION dmrole="mock_ts:TimeSeries.points">
            <TABLE_ROW_TEMPLATE>
          </COLLECTION>
        </INSTANCE>
        <!--
        Light curve for RP band
        -->
        <INSTANCE dmrole="mock_ts:TimeSeries.lightCurves" dmtpe="mock_ts:LightCurve">
          <ATTRIBUTE dmrole="mock_ts:LightCurve.name" dmtpe="ivoa:string" value="Light curve RP band" />
          <INSTANCE dmrole="mock_ts:LightCurve.filter" dmref="PhotFrame_gaiaRP" />
          <!--
          Light curve points: populated by iterating on the data table with a filter on band R
          -->
          <COLLECTION dmrole="mock_ts:TimeSeries.points">
            <TABLE_ROW_TEMPLATE>
          </COLLECTION>
        </INSTANCE>
        <!--
        Light curve for BP band
        -->
        <INSTANCE dmrole="mock_ts:TimeSeries.lightCurves" dmtpe="mock_ts:LightCurve">
          <ATTRIBUTE dmrole="mock_ts:LightCurve.name" dmtpe="ivoa:string" value="Light curve BP band" />
          <INSTANCE dmrole="mock_ts:LightCurve.filter" dmref="PhotFrame_gaiaBP" />
          <!--
          Light curve points: populated by iterating on the data table with a filter on band B
          -->
          <COLLECTION dmrole="mock_ts:TimeSeries.points">
            <TABLE_ROW_TEMPLATE>
          </COLLECTION>
        </INSTANCE>
      </COLLECTION>
    </INSTANCE>
  </TABLE_MAPPING>
  
```

- Mapping of the data enclosed in the *Results* table
- The table contains one Time Series
- Time Series meta-data (folded)
- The TS contains several light curves
- Mapping of one light curve
- Light curve meta-data
- Photometric points mapping with a (folded) table iterator that contains a filter



- Maps both table PARAMS and table DATA
- 3 table row iterators populating the same instance

# Features I haven't mentioned

- Fields Pointers (TBD)
  - Pointers on **FIELD** attributes (ucd, unit or description)
- GROUP\_BY operator
  - Create a collection of collections by grouping rows by some key (e.g.source ID)
- Shortcuts
  - One mapping elements for nested objects that are parts of the standard
- Datatype
  - Mapping for literal enumerations



# ModelInstanceInVot Status

- Working Draft
  - <https://github.com/ivoa-std/ModelInstanceInVot>
- Data annotation on the fly tested on Vizier
  - <http://viz-beta.u-strasbg.fr/viz-bin/Mango?-source=IX/45/csc11&-out.max=10>
- Test code
  - <https://github.com/ivoa/modelinstanceinvot-code>
  - Component-based mapper prototyped
  - Lots of annotated data sample
- No definitive break with the VODML Mapping proposal but significant differences

backup

# History

- **2017**

- Consensus on the global architecture for mapping models in VOTables

- **Work on legacy data (Vizier + TAPservices)**

- Different perspective on the way the mapping is built and consumed
- Would be easier with an appropriate syntax
  - Annotation process
    - Automatic mapper developed in parallel with MANGO
    - On the fly annotation (Vizier)
  - Client design

- **2020**

- `ModelInstanceInVot` proposal

# Tight coupling with the VOTable Structure

```
<!--  
  This block maps the data of the table identified as "_lightcurve_G"  
<TABLE_MAPPING tableref="_lightcurve_G">  
  <!--  
    Table _lightcurve_G contains one instance of cube:SparseCube  
    This instance is made of a ds:experiment.ObsDataset instance,  
    populated with literals and PARAMS, - a collection of points populated  
    by the table rows The parser is "encouraged" to start by the  
    element with "root" as role  
    <!--  
    <INSTANCE dmrole="" dmtpe="cube:SparseCube">  
    <!--  
      DataSet instance : One DataSet per Cube instance  
      -->  
    <INSTANCE dmrole="cube:SparseCube.dataset" dmtpe="ds:experiment.ObsDa
```

This block contains anything related to the `_lightcurve_G` table

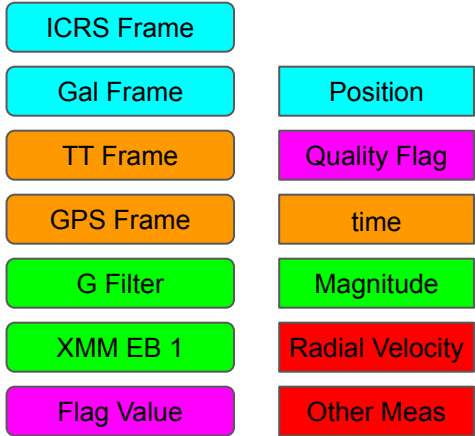
The table contains one `SparseCube` instance

```
<!--  
  This block maps the data of the table identified as "Results"  
  -->  
<TABLE_MAPPING tableref="Results">  
  <!--  
    The table contain a set of Mango instances, this is why we have COLLECTION at the top level  
    <!--  
    <COLLECTION dmrole="root">  
    <!--  
      The TABLE_ROW_TEMPLATE element tells the parser that one mango instance  
      must be instanciated for each table row  
      -->  
    <TABLE_ROW_TEMPLATE>  
    <INSTANCE dmrole="" dmtpe="mango:MangoObject">  
    <!--  
      <INSTANCE dmrole="mango:MangoObject.identifier" dmtpe="ivoa:string" ref="namesaada" />
```

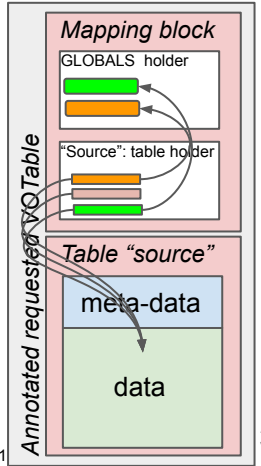
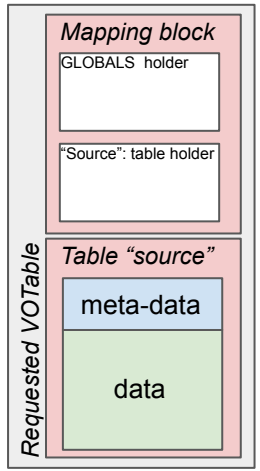
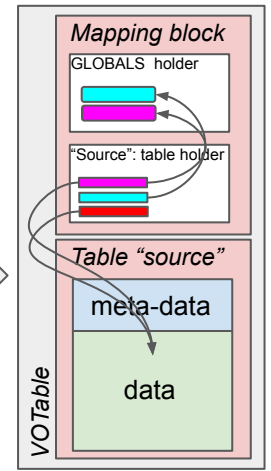
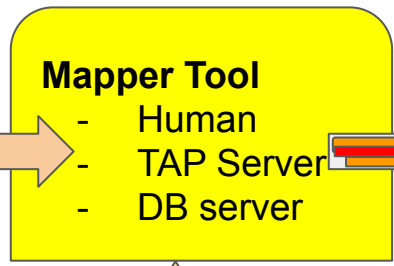
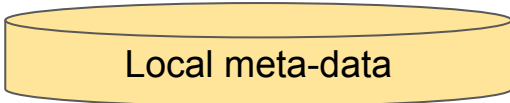
The table contains one collection of **MANGO** instances  
That collection must be populated by iterating on the data table

# #2 Modularity

- The mapping process must be possible by stacking components
- The mapper just needs to know the binding of the components leaves with the actual data
- This information is easy to infer or to store in an external resource

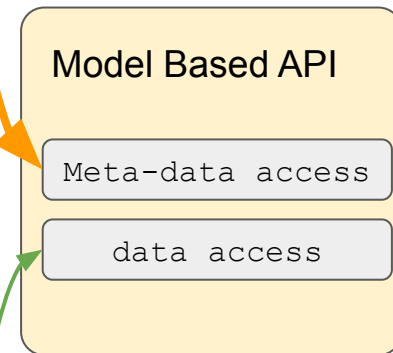
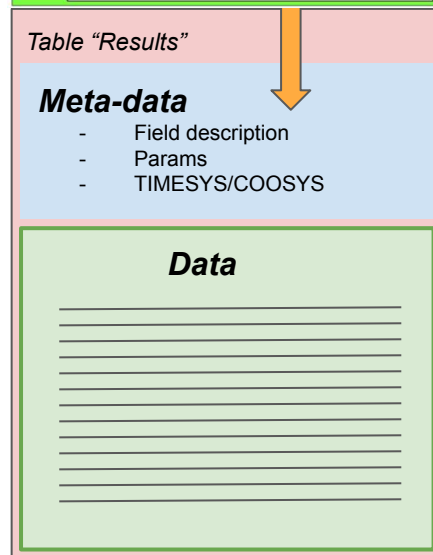
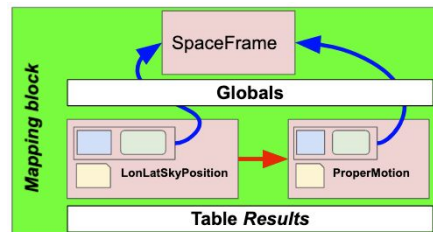
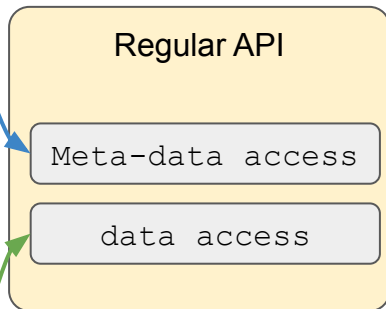
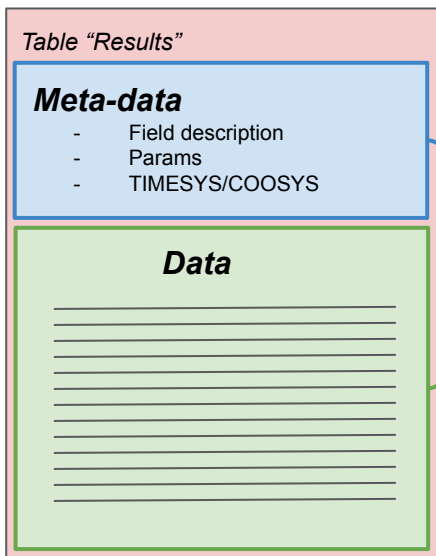


Mapping Components



## #2 Compliance with the Current APIs

- Data of **one table** mapped within **one specific block**
- The **mapping** block can be seen as a **meta-data** view
- Allow client code to easily jump for the mapped element to the actual (meta) data
- **Smooth transition** from the relational to the object world



# #1 Readability

- Any complex data hierarchy can be exchanged with [key-value pairs, tuples and lists]
  - See JSON based Web applications

Key/value pair

```
<ATTRIBUTE
  dmrole="mango:stcextend.PhotFilter.effectiveWavelength"
  dmtype="ivoa:real" value="7740.87" />
```

Tuple

```
<INSTANCE dmrole="mango:Parameter.measure"
  dmtype="mango:Parameter">
  <ATTRIBUTE dmrole="mango:Parameter.semantic"
    dmtype="ivoa:string" value="#position" />
  <ATTRIBUTE dmrole="mango:Parameter.ucd"
    dmtype="ivoa:string" value="pos.eq;meta.main" />
  <ATTRIBUTE dmrole="mango:Parameter.description"
    dmtype="ivoa:string" value="this is the position" />
  <INSTANCE dmrole="mango:Parameter.measure"[]
  <INSTANCE dmrole="meas:Measure.error"[]
</INSTANCE>
```

List

```
<COLLECTION size="-1"
  dmrole="mango:MangoObject.parameters">
  <INSTANCE dmrole="mango:Parameter.measure"[]
  <INSTANCE dmrole="mango:Parameter.measure"[]
  <INSTANCE dmrole="mango:Parameter.measure"[]
  <INSTANCE dmrole="mango:Parameter.measure"[]
  <INSTANCE dmrole="mango:Parameter.measure"[]
  <INSTANCE dmrole="mango:Parameter.measure"[]
</COLLECTION>
```

# #1 Readability

```
Class DogOwner
- String name
- Dog[] dogs
```

```
Class Dog
- String Name
- String Breed
```

```
{
  "name": "Laurent",
  "Dogs": [
    {
      "Name": "Dupond",
      "Breed": "Bulldog"
    },
    {
      "Name": "Dupont",
      "Breed": "Foxhound"
    }
  ]
}
```

```
<INSTANCE dmtpe='DogOwner'>
  <ATTRIBUTE dmrole='name' dmtpe='String' />
  <COLLECTION dmrole='dogs' >
    <INSTANCE dmrole='dogs' dmtpe='Dog'>
      <ATTRIBUTE dmrole='name' dmtpe='String' value="Dupond"/>
      <ATTRIBUTE dmrole='breed' dmtpe='String' value="Bulldog"/>
    </INSTANCE>
    <INSTANCE dmrole='dogs' dmtpe='Dog'>
      <ATTRIBUTE dmrole='name' dmtpe='String' value="Dupont"/>
      <ATTRIBUTE dmrole='breed' dmtpe='String' value="Foxhound"/>
    </INSTANCE>
  </COLLECTION>
</INSTANCE>
```



# Other Features

**Shortcuts:** Model components that are parts of a standard can be folded in compact XML elements

```
<INSTANCE dmrole="coords:Point.axis1" dmtpe="ivoa:RealQuantity">  
  <ATTRIBUTE dmrole="ivoa:RealQuantity.value" dmtpe="ivoa:real" ref="RA_ICRS"/>  
  <ATTRIBUTE dmrole="ivoa:Quantity.unit" dmtpe="ivoa:Unit" value="deg"/>  
</INSTANCE>
```



```
<SC_REALQUANTITY dmrole="coords:Point.axis1"  
  ref="RA_ICRS" unit="deg" />
```

**Row filtering:** Only processing data with a certain field value

```
<COLLECTION dmrole="dmrole">  
  <TABLE_ROW_TEMPLATE>  
    <FILTER ref="ref" value="value"/>  
    <INSTANCE dmref="dmref" dmrole="dmrole" />  
  </TABLE_ROW_TEMPLATE>  
</COLLECTION>
```

**Foreign keys:** Joining data from different tables

```
<TABLE_MAPPING tableref="fgdgdfdf">  
  <COLLECTION dmrole="dmrole">  
    <JOIN tableref="tableref" primary="primary" foreign="foreign">  
      <INSTANCE dmref="dmref" dmrole="dmrole" />  
    </JOIN>  
  </COLLECTION>  
</TABLE_MAPPING>
```

**Row grouping:** Grouping data of the same source spread over multiple rows

```
<GROUPBY ref="ref">  
  <INSTANCE dmref="dmref" dmrole="dmrole">  
    <ATTRIBUTE ref="ref" dmrole="dmrole" dmtpe="dmtpe"  
      value="dmvalue" />  
  </INSTANCE>  
</GROUPBY>
```

# One mapping block per <TABLE>

```
<?xml version="1.0" encoding="UTF-8"?>
<VOTABLE xmlns="http://www.ivoa.net/xml/VOTable/v1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ivoa.net/xml/VOTa
  <DESCRIPTION><![CDATA[
  <COOSYS ID="ICRS" system="ICRS" />
  <RESOURCE type="results">
    <DESCRIPTION>Native Saada</DESCRIPTION>
    <INFO name="QUERY_STATUS" value="OK" />
    <TABLE name="Results">
    <TABLE name="Spectra">
  </RESOURCE>
</VOTABLE>
```

**Raw VOTable (folded)**

```
<MODEL_INSTANCE name="MANGO" syntax="ModelInstanceInVot" >
  <GLOBALS>
  <TABLE_MAPPING tableref="Results">
  <TABLE_MAPPING tableref="Spectra">
</MODEL_INSTANCE>
```

**Mapping block (folded)**

- One mapping block for each <TABLE>
- Such placeholder makes easier the mapping insertion

# ModelInstanceInVot Guidelines

## ● #1 Shy

- Do **not break** working things
  - Model parser **should coexist with existing code**
  - Annotation **shouldn't alter the original data**

## ● #2 Readability

- Compactness
- Human readable and understandable
- Can be used at different levels
  - Get the type of the VOTable content
  - Get some meta data
  - Get everything through the model

## ● #3 Easy to implement

- Breakable in reusable components
- Component placement guided by the model