

A Bit of GLUe for the VO: Aladin Experience

Pierre Fernique, André Schaaff, François Bonnarel, Thomas Boch

*Centre de Données astronomiques de Strasbourg, Observatoire de
Strasbourg, UMR 7550, 11 rue de l'Université, 67000 Strasbourg, France*

Abstract. In this article we describe how the GLU system (Uniform Link Generator) allows the Aladin browsing tool to integrate several image and data servers through a single interface. We describe how it works, how it is updated and how it is implemented in a Java context. We also present the future evolution we foresee for the GLU system in order to allow interaction with the emerging Web Services.

1. Interoperability Needs

Aladin is now widely known as a tool to display and cross-match heterogeneous data and images anticipating future VO portals. It offers transparent access to Simbad, VizieR, CDS image servers (DSS, MAMA, 2MASS), NED, SkyView and SuperCosmos databases, NVSS and FIRST archives, as well as mission logs such as CFHT, Chandra, HST, HUT, ISO, IUE and Merlin.

Like any interoperability tool, Aladin needs to answer the following questions:

1. How to access the data servers? What is the syntax of the query?
2. How to dynamically build interfaces to the data servers?
3. How to manipulate the server results? Which syntax/standard is used?
4. How to update the information required by the previous questions? Who has to do that?

Aladin makes use of the GLU system to answer these questions.

2. GLU Functionalities

The GLU system is a distributed repository of Web server descriptions. It was designed and written in 1996 by the Centre de Données astronomiques de Strasbourg (CDS). Presently, the GLU manages thirty replicated sites all over the world and about five hundred Web resource descriptions.

A GLU resource description, also called a GLU record, typically contains a unique identifier for the resource, a short description, a URL template to query the resource, and additional information describing the data type of the query parameters. The latter may include celestial coordinates, astronomical objects designations, bibliographic codes and so forth. The format of the results, VOTable, FITS image, HTML pages etc must also be included.

```

<RESOURCE ID="CDS/aladin/SSS.img">
  <DESCRIPTION>SuperCOSMOS image server- Edinburgh (UK)</DESCRIPTION>
  <QUERY>
    <URL>http://www-wfau.roe.ac.uk/~sss/cgi-bin/sss_aladin_pix.cgi?
      ra=$1&dec=$2&mime-type=image/x-gfits&
      x=$3&y=$4&waveband=$5</URL>
    <VAR NAME="1">
      <DESCRIPTION>Right Ascension (J2000)</DESCRIPTION>
      <TYPE REF="Coo(J2000,RA)"/>
    </VAR>
    <VAR NAME="2">
      <DESCRIPTION>Declination (J2000)</DESCRIPTION>
      <TYPE REF="Coo(J2000,DE)"/>
    </VAR>
    <VAR NAME="3">
      <DESCRIPTION>Width (arcmin)</DESCRIPTION>
      <TYPE REF="Field(RAm)"/>
      <VALUE DEFAULT="true">10</VALUE>
    </VAR>
    <VAR NAME="4">
      <DESCRIPTION>Height (arcmin)</DESCRIPTION>
      <TYPE REF="Field(DEm)"/>
      <VALUE DEFAULT="true">10</VALUE>
    </VAR>
    <VAR NAME="5">
      <DESCRIPTION>Waveband</DESCRIPTION>
      <VALUE DEFAULT="true">1 - UKST Blue (Bj)</VALUE>
      <VALUE>2 - UKST Red (R)</VALUE>
      <VALUE>3 - UKST Infrared (I)</VALUE>
      <VALUE>4 - ESO red (R)</VALUE>
      <VALUE>5 - POSS-I red (R)</VALUE>
    </VAR>
  </QUERY>
  <RESULT><CONTENT-TYPE>image/gfits</CONTENT-TYPE></RESULT>
  <DOC ROLE="user" HREF="http://www-wfau.roe.ac.uk/sss/" />
</RESOURCE>

```

Figure 1. Example GLU record for the SuperCosmos resource.

The GLU is distributed with a library toolkit in C and Perl to access the repository (*gludic tool*) and to generate the URL corresponding to a Web resource (*glufilter tool*). In this latter task, the GLU not only maps the query parameters passed by the user into the correct fields of the query URL template, but the GLU is also able to make the appropriate transformations to adapt the query parameters to the format and parameter types required by the remote server. For example, astronomical object identifiers are converted into their celestial coordinates, or the coordinates can be precessed and/or edited according to the server's requirements.

Furthermore the GLU has advanced functionalities such as:

- Mirror site management;
- Control of the GLU record distribution to any subset of the existing GLU domains (partial distributions);
- Remote access to the GLU functions via HTTP on each GLU site.

3. Aladin/GLU Algorithm

To understand how Aladin and GLU interact, we describe step by step the Aladin/GLU algorithm:

1. When it is launched, Aladin looks for the nearest GLU sites, either a local implementation (access by local GLU toolkit) or a remote implementation (access by CGI) depending on the user configuration and on the Java mode — Aladin may run as a Java standalone application, or as a Java applet;
2. Once the GLU site is located, Aladin asks for all the GLU records published in a particular GLU domain named ALADIN;
3. With these GLU records, Aladin dynamically builds the server access forms, using the description of each required parameter, their default values, the data type assigned to each of them and so on;
4. All server accesses are subsequently done by a GLU call (locally or remotely) which generates the appropriate URL which implies: locate the nearest mirror site, map the parameters according to the URL template, convert the parameters if necessary, generate proxy queries if Aladin is run in applet mode.

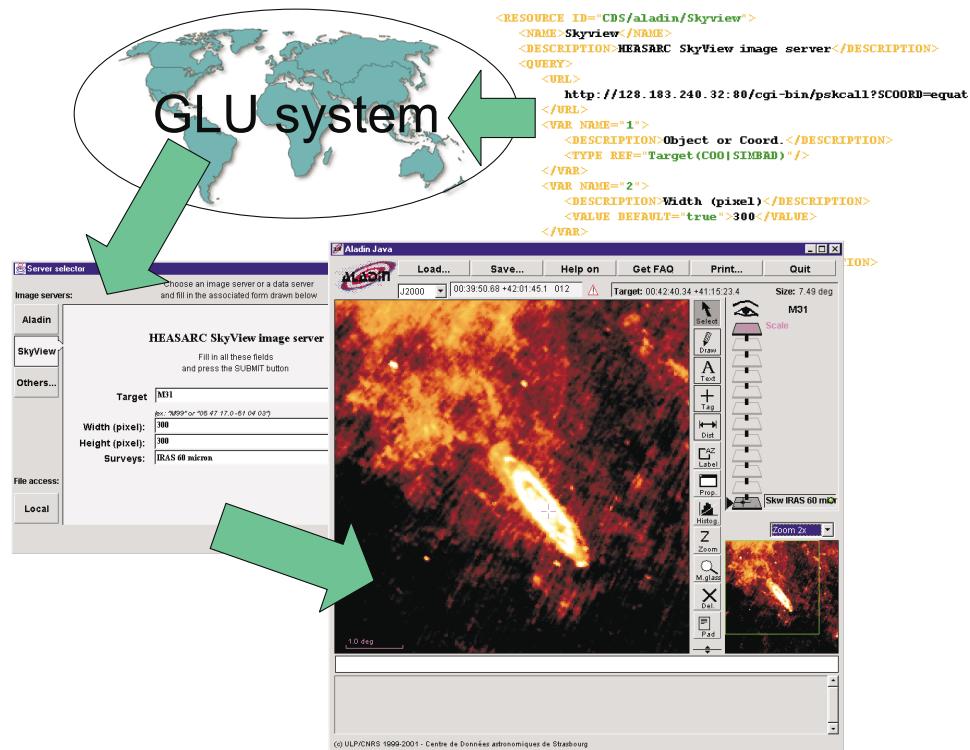


Figure 2. GLU/Aladin interactions

4. GLU and Web Services

With SOAP, WSDL and UDDI, the new Web Services will offer an alternative to describe and access Web resources. With these new standards we anticipate that servers will provide one or more SOAP options within the next two years in addition to the classical HTTP access.

In this context, we have planned to extend the next GLU release in three directions:

- The GLU will be able to distribute WSDL descriptions in order to offer an alternative to the UDDI mechanism for which the future seems uncertain;
- Each GLU site will install SOAP server methods in parallel to the classical CGI access, allowing GLU users to consult and use the GLU repository via SOAP Web services;
- Each GLU site will be able to query the Web server via SOAP and then translate the result into a classical plain ascii text stream — in other words, each GLU site would play the role of a gateway between the SOAP world and the basic HTTP world. This new functionality will allow VO clients such as Aladin to access SOAP servers right away without having to modify their code¹.

References

- Bonnarel, B. et al. 2001, in ASP Conf. Ser., Vol. 238, Astronomical Data Analysis Software and Systems X, ed. F. R. Harnden, Jr., F. A. Primini, & H. E. Payne (San Francisco: ASP), 74
- Ochsenbein, F. et al. 2000, in ASP Conf. Ser., Vol. 216, Astronomical Data Analysis Software and Systems IX, ed. N. Manset, C. Veillet, & D. Crabtree (San Francisco: ASP), 830
- Wenger, W. et al. 2000, A&AS, 143, 9
- Bonnarel, B. et al. 2000, A&AS, 143, 33
- Ochsenbein, F. et al. 2000, A&AS, 143, 230
- Fernique, P. Ochsenbein, F. Wenger, M. 1998, in ASP Conf. Ser., Vol. 145, Astronomical Data Analysis Software and Systems VII, ed. R. Albrecht, R. N. Hook, & H. A. Bushouse (San Francisco: ASP), 466

¹In order to create a new client to a SOAP service three steps are required: 1) retrieve the associated WSDL description; 2) generate the corresponding source code; 3) compile and plug it. These tasks can easily be done using a development environment such as AXIS or Visual Studio; in the context of a generic interoperability tool such Aladin, it is difficult to do it dynamically, especially in applet mode.