

# *Serializing Time Domain Data*

*A (Simplified) VO-DML Approach*

# Time Domain Use Cases

- **Purpose of the exercise**

- Science driven
- Investigating different ways to serialize time domain data
  - No prerequisite nor on the choice of the model neither on the annotation scheme

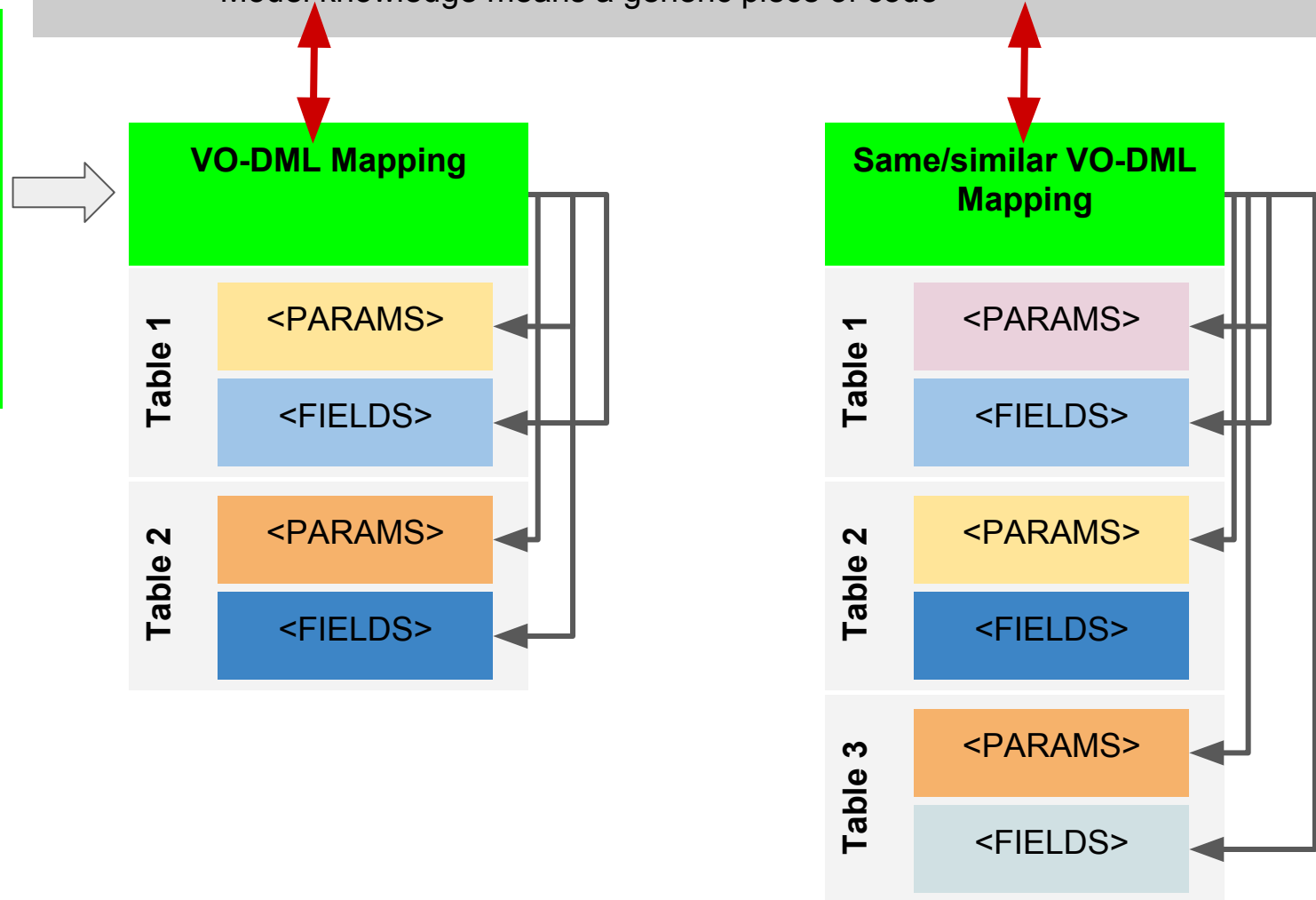
- **Sticking to Real Data**

- A few datasets proposed to test different annotations methods
  - Data distributed in multiple tables
  - Multiple light curves

## VO-DML Aware Client

- Only sees the mapping block
- No need to search annotations within tables
- The model knowledge is sufficient to process any table instance
  - Model knowledge means a generic piece of code

VO-DML  
representation of the  
model(s)



# VO-DML Mapping Strengths (DM1 session)

- **It works**

- Self-consistent framework
- Allows the reusability of models
- Client code available
- Available mapping tools and mapping helpers

- **Independant from the <TABLE> Blocks**

- Can be inserted in existing VOTables without other modifications
- No need to modify the VOTable schema
  - Just insert a sub-schema
- <VODML> annotations easy to locate for the client
- Ability to Retrieve Elements in Different Tables
  - Logical links between references

# Some Issues However

1. *A disadvantage*: The size of the mapping block
2. *Something more Basic*: The generation of the mapping blocks

# The Size of the Mapping Block

- **Example: BetaLyr\_Vizier\_ts.xml (on Volute time\_domain)**
  - A time series with 5 filters mapped on the `SimpleTimeSeries` model needs 775/1040 XML lines
    - Scientific quantities: 26 fields + some literals

```
5 <VODML>
6   <MODEL>
7     <NAME>coords</NAME>
8     <URL>https://volute.g-vo.org/svn/trunk/projects/dm/STC/vo-dml/STC_coords-v2.0.vo-dml.xml</URL>
9   </MODEL>
10  <MODEL>
11    NAME=... NAME=...
```

```
774    </REFERENCE>
775    </INSTANCE>
776  </COMPOSITION>
777  </INSTANCE>
778  </COMPOSITION>
779  </INSTANCE>
780  </TEMPLATES>
781 </VODML>
```

# The Size of the Mapping Block

## • Why is it so Long?

- Model is complex
  - STC + DataSet + CubeDM
  - Lots of abstractions and references
- Chatty mapping (personal point of view)
  - The mapping is a direct instantiation of the model
  - Some <ELEMENT> useless to extract data

## • Is that Size an Issue?

- Document size
  - Generation / storage / transfert
- Reliability
  - Mapping data is a complex process which needs to be checked: difficult here
  - Interoperability requires comprehensive messages

**It may be worth to consider a shorter syntax**

```
<INSTANCE dntype="cube:Observable">
  <!-- MAG -->
  <ATTRIBUTE dnrole="cube:DataAxis.dependent">
    <LITERAL dntype="ivoa:boolean" value="True"/>
  </ATTRIBUTE>
  <COMPOSITION dnrole="cube:MeasurementAxis.measure">
    <INSTANCE dntype="ts:spec.LuminosityMeasure">
      <ATTRIBUTE dnrole="ts:spec.LuminosityMeasure.type">
        <LITERAL dntype="ts:spec.LuminosityType" value="magnitude"/>
      </ATTRIBUTE>
      <ATTRIBUTE dnrole="meas:CoordMeasure.coord">
        <INSTANCE dntype="coords:GenericCoordValue">
          <ATTRIBUTE dnrole="coords:PhysicalCoordValue.cval">
            <COLUMN ref="Jmag" dntype="ivoa:RealQuantity"/>
            <!-- MAG Value -->
          </ATTRIBUTE>
          <REFERENCE dnrole="coords:CoordValue.axis">
            <FOREIGNKEY>
              <PKFIELD>
                <LITERAL dntype="ivoa:string" value="Magnitude_Axis"/>
              </PKFIELD>
            </FOREIGNKEY>
          </REFERENCE>
          <REFERENCE dnrole="coords:Coordinate.frame">
            <FOREIGNKEY>
              <PKFIELD>
                <LITERAL dntype="ivoa:string" value="_PhotFrame"/>
              </PKFIELD>
            </FOREIGNKEY>
          </REFERENCE>
        </INSTANCE>
      </ATTRIBUTE>
      <COMPOSITION dnrole="meas:CoordMeasure.error">
        <!-- MAG Error -->
        <INSTANCE dntype="meas:ErrorID">
          <ATTRIBUTE dnrole="meas:ErrorID.statError">
            <INSTANCE dntype="meas:SymmetricalID">
              <ATTRIBUTE dnrole="meas:SymmetricalID.radius">
                <COLUMN ref="e_Jmag" dntype="ivoa:RealQuantity"/>
              </ATTRIBUTE>
            </INSTANCE>
          </ATTRIBUTE>
        </INSTANCE>
      </COMPOSITION>
      <REFERENCE dnrole="ts:spec.LuminosityMeasure.photProv">
        <FOREIGNKEY>
          <PKFIELD>
            <LITERAL dntype="ivoa:string" value="_Phot_Info"/>
          </PKFIELD>
        </FOREIGNKEY>
      </REFERENCE>
    </INSTANCE>
  </COMPOSITION>
</INSTANCE>
```

# Mapping Generation: The Vizier or TAP Cases

- **Broad Variety of Data** (See Sebastien's talk Shanghai)
  - 14000 catalogues
    - Time data can have various formats and various locations in the VOTables
  - Dynamically generated data
    - TS generated from multiple source catalogues`
    - TAP reponses
- **Data Annotation Process Must be Adapted to Such Data Collections**
  - Not overloading the documentalist tasks
  - As much scriptable as possible.
  - Easy parsing for light weight clients
    - E.g. plotter embedded in a web interface

**This question must have a clear answer before to adopt any solution**

Crowd science is not a valid answer :=)



# Is a Simpler Mapping Possible?

- **Not Reinventing the Wheel:**

- Keeping the Actual VO-DML workflow
- Starting from the *vodml.xml* model representation
- Keeping the mapping structure
  - `<VODML><MODELS/><GLOBALS/><TEMPLATES/></VODML>`
- Keeping the class hierarchies in `<TEMPLATES/>` dedicated each to one `<TABLE>`

- **Simplifying the Data Binding**

- Only expose the model features necessary for the clients
- Hidden model features can be retrieved in the *vodml.xml* files referenced
- Mostly be derived from the syntax of the VO-DML mapping proposal

# Attributes

```
</INSTANCE>  
<INSTANCE dmtype="ds:experiment.ObsDataset">  
  <ATTRIBUTE dmrole="ds:dataset.Dataset.dataProductType">  
    <COLUMN ref="productType" dmtype="ds:dataset.DataProductType"/>  
  </ATTRIBUTE>  
  <ATTRIBUTE dmrole="ds:dataset.Dataset.dataProductSubtype">  
    <LITERAL dmtype="ivoa:string" value="Sparse Cube"/>  
  </ATTRIBUTE>  
  <ATTRIBUTE dmrole="ds:experiment.ObsDataset.calibLevel">  
    <COLUMN ref="calibLevel" dmtype="ivoa:integer"/>  
  </ATTRIBUTE>  
  <COMPOSITION dmrole="ds:dataset.Dataset.curation">  
    <INSTANCE dmtype="ds:dataset.Curation">  
      <ATTRIBUTE dmrole="ds:dataset.Curation.publisherDID">  
        <COLUMN ref="pupDID" dmtype="ivoa:anyURI"/>  
      </ATTRIBUTE>  
    </INSTANCE>  
  </COMPOSITION>  
</INSTANCE>
```



```
<INSTANCE dmrole='lm_timeseries:Timeseries.dataSet' >  
  <VALUE dmrole='lm_timeseries:dataset.DataSet.DataProductType' ref='_type' />  
  <VALUE dmrole='lm_timeseries:dataset.DataSet.dataProductSubtype' ref='_subt' />  
  <VALUE dmrole='lm_timeseries:dataset.DataSet.calibLevel' ref='_cal' />  
  <VALUE dmrole='lm_timeseries:dataset.DataSet.curation' ref='_cur' />  
</INSTANCE>
```

# Simpler References to Instances

```
<COMPOSITION dmrole="ds:dataset.DataID.creator">  
  <INSTANCE dmtype="ds:dataset.Creator">  
    <REFERENCE dmrole="ds:party.Role.party">  
      <FOREIGNKEY>  
        <PKFIELD>  
          <LITERAL dmtype="ivoa:string" value="_002J6U7FbgCwoWQF" />  
        </PKFIELD>  
      </FOREIGNKEY>  
    </REFERENCE>  
  </INSTANCE>  
</COMPOSITION>
```



```
<COLLECTION dmrole="ds:dataset.DataID.creator">  
  <INSTANCE dmrole="ds:party.Role.party" ref="_002J6U7FbgCwoWQF" />  
</COLLECTION>
```

# Toward Templates

- **Using XML element @attribute**

- Facilitate the usage of templates
  - Easier to change attribute values than XML nodes

- **One Tag to Grab Values**

- The same <VALUE> tag can be used to point on either <FIELD>, <PARAM> or literals

```
<VALUE dmrole='lm_timeseries:spaceaxis.RefFrame.frame' ref='SpaceReFrame' value='ICRS' />
```

# Mapping Block Template

- **Unresolved References to Data Replaced with String Patterns**

- Easy to process

- **One single <TEMPLATES>**

- the mapping template must be adapted by hand or by script

- **A (little) step toward automatic annotation**

- Template easy to process
- To be validated

```
<INSTANCE dmrole="cube:NDPoint.observable" dntype="cube:Observable">
  <VALUE dmrole="cube:DataAxis.dependent" ref="@@@@@"/>
  <INSTANCE dmrole="cube:MeasurementAxis.measure" dntype="meas:StdTimeMeasure">
    <INSTANCE dmrole="meas:CoordMeasure.error">
      <INSTANCE dmrole="meas:Error1D.ranError" dntype="meas:Symmetrical1D">
        <INSTANCE dmrole="meas:Symmetrical1D.radius">
          <VALUE dmrole="ivoa:Quantity.unit" ref="@@@@@"/>
          <VALUE dmrole="ivoa:RealQuantity.value" ref="@@@@@"/>
        </INSTANCE>
      </INSTANCE>
      <INSTANCE dmrole="meas:Error1D.statError" dntype="meas:Symmetrical1D">
        <INSTANCE dmrole="meas:Symmetrical1D.radius">
          <VALUE dmrole="ivoa:Quantity.unit" ref="@@@@@"/>
          <VALUE dmrole="ivoa:RealQuantity.value" ref="@@@@@"/>
        </INSTANCE>
      </INSTANCE>
      <INSTANCE dmrole="meas:Error1D.sysError" dntype="meas:Symmetrical1D">
        <INSTANCE dmrole="meas:Symmetrical1D.radius">
          <VALUE dmrole="ivoa:Quantity.unit" ref="@@@@@"/>
          <VALUE dmrole="ivoa:RealQuantity.value" ref="@@@@@"/>
        </INSTANCE>
      </INSTANCE>
    </INSTANCE>
  </INSTANCE>
  <INSTANCE dmrole="meas:CoordMeasure.coord">
    <INSTANCE dmrole="coords:CoordValue.axis">
      <INSTANCE dmrole="coords:ContinuousAxis.domainMin" dntype="ivoa:Quantity" abstract="true">
        <VALUE dmrole="ivoa:Quantity.unit" ref="@@@@@"/>
      </INSTANCE>
      <VALUE dmrole="coords:ContinuousAxis.cyclic" ref="@@@@@"/>
      <INSTANCE dmrole="coords:ContinuousAxis.domainMax" dntype="ivoa:Quantity" abstract="true">
        <VALUE dmrole="ivoa:Quantity.unit" ref="@@@@@"/>
      </INSTANCE>
      <VALUE dmrole="coords:Axis.name" ref="@@@@@"/>
    </INSTANCE>
    <INSTANCE dmrole="coords:Coordinate.frame">
      <INSTANCE dmrole="coords:domain.time.TimeFrame.refDirection" dntype="coords:domain.space.StdRefLocation">
        <VALUE dmrole="coords:domain.space.StdRefLocation.position" ref="@@@@@"/>
      </INSTANCE>
      <VALUE dmrole="coords:domain.time.TimeFrame.timescale" ref="@@@@@"/>
      <INSTANCE dmrole="coords:domain.time.TimeFrame.refPosition" dntype="coords:domain.space.StdRefLocation">
        <VALUE dmrole="coords:domain.space.StdRefLocation.position" ref="@@@@@"/>
      </INSTANCE>
    </INSTANCE>
    <INSTANCE dmrole="coords:PhysicalCoordValue.cval" dntype="ivoa:Quantity" abstract="true">
      <VALUE dmrole="ivoa:Quantity.unit" ref="@@@@@"/>
    </INSTANCE>
  </INSTANCE>
</INSTANCE>
```

# What I Did

- **Snippet Tests**

- Tested mapping features out of any model context

- **Test on Time Domain Data**

- Ad Hoc mapping of a shadow model
- Based on a simple model developed for that purpose (Im\_timeseries)
- Based on SimpleTimeSeries (MCD)
  - Rose a couple of issues

- **Python Client Consuming Annotated Data**

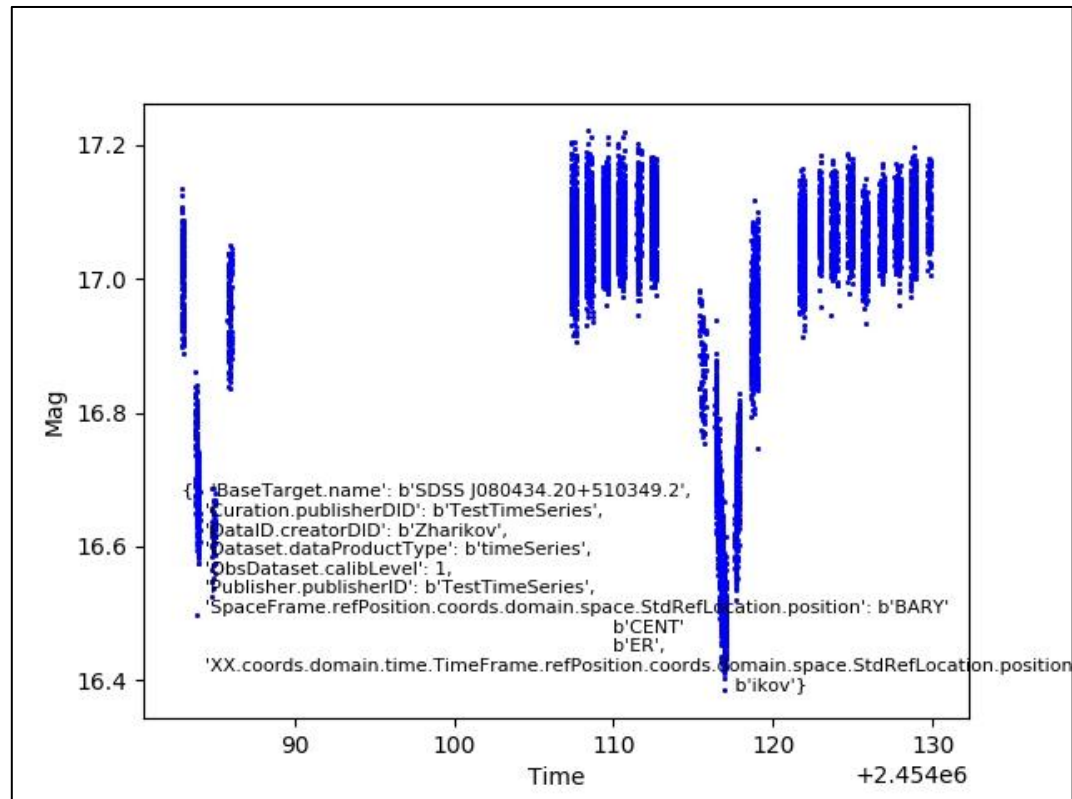
- Python code enable to data plotting

- **A Python Mapping Generator**

- Convert the vo-dml.xml model into a mapping block template
- Warrants the compliance with the VO-DML spec.

# Basic Time Series (Vizier SDSS)

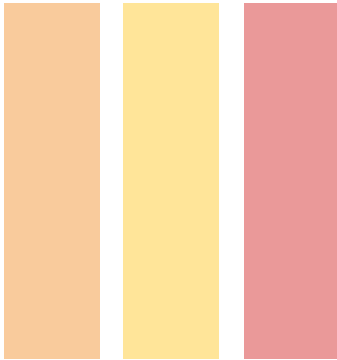
Meta data	
time	flux



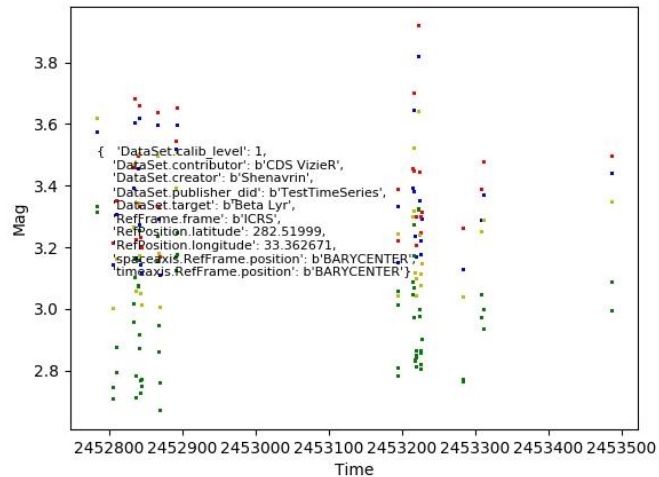
# Multiple Light Curves (Vizier BetaLyr)

Meta data

J H K



```
<TEMPLATES tableref="data">
  <INSTANCE dmrole='timeseries:data.Point'>
    <VALUE dmrole='timeseries:data.Point.timestamp' ref='JDJ' />
    <VALUE dmrole='timeseries:data.Point.observable' ref='Jmag' />
  </INSTANCE>
  <INSTANCE dmrole='timeseries:data.Point'>
    <VALUE dmrole='timeseries:data.Point.timestamp' ref='JDH' />
    <VALUE dmrole='timeseries:data.Point.observable' ref='Hmag' />
  </INSTANCE>
  <INSTANCE dmrole='timeseries:data.Point'>
    <VALUE dmrole='timeseries:data.Point.timestamp' ref='JDK' />
    <VALUE dmrole='timeseries:data.Point.observable' ref='Kmag' />
  </INSTANCE>
  <INSTANCE dmrole='timeseries:data.Point'>
    <VALUE dmrole='timeseries:data.Point.timestamp' ref='JDL' />
    <VALUE dmrole='timeseries:data.Point.observable' ref='Lmag' />
  </INSTANCE>
  <INSTANCE dmrole='timeseries:data.Point'>
    <VALUE dmrole='timeseries:data.Point.timestamp' ref='JDM' />
    <VALUE dmrole='timeseries:data.Point.observable' ref='Mmag' />
  </INSTANCE>
</TEMPLATES>
```





# @filter: Mixed Light Curves (Gaia)

Meta data

time	flux	filter
------	------	--------

		G
--	--	---

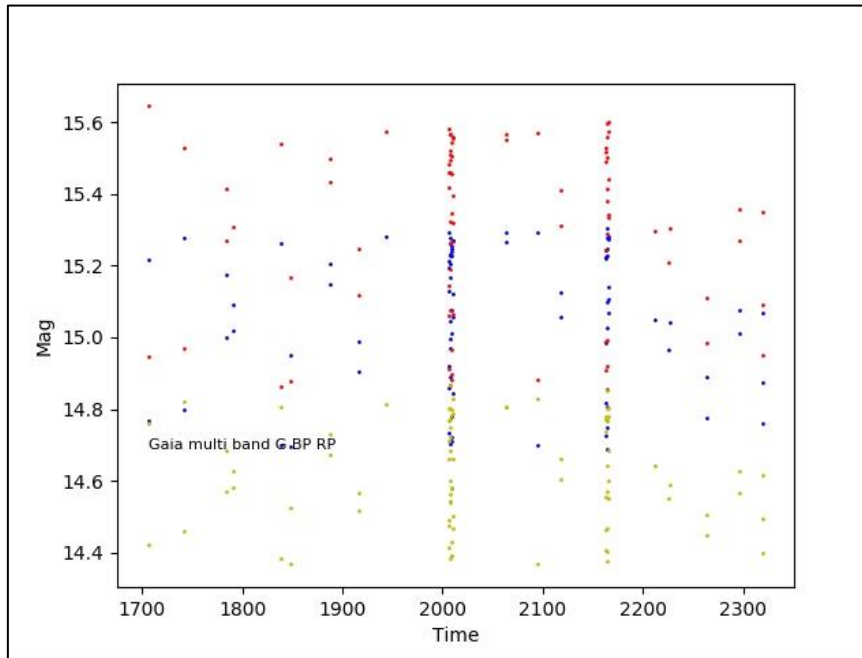
		G
--	--	---

		RP
--	--	----

		RP
--	--	----

		G
--	--	---

```
<INSTANCE dmrole="root" dmtype="lmsource:timeseries.point">  
  <VALUE dmrole="time_series:point.timestamp" ref="time"/>  
  <VALUE dmrole="time_series:point.magnitude.G" ref="mag" filter="@band='G'"/>  
  <VALUE dmrole="time_series:point.magnitude.BP" ref="mag" filter="@band='BP'"/>  
  <VALUE dmrole="time_series:point.magnitude.RP" ref="mag" filter="@band='RP'"/>  
  <VALUE dmrole="time_series:point.filter" ref="band" />  
</INSTANCE>
```



# Conclusions

## ● What Worked Out

- TDIG data challenge passed
- Generation of mapping block templates tested on various models
- Easy to gather data distributed in different tables
- Any piece of information can be retrieved
- VO-DML concepts not broken
- Mapping more compact (about 3x) and more readable

## ● Still to Do

- Cross-reference mechanism to improve
- Foreign keys just prototyped
- <COLLECTION> usage to be refined
- XML schema to write

## ● What is the Price for It

- Client code generation a bit less straightforward

# One Question and One Sketch

The question of the mapping generation process must have a clear answer before to adopt any solution

