

## Searchable Registry for the National Virtual Observatory

Gretchen Greene, Bob Hanisch, Niall Gaffney  
*Space Telescope Science Institute*

William O'Mullane<sup>1</sup>  
*The Johns Hopkins University*

**Abstract.** As part of the NVO framework development initiative a prototype Astronomical Registry was designed for serving resource metadata across the Internet to the world community. While this registry incorporates many VO standard Cone Search and Simple Image Access (SIA) services it provides mechanisms for publishing custom archive services with associated metadata as well. The registry is mirrored at two sites, Space Telescope Science Institute and Johns Hopkins University, and additionally harvests resources at Caltech and NCSA<sup>2</sup> OAI<sup>3</sup> repositories. Web services and forms were implemented for independent higher level application integration with the registry such as the NASA Data Inventory Service (DIS). These interface methods provide fundamental add, edit and remove features as well as standard SQL query support. This registry is built with .NET technology integrated with MS SQL Server Database, IIS Web server, and C-Sharp product code.

### 1. Introduction

This registry is an early prototype for the NVO and IVOA infrastructure development. The initial registry developed by the NVO for Cone Search services was ingested and enhanced to provide additional metadata about each resource. There are several mechanisms in place which allow users to explore various capabilities such as networking remote registries to exchange resource content, querying for specific resource information, using web services and conventional web form access. Since this development occurred ahead of the finalization of the IVOA VOResource schema, we created a simplified metadata model to comply with the standard NVO Resource Metadata Document and created a class called SimpleResource. This class will be superseded in the near future to comply with the standard VOResource IVOA schemas.

---

<sup>1</sup>Department of Physics and Astronomy

<sup>2</sup>National Center for Supercomputing Applications

<sup>3</sup>Open Archives Initiative

## 2. Architecture and Design

The NVO has two types of registry, harvestable and searchable (the IVOA now foresees three types). We describe here the searchable registry and how it uses the harvestable registries. The notion is that harvestable registries should be easy to set up to allow organizations publish metadata with minimum cost, while the searchable registry will require some more effort. Clients would generally use the searchable registry to find what they are looking for.

We believe SOAP and WSDL (Web Services Description Language) provide an excellent medium for defining standard interfaces. Hence we designed the registry with a SOAP interface which allows querying and administration. We expose the power of the underlying SQL (Structured Query Language) based database systems through the SOAP interface.

We chose Microsoft's .NET framework to implement our service and used the SQL Server database to hold the meta information and facilitate querying. We knew this to be an efficient framework for implementing this type of system. Web methods are written in the C-Sharp language and the .NET project generates both web forms for more traditional browser usage as well as the SOAP based web service interface.

A more sophisticated set of web forms were built on the SOAP service using ASPX technology. This may be seen at <http://sdssdbs1.stsci.edu/nvo/registry>.

### 2.1. Web Services

The web service enabled features can be accessed by using any SOAP toolkit. One advantage being you can develop client applications in other languages such as Perl and JAVA. By using the conventional form for accessing the WSDL file, i.e. <http://sdssdbs1.stsci.edu/nvo/registry/registry.asmx?wsdl>, one gains access to the classes for accessing the service as well as the class which describes the returned XML, in this case SimpleResource.

### 2.2. Database

The SQL Server database was used to store the multiple resources and associated metadata. The database schema for this early prototype has a very simple structure. There is one primary resource table with the metadata content. Then several smaller tables were made for associating enumerated types for certain metadata elements i.e. service types. In the future we plan to modify this schema structure to correlate closely with the IVOA standard VOResource schemas.

While SQL Server is a relational database, there are multiple classes available in the .NET framework which facilitate database ingest of XML data types and retrieval into XML format. The main tool in use is the XSD.exe tool, equivalents in Java are Castor and JAXB (Java Advanced XML Binding). Initial versions of the VOResource (Plante et al. 2004) did not work well with these tools and was in a state of flux while the prototyping was underway. Hence for this prototype an XML parser was written as this was most resilient.

Another approach would be to use an XML database. Other groups are investigating this avenue. We feel performance will be an issue with the registry

as it will form the core of many VO services. Relational technology is well established and we shall continue on this route for now.

### 3. Functionality

#### 3.1. Searchable Registry

As mentioned earlier, this registry was designed to be a searchable registry and support basic SQL select queries. The web services have a direct connection to the database requiring no additional middleware layers to perform queries to the registry. The web form interface allows a user to type in a query string, this query string is sent directly to the web service and the results are rendered in an HTML table.

Any client may call the Web Service's QueryRegistry port directly passing the predicate to filter the desired resources. The resulting answers come back as XML SimpleResource. For multiple resources, there are multiple returned SimpleResources. If one is using a SOAP tool kit the XML is never seen, instead in one's code there will be an array of SimpleResource objects. More information about clients is given below in section 4.

#### 3.2. Harvesting

The Open Archives Initiative (OAI) protocol has been adopted by multiple NVO publishing Registries. Such as those developed at NCSA and Caltech. The searchable registry has the capability of harvesting OAI repositories to obtain new resource information. As described above an XML parser was written to ingest the XML coming from the OAI repositories. Currently we are working on using automated tools to parse the XML.

#### 3.3. Administrative

Several administrative features are available with the registry. Users may of course create new resources directly on the searchable site (they do not have to be harvested). When creating a resource a password is also provided. With the password a user may later modify or delete the resource. In the next version deletion will only be done with a flag - all records will be kept for posterity. We have not yet had a problem with fake resources and spam in the registry. We have no concerted plan for dealing with this eventuality. A rating scheme is under discussion - this would add recognized ratings/stamps of approval to registry entries. These could then be used in queries to filter for *approved* resources.

As the NVO and IVOA advance further, more sophisticated security methods will be implemented on an as needed basis. Security in SOAP services is a hot topic in the Web Services community but it appears a consensus between Grid and Web Services security may soon be reached. Until this is resolved it seems premature to implement strict certificate based security in our prototype registry.

## 4. Registry Clients

There are currently several clients using the prototype registry to find VO resources. We mention three here.

### 4.1. Data Inventory Service(DIS)

DIS (McGlynn et al. 2004) was the first client of the registry and the main driver for the searchable functionality. DIS provides a digest of VO information available about a source or positing on the sky by using the registry to find resources and then query the resources to confirm what information they have exactly. See <http://heasarc.gsfc.nasa.gov/vo/data-inventory.html>.

### 4.2. Web Form Builder

The query interface provided on the registry website is powerful but not so intuitive. STScI have funded the development of a query building interface for the registry. This is a JSP(Java Server Page) implementation which dynamically builds forms for a user then formulates the query and makes the SOAP request to the registry.

Try it at <http://tomcatdev1.stsci.edu:8080/voregistry/FormBuilder.jsp>.

### 4.3. Download Manager Mirage

Mirage is a powerful data analysis tool which has been wrapped with the IVOA download manager (Carliles et al. 2004) to allow easy loading of VO data. The download manager uses the registry to present the user with possible resources. It will then perform a cone search on the selected resources and load the data into mirage. Images also appear in the image module. The wrapped mirage is available at <http://skyservice.pha.jhu.edu/develop/vo/mirage>. The download manager is available separate (with source) to include in your own tool at <http://skyservice.pha.jhu.edu/develop/vo/ivoa>.

## References

- Carliles S., Kam Ho T. & O'Mullane W. 2004, this volume, 300  
McGlynn T., Lee J., Hanisch R., O'Mullane W. & Greene G. 2004, this volume, 319  
Plante R., Greene G., Hanisch B., McGlynn T., O'Mullane W., Williams R. & Williamson, R. 2004, this volume, 585