Web services and XML

By Matthew J. Graham (Caltech, NVO)
Representing data

<table>
<thead>
<tr>
<th>Name</th>
<th>Distance</th>
<th>Brightness</th>
<th>Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sirius</td>
<td>2.64</td>
<td>-1.47</td>
<td>9940</td>
</tr>
</tbody>
</table>

- HTML focuses on presentation:
  `<TD>Sirius</TD>`
  `<TD>2.64</TD>`
  `<TD>-1.47</TD>`
  `<TD>9940</TD>`

- XML focuses on meaning:
  `<Source>`
  `<Star>`
  `<Name>Sirius</Name>`
  `<Distance>2.64</Distance>`
  `<Brightness>-1.47</Brightness>`
  `<Temperature>9940</Temperature>`
  `</Star>`
  `</Source>`

What do these values refer to?

Every value has a meaningful name

Some values can be complex
eXtensible Markup Language

- Ideal for encoding (meta)data
- “Read/write-ability”
  - When prettified can be interpreted by visual inspection
  - Simple XML can be composed with favourite editor
- Hierarchical
- OS/Platform agnostic: just text
- Extensible:
  - Every application can use a customized set of tags and structure
  - Foundation for additional capabilities for metadata definition and processing
- Uses in the VO:
  - Metadata exchange format
  - Service input/output message format
VOTable is XML

```xml
<?xml version="1.0" encoding="utf-8" ?>
<VOTABLE version="1.1" xmlns="http://www.ivoa.net/xml/VOTable/v1.1">
  <DESCRIPTION> Stars within 100pc </DESCRIPTION>
  <!-- A subset of the full Hipparcos catalogue -->
  <RESOURCE>
    <TABLE ID="Main">
      <FIELD name="HIP" ucd="meta.id;meta.main" datatype="int" width="6">
        <DESCRIPTION> Hipparcos Identifier </DESCRIPTION>
      </FIELD>
      <FIELD name="Plx" ucd="pos.parallax.trig" datatype="float" width="7" precision="2" unit="mas"/>
      <FIELD name="BTmag" ucd="phot.mag;em.opt.B" datatype="float" width="6" precision="3" unit="mag"/>
      <FIELD name="logTeff" ucd="phys.temperature.effective" datatype="float" width="7" precision="4" unit="[K]"/>
      <DATA>
        <TABLEDATA>
          <TR><TD>32349</TD><TD>379.21</TD><TD>1.43</TD><TD>3.9974</TD></TR>
          <TR><TD>71681</TD><TD>742.12</TD><TD>5.03</TD><TD>3.7364</TD></TR>
          <TR><TD>70890</TD><TD>772.33</TD><TD>17.39</TD><TD>3.4829</TD></TR>
        </TABLEDATA>
      </DATA>
    </TABLE>
  </RESOURCE>
</VOTABLE>
```

This is an XML file
It conforms to VOTable syntax
Element with simple content
Element with complex content
Element with no content
Comment
Attributes
Why HTML is not XML: well-formededness

- Document conforms to basic XML syntax
- Starts with `<?xml ... ?>` declaration
- Proper element syntax
  - All elements have closing tags:
    - With content: `<DESCRIPTION> text </DESCRIPTION>`
    - Without content: `<FIELD></FIELD` or `<FIELD/>`
  - All content fully enclosed – no overlapping content:
    - `<FIELD> <DESCRIPTION>text</FIELD> <DESCRIPTION>` ✗
    - `<FIELD><DESCRIPTION> text </DESCRIPTION></FIELD>` ✓
- One root element
  - `<xml version="1.0"?>`  `<xml version="1.0"?>`
  - `<VOTABLE> ... </VOTABLE>`  `<VOTABLE>`
  - `<VOTABLE> ... </VOTABLE>` ✗
    - `<VOTABLE> ... </VOTABLE>` ✓
- All attribute values are quoted
- Well-formed HTML is XHTML

28 March 2008

Aus-VO Summer School
XML Schema

- Defines set of allowed names and syntax
- Separate document that defines schema
- Several ways to specify:
  - Document Type Definition (DTD)
    ```xml
    <!DOCTYPE VOTABLE SYSTEM "http://us-vo.org/xml/VOTable.dtd">
    ```
  - (W3C) XML Schema
    ```xml
    <VOTable xmlns="http://www.ivoa.net/xml/VOTable/VOTable/v1.1">
    ```
  - RELAX NG, Schematron
- Validating an XML document
  - Software process to check that document conforms to the rule set declared in the schema
  - Downloadable XSV from same source for Python
VOTable as data format

<?xml version="1.0" encoding="utf-8" ?>
<VOTABLE version="1.1" xmlns="http://www.ivoa.net/xml/VOTable/v1.1">
  <DESCRIPTION> Stars within 100pc </DESCRIPTION>
  <!-- A subset of the full Hipparcos catalogue -->
  <RESOURCE>
    <TABLE ID="Main">
      <FIELD name="HIP" ucd="meta.id;meta.main" datatype="int" width="6">
        <DESCRIPTION> Hipparcos Identifier </DESCRIPTION>
      </FIELD>
      <FIELD name="Plx" ucd="pos.parallax.trig" datatype="float" width="7" precision="2" unit="mas"/>
      <FIELD name="BTmag" ucd="phot.mag;em.opt.B" datatype="float" width="6" precision="3" unit="mag"/>
      <FIELD name="logTeff" ucd="phys.temperature.effective" datatype="float" width="7" precision="4" unit="[K]"/>
      <DATA>
        <TABLEDATA>
          <TR><TD>32349</TD><TD>379.21</TD><TD>1.43</TD><TD>3.9974</TD></TR>
          <TR><TD>71681</TD><TD>742.12</TD><TD>5.03</TD><TD>3.7364</TD></TR>
          <TR><TD>70890</TD><TD>772.33</TD><TD>17.39</TD><TD>3.4829</TD></TR>
        </TABLEDATA>
      </DATA>
    </TABLE>
  </RESOURCE>
</VOTABLE>
Working with XML

- Navigating (XPath)
  \[/VOTABLE/RESOURCE/TABLE[@ID = "Main"]/DATA/TABLEDATA/TR[TD[1] = "32349"]/TD[4]\]

- Stylesheets (XSLT)
  - Set of templates to convert XML into other forms:
    <xsl:template match="TABLE">
      <xsl:for-each select="FIELD">
        <xsl:value-of select="concat(@name, `&amp;`)"/>
      </xsl:for-each>
    </xsl:template>

- Querying (XQuery)
  - What SQL is for a relational database, XQuery is for XML
    for $tab in /VOTABLE
    where $tab/RESOURCE/TABLE/@ID = "Main"
    return <Temperature>{$var/TR[TD[1] = "32349"]/TD[4]}</temperature>
Web Services

- A piece of software available over a network (preferably with a formal description of how it is called and what it returns that a computer can understand).
- XML is the data format of choice (see also JSON, YAML, ATOM, RDF/a)
Two species

- **RESTful**
  - Uses basic infrastructure of the Web:
  - HTTP Protocol
    - PUT: CREATE
    - GET: RETRIEVE
    - POST: UPDATE
    - DELETE: DELETE
  - Popular with Web 2.0

- **SOAP**
  - XML in, XML out
  - Formal description of interface for machines (WSDL)
  - Foundation for advanced functionalities:
    - Stateful services
    - Asynchronous messages
    - Secure access
    - Reliable messages
RESTful VO Services

- **Cone Search**
  - Search a source catalogue by position
  - Input is a “cone” – a position and a radius
  - [http://chart.stsci.edu/GSCVO/GSC22VO.jsp?RA=80.1&DEC=25.3&SR=0.02](http://chart.stsci.edu/GSCVO/GSC22VO.jsp?RA=80.1&DEC=25.3&SR=0.02)
  - All inputs in decimal degrees
  - Results returned in a VOTable

- **Simple Image Access (SIA)**
  - Search for images within a rectangular region
  - [http://adil.ncsa.uiuc.edu/cgi-bin/voimquery?POS=80,25.3&SIZE=0.5](http://adil.ncsa.uiuc.edu/cgi-bin/voimquery?POS=80,25.3&SIZE=0.5)
  - All inputs in decimal degrees
  - Additional parameters allowed (FORMAT, PROJ, NAXIS, etc)
  - May return standard images or custom cutouts
  - Two-phase process
    - Returns VOTable containing list of images and download URLs
    - Retrieve desired images

- **Simple Spectral Access (SSA)**
  - Search for spectra within a rectangular region, time and wavelength range
  - [http://.../ssa/queryData?POS=80.25,25.3&SIZE=0.5&BAND=2.7E-7/0.1&TIME=1998-05-21/1999](http://.../ssa/queryData?POS=80.25,25.3&SIZE=0.5&BAND=2.7E-7/0.1&TIME=1998-05-21/1999)
  - A lot more additional parameters
  - Two-phase process: queryData and getData
SOAP-based VO Services

- VO Registry
  - Search for data and services available in the VO
- VOSpace
  - Manage data stored in distributed locations
- OpenSkyQuery
  - Crossmatch catalogues
- WESIX
  - Object detection in an image
- Footprint services
  - Determine the sky coverage of a survey or overlaps between surveys and observations
- Spectrum Services
  - Access and work with spectra
- Cosmological Calculator (VOServices)
  - Determine cosmological distances
SOAP examples

**Request:**

```xml
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
  <soap:Body>
    <ComovingLineOfSight xmlns="http://www.skyservice.pha.jhu.edu">
      <z>float</z>
      <hubble>float</hubble>
      <omega>float</omega>
      <lambda>float</lambda>
    </ComovingLineOfSight>
  </soap:Body>
</soap:Envelope>
```

**Response:**

```xml
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns:xsd="http://www.w3.org/2001/XMLSchema"
               xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
  <soap:Body>
    <ComovingLineOfSightResponse xmlns="http://www.skyservice.pha.jhu.edu">
      <ComovingLineOfSightResult>float</ComovingLineOfSightResult>
    </ComovingLineOfSightResponse>
  </soap:Body>
</soap:Envelope>
```