

# SED Construction

DS5 – Cycle 1 review meeting

*Brice GASSMANN*  
*Sebastien DERRIERE*  
*Bernd VOLLMER*  
*Thomas BOCH*

# Summary

- Scope and goal of SED Construction
- Infrastructure setup
- Development of a SED Construction prototype tool

# Scope and goal of SED Construction

# Scope

- Definition:
  - *SED (Spectral Energy Distribution) Construction provides associations between wavelength and intensity of emission for objects in the sky.*
- SEDs are of great interest for astronomers.
- Three steps:
  - Find relevant VO resources according to some restriction like « *I want all resources referring to radio emission* ».
  - Extract relevant data from the obtained resources.
  - Merge data at different wavelengths

# The need for a registry

- The registry contains resources metadata that can help us to find relevant resources.
- Excerpt from the registry (tabular resource):

```
<vs:Resource>
...
<vs:table xmlns="http://www.ivoa.net/xml/VODDataService/v0.5">
  <vs:name>I/239/hip_main</vs:name>
  <vs:description>The Hipparcos Main Catalogue\vizContent{timeSerie}</vs:description>
  <vs:column>
    <name>recno</name>
    <description>Record number within the original table (starting from 1)</description>
    <unit/>
    <ucd>meta.record</ucd>
  </vs:column>
  <vs:column>
    <name>HIP</name>
    <description>Identifier (HIP number) (H1)</description>
    <unit/>
    <ucd>meta.id;meta.main</ucd>
  </vs:column>
  ...
</vs:table>
</Resource>
```

# Our goal

- To perform intelligent resources discovery in the scope of SED Construction
  - Exploits the registry to filter VO resources according to some criteria on its content (wavelength...).
  - Automatically extracts relevant data from the obtained resources.
- To comply with VO standards
- To make a prototype tool to demonstrate the feasibility

# Infrastructure setup

# A local registry

- Working on a local registry => Customizing it for our needs
- We chose the native XML database **eXist**.
- We populated the registry with data from the **Carnivore** NVO registry (>11.000 resources)
- Conversion from **UCD1** to **UCD1+** using a Web Service



# eXist

- An Open Source native XML database written in java
- Several interesting functionalities:
  - XQuery (W3C working draft 1.0 compliant)
  - Automatic node indexing
  - Users management (rights, passwords...)
- ***<http://exist.sourceforge.net>***

# Benchmarks

- For performances, tuning eXist is very important !
- Benchmarks after tuning:
  - Retrieving a single VO resource by identifier (among all resources):
    - `/vr:Resource[vr:identifier='ivo://CDS/VizieR/J/A+AS/123/575/levels3']`
    - Time: **2s**
  - Retrieving all the VO resources having specific UCDs:
    - `/vr:Resource[vs:table/vs:column/vs:ucd='phot.mag;arith.diff'  
or vs:table/vs:column/vs:ucd="phot.color;em.opt.B;em.opt.V"]`
    - Time: **3s**
- **Conclusion:** the time seems to be reasonable so we decided to develop a tool prototype making use of the registry.

# Development of a SED Construction prototype tool

# Tool introduction

- A java tool
- Why java ?
  - Java is **portable** => easy to diffuse to the VO community at large
  - Java has many **XML** treatment librairies.
  - XML databases often have a **java driver**.
- Using the ***XMLDB*** API to communicate with the eXist XML database (performing Xquery...)

# Registry metadata

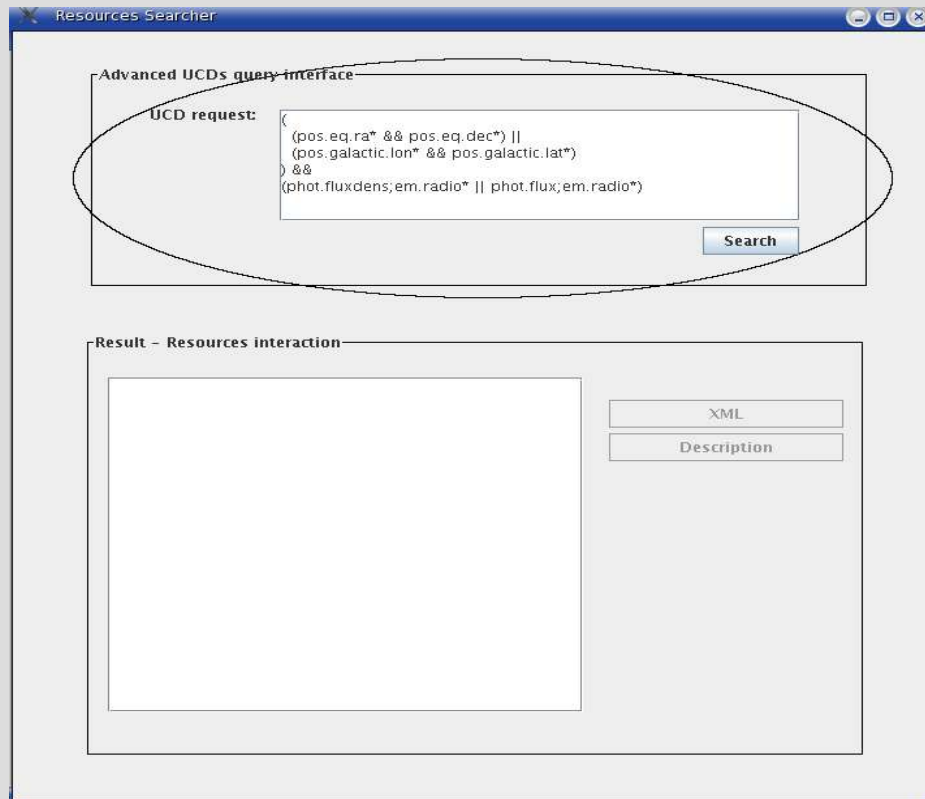
- What metadata are in the registry ? How can our tool interact with it ?
- Metadata for a tabular VO resource:
  - **identifier**: unambiguous identification of the VO resource (e.g. 'ivo://CDS/VizieR/I/239/hip\_main')
  - **ucd**: describes the content of a column (e.g. 'pos.eq.dec;meta.main')
  - **interface**: tells how a human or a tool can view the resource or extract data from it (there can be several interfaces for one resource)
- Exploiting those metadata:
  - Browsing all the UCDs of each VO tabular resource
  - **But**: a poor generic description of interfaces in the registry

# Functionalities

- Find relevant resources according to UCD's restrictions:
  - The astronomer must enter a logical condition on UCDs.
  - Then the tool searches in the registry and shows a list of the identifiers for the relevant resources.
  - **This functionality works with the current v0.10 registry schema.**

# Functionalities

- The astronomer enters the UCD logical condition:



$((ucd1 \text{ and } ucd2) \text{ or } (ucd3 \text{ and } ucd4))$   
and  
 $(ucd5 \text{ or } ucd6)$



Validate **Xquery** expression

# Functionalities

- A list of the relevant resources' identifiers is obtained:

The screenshot displays the 'Resources Searcher' application window. It features an 'Advanced UCDs query interface' with a text input field containing a query: `{ (pos.eq.ra* && pos.eq.dec*) || (pos.galactic.lon* && pos.galactic.lat*) } && (phot.fluxdens;em.radio* || phot.flux;em.radio*)`. A 'Search' button is located below the input field. Below the search interface is a 'Result - Resources interaction' section. This section contains a list of resource identifiers, such as `ivo://CDS/VizieR/J/A+A/391/967/table2`, `ivo://CDS/VizieR/J/A+A/391/967/table3`, and `ivo://CDS/VizieR/J/MNRAS/335/1085/table3`. To the right of the list are two buttons labeled 'XML' and 'Description'. At the bottom of the list, it states 'Found resources: 456'. A large, faint watermark 'SED' is visible in the background of the results area.

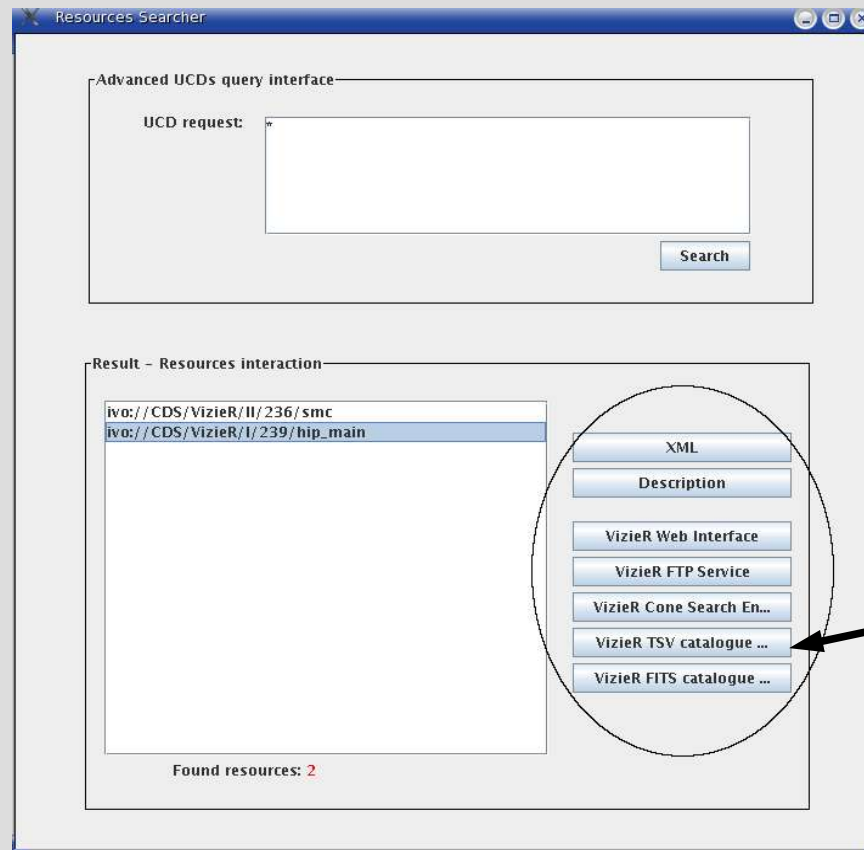


# Functionalities

- Interact with the retrieved resources and extract data from them.
  - By selecting a resource in the list the astronomer can interact with it:
    - Viewing the metadata of the resource in XML format
    - **Accessing all possible interfaces** of this resource (if an interface needs parameters it is possible to fill them)
  - The data extraction can be done via one of the interfaces (in general for each output format one interface is available)
  - **This functionality only works with our trial registry.**

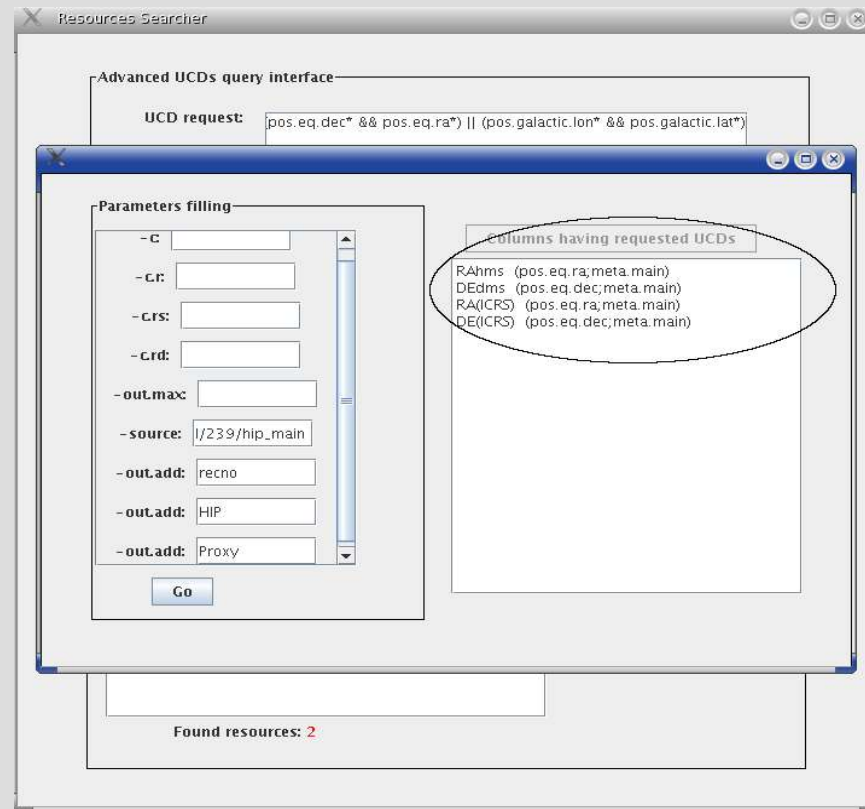
# Functionalities

- Once a resource is selected its interfaces appear:



# Functionalities

- For example one can use the « VizieR TSV Catalog » interface:



# Dynamic HTTP form construction from Registry resources

<Resource xsi:type="DataSet">

....  
 <interface xsi:type="vs:ParamHTTP" ref="ivo://CDS/VizieR/TSV">

```

  <vs:param>
    <vs:name>-source</vs:name>
    <xsi:value>I/239/hip_main</xsi:value>
    <vs:description>Name of the VizieR table to be queried. </vs:description>
  </vs:param>
  <vs:param>
    <vs:name>-out.add</vs:name>
    <xsi:value>recno</xsi:value>
    <vs:description>Add this parameter to the output table. </vs:description>
  </vs:param>
  <vs:param>
    <vs:name>-out.add</vs:name>
    <xsi:value>HIP</xsi:value>
  </vs:param>
  <vs:param>
    <vs:name>-out.add</vs:name>
    <xsi:value>H1P</xsi:value>
  </vs:param>
  </interface>
  </Resource>
  </i>
  
```

<Resource xsi:type="vr:Service" status="active" ... >  
 <identifier>ivo://CDS/VizieR/TSV</identifier>

```

  <vr:interface xsi:type="vs:ParamHTTP" qtype="POST">
    <vr:accessURL use="base">http://vizier.u-strasbg.fr/viz-bin/asu-tsv?</vr:accessURL>
    <vs:resultType>application/fits</vs:resultType>
    <vs:param>
      <vs:name>-c</vs:name>
      <vs:description>Target's name or coordinates</vs:description>
    </vs:param>
    <vs:param>
      <vs:name>-c.r</vs:name>
      <vs:description>Search radius</vs:description>
      <vs:unit>arcmin</vs:unit>
    </vs:param>
    <vs:param>
      <vs:name>-c.rs</vs:name>
      <vs:description>Alternate Search radius in arcseconds</vs:description>
      <vs:unit>arcsec</vs:unit>
    </vs:param>
    <vs:param>
      <vs:name>-c.rd</vs:name>
      <vs:description>Alternate Search radius in degrees</vs:description>
      <vs:unit>deg</vs:unit>
    </vs:param>
    <vs:param>
      <vs:name>-out.max</vs:name>
      <vs:description>Maximum number of output matches</vs:description>
    </vs:param>
  </vr:interface>
  </Resource>
  </i>
  
```

