VOTECH
The European Virtual Observatory
VO Technology Center

DS5 – Intelligent Resource Discovery

Study Report

Editor: Sébastien Derriere, CDS
on behalf of DS5 co–workers

Contract Number: 011892

Project Website: http://eurovotech.org/

March 2008

Table of contents

Introduction

The first goal of DS5 is to undertake a **feasibility study** for developing components based on emergent technologies in the areas of the Semantic Web and Ontologies. This study will be used as a reference for actual **component designs and trial implementations**, and ultimately developments of standards at the IVOA level.
The **feasibility study** will consist in two parallel actions:

- surveying available techniques, standards and tools related to the Semantic Web and Ontologies, identifying their scope, community–support and limitations.
- identifying a list of possible science cases where intelligent resource discovery tools could be developed for astronomers, highlighting interactions with related IVOA working groups when relevant.

A short–list of **component designs and trial implementations** will stem from the combination of these two actions. Participants in DS5 will then focus on these selected science–cases. Building on the knowledge gained from these experiments, standards should be suggested for acceptance by the IVOA.

Because the semantic web and ontologies are active research topics, a technology watch for the evolution of techniques, standards and tools will have to be performed throughout the VOTECH project duration (e.g. the evolution of available techniques could allow during year 3 the development of applications that would have been judged unrealistic in Stage 01).

This document summarizes work done for DS5 until december2007. The work has been organized around two main topics:

- ontologies and knowledge bases (chapters 1 and 2)
- metadata management (chapters 3 through 5)

1 Ontologies exploration

The application of ontologies to astronomy is rather new. A first step was to review what had been already done with ontologies in other disciplines.

Two different axes were then explored for the construction of ontologies. The first one consisted in trying to derive ontologies from existing VO standards (STC, VOEvent, Characterization, Registry). The second axis was to build an ontology from scratch, and the topic of astronomical object types was chosen as a testbed. In both cases, the ontology development was done with a list of use cases in mind, so practical usage could be assessed.

1.1.Formal and Informal Ontologies

DS5 ontology work began with a survey of existing formal and informal ontologies across many academic disciplines. The survey included links to ontologies,

papers, discipline–specific use cases, and software incorporating ontologies, for example:

Formal Ontologies[1] :

- GO – Gene Ontology
- OBO – Open Biomedical Ontologies
- Oncology Ontology
- VSTO – Virtual Solar Terrestrial Observatory Ontologies
- Dublin Core

Informal Ontologies[2] :

- Biology – phylogenies
- Library systems – Dewey decimal, Library of Congress

1.2.Ontology Construction

After surveying ontologies in other discipline, the next task focused on construction of ontologies from existing virtual observatory data models using the Web Ontology Language (OWL) [1]. The Space–Time Coordinate (STC), VOEvent, and Characterisation schemas were identified as candidates for initial test cases [2, 3, 4]. Conversion from an XML schema to OWL format does not inherently provide additional exploitable metadata relationships. However, reformatting these three schemas as ontologies has allowed them to be imported into both knowledge bases and additional ontologies that define new metadata relationships.

The STC schema is a long, complex document with many substitution groups. Therefore, an early experiment in ontology construction focused on conversion of an STC UML diagram to OWL format. Arnold Rots provided a UML representation of STC formatted as a Microsoft Visio 2003 UML diagram. Using the Microsoft XMLExprt plug–in utility for Visio 2003, the UML diagram was converted to an intermediate XML Metadata Interchange (XMI) format. The resulting XMI file was imported into the Gentleware Poseidon for UML application for conversion to OWL [5]. Unfortunately, unresolved XMI formatting problems prevented successful conversion to OWL, so the experiment was abandoned. Further details about the UML to OWL experiment can be found at http://wiki.eurovotech.org/bin/view/VOTech/OntologiesFromUML.

As automated conversion from UML to OWL proved troublesome, the next step involved constructing ontologies by hand using the Protégé Ontology Editor (v 3.0 and 3.2 beta) [6]. The STC (v. 1.3), Characterisation (v. 0.95, 1.0, 1.11), and

1 http://wiki.eurovotech.org/bin/view/VOTech/SurveyFormalOntologies
2 http://wiki.eurovotech.org/bin/view/VOTech/SurveyInformalOntologies

VOEvent (v. 0.90, 1.0, 1.1) schemas were reconstructed as OWL–DL ontologies using the following guidelines:

- complexType elements containing other elements are treated as classes
- simpleType elements containing text are also treated as classes
- relationship between parent and child elements is described through object properties using the nomenclature “hasChildElement”
- elements with substitution group attributes are treated as subclasses of the class corresponding to the relevant substitution group
- attributes are treated as datatype properties using the nomenclature “attribute”
- additional OWL relationships such as datatype property enumerations or class restrictions, equivalences, and disjoints are used sparingly

The STC ontology was constructed first so that its corresponding OWL file could be imported into the VOEvent and Characterisation OWL files. Links to the original schemas, the constructed ontologies, and issues highlighted through each construction can be found on the following VOTECH wiki pages:

- http://wiki.eurovotech.org/twiki/bin/view/VOTech/StcOntology
- http://wiki.eurovotech.org/twiki/bin/view/VOTech/VoEventOntology
- http://wiki.eurovotech.org/twiki/bin/view/VOTech/CharacterisationOntology

1.3.Ontology of object types

a)Defined Ontologies Exploration

The ontology work in DS5 covers a wide spectrum of ontologies, associated technologies and their applications. Among these, an in–depth exploration of formal defined ontologies is performed.

These defined ontologies, while being more restrictive and difficult to build since they require formal definitions of the concepts, allow the use of automated inference tools ranging from consistency checkers to advanced semantic reasoning engines. This is especially interesting when considering databases since a semantic layer with such tools would allow automated consistency checks of the entries or advanced querying.

To experiment on these possibilities and the feasibility of a defined ontology–based system, a test case was chosen: an ontology of astronomical object types. Indeed the field is well–known, of manageable size, and related potential use–cases existed. Furthermore, standardizations of astronomical object types existed

and could be used as a starting point. The SIMBAD[3] database list of object types[4] was a good candidate since it was of good size and the goal was to create and test a knowledge engine to couple with databases.

The work on this ontology has led to an IVOA Technical Note [22] in which in-depth information on the ontology can be found. Additional information can also be found at:

http://wiki.eurovotech.org/twiki/bin/view/VOTech/OntologyOfObjectTypes

b)Ontology Construction

Building a defined ontology requires formalizing conditions and definitions on the ontology's concepts. Description Logics[5] is an adequate and mature means of representing such ontologies and the Web Ontology Language[6] (OWL) is based on description logics and is probably the most widespread language for implementing ontologies. As for the OWL flavor to use, OWL-DL and its recent evolution OWL1.1[7] were a natural choice since only them allowed enough expressiveness to build exploitable definitions while still being decidable. Moreover, both are well-supported by existing automated reasoners.

The ontology's construction was done by hand, using the Protégé-OWL editor[8], and relied on formalizing in description logics the knowledge on object types from both documentary sources and experts of this field. However, for both performance and maintenance reasons, the goal is to include all the knowledge to be used by applications but no more.

The guidelines for ontology construction were:

- Only add conditions on concepts that are always true. This is necessary to ensure correct inferences from the reasoner.
- As a consequence, conditions expressing possibilities have to be expressed backwards (e.g. It cannot be guaranteed that a given stellar object has an proper motion in the databases though it *can* have one, but it can be guaranteed that a proper motion is always associated with stellar or sub-stellar objects)
- The main hierarchy being based on subsumption (a more general/more specific relationship), relationships between compound objects and their components are to be represented with properties *hasConstituent/hasComponent/hasPortion* created towards this end.
- Reasoning complexity has to be kept low. This has led to avoid using qualified cardinality restrictions when possible, and avoid putting restrictions on

3 http://simbad.u-strasbg.fr/
4 http://simbad.u-strasbg.fr/guide/chF.htx
5 http://wiki.eurovotech.org/twiki/bin/view/VOTech/DescriptionLogics
6 http://www.w3.org/TR/owl-guide/
7 http://owl1_1.cs.manchester.ac.uk/
8 http://protege.stanford.edu/overview/protege-owl.html

enumerations or intervals. Testing on intervals or even enumerations can be externalized though, so it is possible to keep the complexity lower in the ontology without sacrificing such restrictions.

- The consistency of the structure and the performance level of the reasoning is to be checked as often as needed using the reasoner
- Keep the ontology application–oriented.
- To help linking real–world objects such as entries in object databases to the abstract concepts of the ontology, real–world data from databases such as measurements and labels is added or linked to the concepts using annotation properties.

Though still being polished, the resulting ontology covers the whole field of astronomical object types, most of them being at least partly defined. Also, externalizing the restrictions on intervals and optimizing some restrictions enabled keeping the reasoning times very low though the complexity of *SHOIN*[9], the description logic used, is exponential.

c)Implementation

Alongside with the ontology building, means of developing applications were set–up. This required mainly a reasoner and an API able to handle OWL–based ontologies manipulation and reasoner calls.

Reasoner

Choosing a reasoner required a thorough study[10]. It temporarily led to the choice of RACER[11]. But eventually Pellet[12] turned out to be better since it meanwhile reached higher levels of performance while having a much better support for non–commercial applications.

OWL API

The choice of an OWL API is basically a problem of compromise. On the one hand, until recently most developments were made using the Jena[13] RDF/RDFS Framework, but a great shortcoming is that it lacks specific primitives for OWL–based applications. OWL being an evolution of RDFS, it is possible to manipulate it with Jena, but at the cost of some heavy additional development.
On the other hand, most OWL API are young and still in alpha stages. Eventually the Protégé–OWL API[14] was judged the best compromise since it provides all the wished functionalities and is well supported, being derived from Jena and used as basis for the Protégé–OWL editor which is itself upgraded on a regular basis.

Use-cases Implementation

The Protégé–OWL API being written in Java, and the wish to be able to have the

9 http://en.wikipedia.org/wiki/Description_logic
10 http://wiki.eurovotech.org/twiki/bin/view/VOTech/InferenceEngineTests
11 http://www.racer–systems.com/
12 http://pellet.owldl.com/
13 http://jena.sourceforge.net/
14 http://protege.stanford.edu/plugins/owl/api/

test applications running as web services led to use Apache Tomcat as the web server and develop everything in Java / Servlet[15] / JSP[16], beginning with an extension of the API. This extension is designed for handling defined ontologies in conjunction with a reasoner and is not specific to the ontology of object types.

Additionally, a bridge to the Graphviz[17] representation software has been implemented to allow graph representations of data and especially parts of the ontology subsumption structure. It includes methods that automatically build from basic data a script in DOT language to be interpreted by Graphviz as well as methods to retrieve the image data within the Java program calling the bridge so that no knowledge of Graphviz itself is required to use it.

d)Application prototypes

Registry Request Builder

The first application exploiting the ontology of astronomical object types was a request builder for querying astronomical registries. The idea of such a tool came from the limitations of existing registry querying methods. Indeed, when putting conditions on object types within a registry query, one must use existing keywords of the registry. But the following problems arise when considering astronomical object types:

- Some object types do not have a keyword associated.
- More specific keywords are not taken into account in a broader query.
- All the keywords have to be selected manually by the user if he wants the best query possible.

The ontology's main relationship – the subsumption – is the one needed to retrieve more specific or more general keywords. Starting with the concept queried on, going down the subsumption leads to more specific concepts and going up the subsumption leads to broader concepts. Hence, if the concepts are tagged with registry keywords, harvesting more specific or more general keywords. Currently only the VizieR registry keywords were added as annotations to the concepts. Indeed, though the builder is not dependent on any specific registry it requires object types keywords to achieve some results, and VizieR was richer than most registries with regard to such keywords.

Starting from the concept queried on, the search for keywords is done in two steps:

- first find any keywords associated to the queried concept and any associated to more specific concepts
- Then, if no keyword has been found at this point another search is performed, this time to get the most specific subsumer having an associated keyword in order to be able to propose a query as close as possible to the original concept, albeit broader.

15 http://java.sun.com/products/servlet/
16 http://java.sun.com/products/jsp/
17 http://www.graphviz.org/

Ontology Explorer

Another application using this ontology is a prototype of ontology explorer which was designed both to allow browsing the contents of the object types ontology (or any other one) and to test the performance of reasoning engines when it came to identifying an unknown concept from conditions put on it.

The interface gives the details of the current concept. Conditions can be put on the concept using drop–down boxes which only allow building conditions from material the ontology and reasoner can relate to. Each time a concept is altered, be it an existing concept or a new one just added, the reasoner checks if the concept is still consistent.

Asserted knowledge on the current concept and knowledge inferred by the reasoner are shown separately. Additionally, a dynamic graph showing the neighborhood of the concept (the direct ancestors and children) is shown to help visualize the hierarchy within concepts (Fig. 1). The graph also shows which concepts are defined and which are not by using different colors and the current concept can be changed by clicking on the graph, which allows an easier navigation for users willing to browse the ontology without further needs like testing the performance of a reasoner.

Extensive tests have led to the conclusion that getting good inference results was highly dependent on the definitions of the concepts and the input data, which has led to building more adequate definitions for astronomical object types.

SIMBAD Consistency Checker

Cross-identification's Consistency Checker

A consistency checker for entries of the SIMBAD database is also being developed. Indeed, there are about 3.8 million objects in SIMBAD, each of which is tagged with *otypes* which are the SIMBAD object types keywords. But most of the time, only the main otype has been set by an expert, the other otypes are inherited: a SIMBAD object inherits the dominant otype of each catalog where it is referenced. Consequently if a catalog covers a field where very different object types are considered, this can lead to inconsistencies.

The reasoner is able to check the consistency of any new element with regard to the ontology. Therefore if a concept with the same characteristics as the SIMBAD item to check is created, its consistency can be checked with regard to the ontology, which is consistent itself.

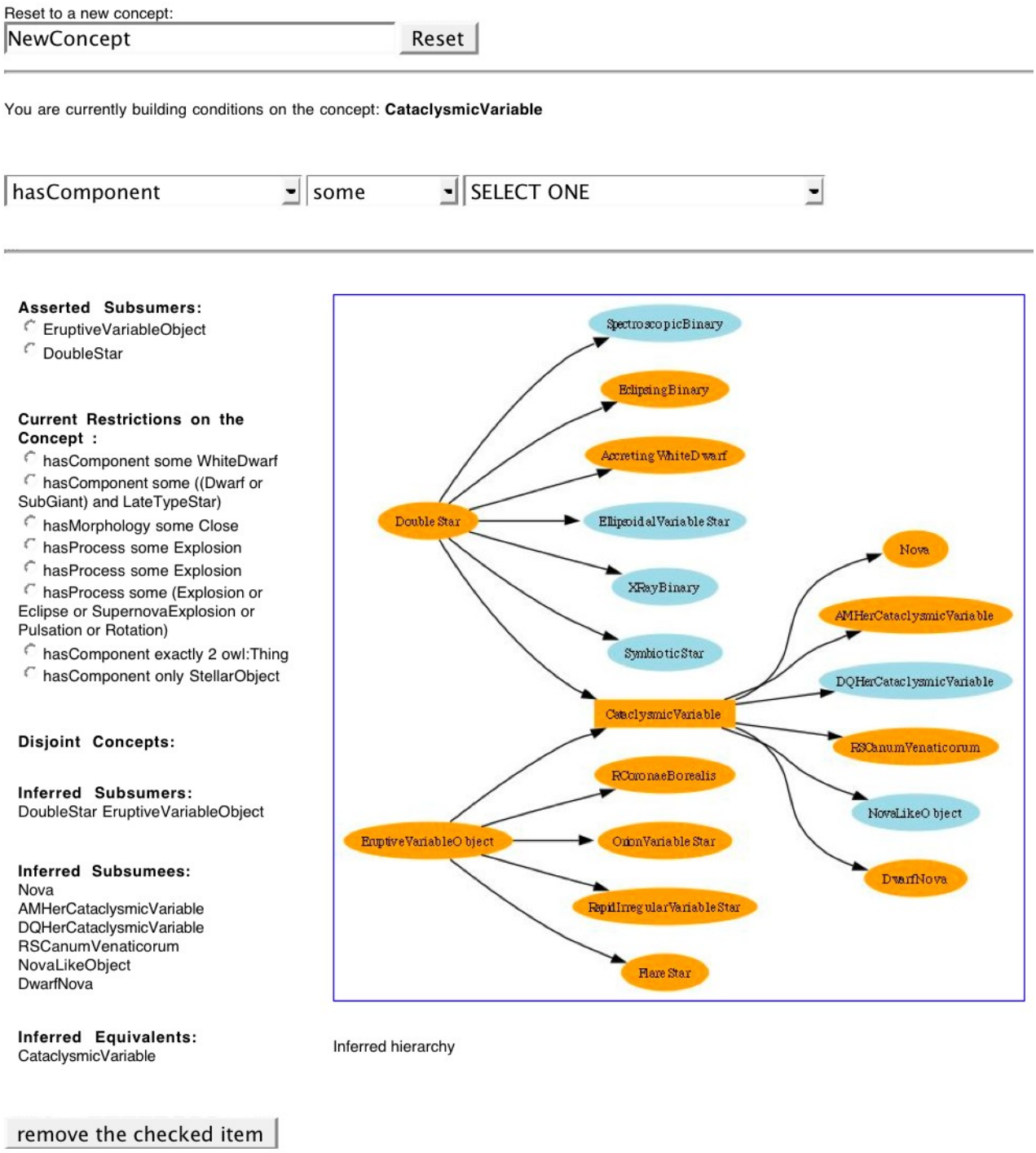


Figure 1: Screenshot of the concept explorer used to browse the ontology and showing the details and neighborhood of the concept CataclysmicVariable.

To create such a new concept conveniently, the concepts of the ontology have being annotated with their corresponding otypes. This way, the otypes from the SIMBAD entry provide a list of concepts that the new concept to check is to inherit from. After the check, if the concept is inconsistent, then the program indicates the inconsistent otypes (Fig. 2).

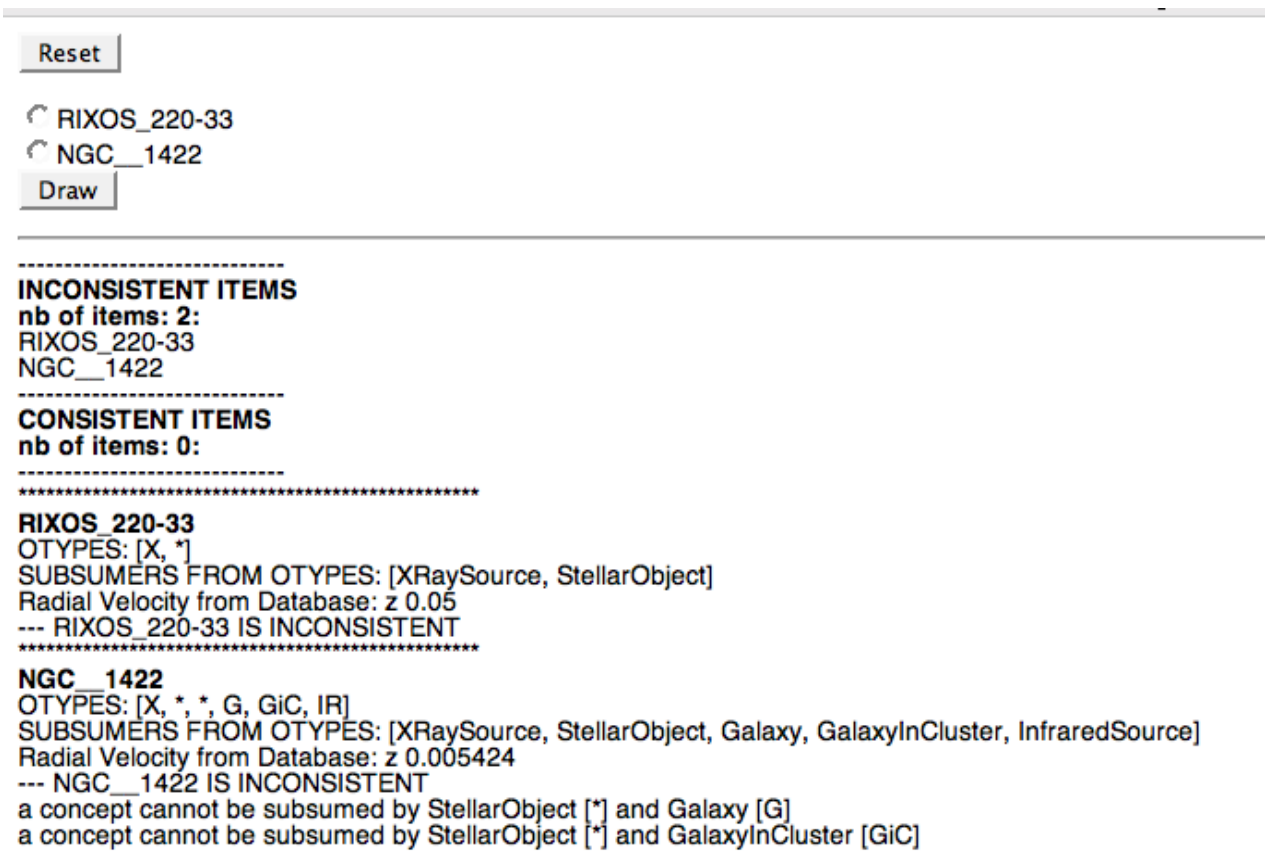


Figure 2: Screenshot of the SIMBAD Consistency checker on two items detected as inconsistent, the first for having a redshift value while its otypes state it should not have not and the second for being tagged both as a stellar object and a galaxy.

Components And Measurements Checker

Once the cross–identification has been completed, additional consistency checks have been implemented both to detect more potential inconsistencies but also give more explanations about found inconsistencies. This work has taken two directions.

The first part of this extension has been checking if inconsistent otypes are not the result of the merging of compound objects and their components (e.g. A double star and its main component). If the inconsistent otypes refer to concepts one of which can be a component of the other, then it is likely that there is no real inconsistency but rather a merging of the two.

The second is checking if measurements from the SIMBAD entry are consistent with the object type of the entry itself. Currently, the measurements taken into account are the redshift and radial velocities. If such measurements are found for a given entry, then they are checked with regard with the ontology. To be consistent with the ontology, it is impossible to have a radial velocity for extra–galactic objects or having a redshift for a star.

More information about the ontology of object types use–cases can be found at : http://wiki.eurovotech.org/twiki/bin/view/VOTech/OntologyOfObjectTypesUseCases

2. RDF Experimentation

2.1. Knowledge Bases and Queries

Once the STC, VOEvent, and Characterisation OWL files were constructed, a selection of virtual observatory datasets was collected in a central repository to test the metadata relationships encapsulated in the three ontologies. Two use cases were defined for semantic queries:

1. Solar: find coronal mass ejections that occurred within 24 hours of flares
2. Astrophysical: match VOEvent packets that fall within an astronomical dataset's observation time and right ascension (RA) and declination (Dec) box defined by a Characterisation file

Test XML files:

- Astronomical VOEvent packets from the OGLE and GCN feeds harvested from the eSTAR VOEvent broker.
- Solar VOEvent packets generated from SOHO–LASCO, NOAA GOES, and BATSE x–ray flare and coronal mass ejection catalogues
- 2MASS Characterisation files provided by Francois Bonnarel

OWL relationships can be used to query data formatted with the Resource Description Framework (RDF) [7]. One such query mechanism is SPARQL, a recursive acronym standing for SPARQL Protocol and RDF Query Languages [8]. Therefore, it was necessary to convert test data from native XML formats to RDF.

To convert VOEvent and Characterisation files from XML to RDF, the W3C RDF validator (http://www.w3.org/RDF/Validator) was used to hand–generate empty RDF templates (one for VOEvent files and one for Characterisation files). Next, two XSLT stylesheets were constructed to guide conversion from either VOEvent or Characterisation XML to the corresponding RDF templates. Mass conversion of the VOEvent and Characterisation files was achieved using a shell script that executed the command line xsltproc tool against each VOEvent and Characterisation XML file along with the appropriate XSLT stylesheet [9]. The resulting test dataset contains one RDF file for each original VOEvent or Characterisation XML file. Please see the process log at:

http://wiki.eurovotech.org/twiki/bin/view/VOTech/VoEventRDFnotes.

SPARQL queries can be executed in a knowledge base that contains both

ontologies and data in a triple format, such as RDF or N3 [10]. A data triple consists of a subject, a predicate (also called a property), and an object. The first knowledge base tested was Quaestor, a web application knowledge base developed during DS5 by Norman Gray [11]. For initial tests, specific elements (start and stop date, spectral unit and band pass, reference URI's, parameters, event concept, and event IVORN) from twenty VOEvent packets were encoded as N3 triples. The VOEvent OWL ontology plus a file containing the twenty N3–encoded events were uploaded to an MSSL Quaestor endpoint using HTTP put, and a series of SPARQL queries was executed, including

- Return all event IVORNs based on "ivorn" datatype property
- Return all coronal mass ejections based on event concept class
- Return solar x–ray flares of class "M" or higher based on event concept class and flare class datatype property

While these queries were successful, two issues arose with Quaestor: first, difficulty performing numerical and date based queries, and second, lack of data persistence following a restart of the host Tomcat application server. For further information, see http://wiki.eurovotech.org/twiki/bin/view/VOTech/VoEventSPARQLNotes.

Following recommendations at the 5th International Semantic Web Conference, Sesame was the next knowledge base chosen for RDF experimentation. Sesame is an open–source web application allowing persistent database storage of RDF data, RDF schemas, and OWL ontologies [12]. However, instead of SPARQL, Sesame uses the proprietary query mechanism SeRQL: the Sesame RDF Query Language [13]. A Sesame knowledge base was deployed at MSSL, and the three ontologies, characterisation RDF files, plus astronomical and solar VOEvent RDF files were uploaded once. SeRQL allows execution of queries on strings, numbers, dates, and multiple ontologies. An example of such a query selects filename, RA, and Dec from both Characterisation files and matching VOEvents where RA and Dec fall into a defined box, and the corresponding result is shown in Fig. 3:

```
select distinct X, RA, Dec
      from {X}
      charo:hascharacterizationAxis {characterizationAxis}
      charo:hascoverage {coverage}
      charo:haslocation {location}
      charo:hascoord {coord}
      stco:hasPosition2D {Position2D}
      stco:hasValue2 {Value2}
            stco:C1 {RA};
            stco:C2 {Dec}
      where RA > "270.0"^^xsd:float
            and Dec < "-29.0"^^xsd:float
            and RA < "270.5"^^xsd:float
            and Dec > "-29.5"^^xsd:float
      union
select X, RA, Dec
      from {X}
      voeo:hasWhereWhen {WhereWhen}
      voeo:hasObsDataLocation {ObsDataLocation}
      stco:hasObservationLocation {ObservationLocation}
      stco:hasAstroCoords {AstroCoords}
      stco:hasPosition2D {Position2D}
      stco:hasValue2 {Value2}
```

```
            stco:C1 {RA};
            stco:C2 {Dec}
      where RA > "270.0"^^xsd:float
            and Dec < "-29.0"^^xsd:float
            and RA < "270.5"^^xsd:float
            and Dec > "-29.5"^^xsd:float
      using namespace
voeo = <http://wiki.eurovotech.org/twiki/bin/viewfile/VOTech/VoEventOntology?rev=1;filename=VOEvent1.1.owl>,
stco = <http://wiki.eurovotech.org/twiki/bin/viewfile/VOTech/StcOntology?rev=3;filename=STC1.3.owl>,
charo = <http://eurovotech.org/twiki/bin/viewfile/VOTech/CharacterisationOntology?rev=1;filename=characterisation1.0.owl>
```

Links to the MSSL Sesame knowledge base along with twelve example queries can be found on the project wiki:

http://wiki.eurovotech.org/twiki/bin/view/VOTech/VoEventSERQLnotes.

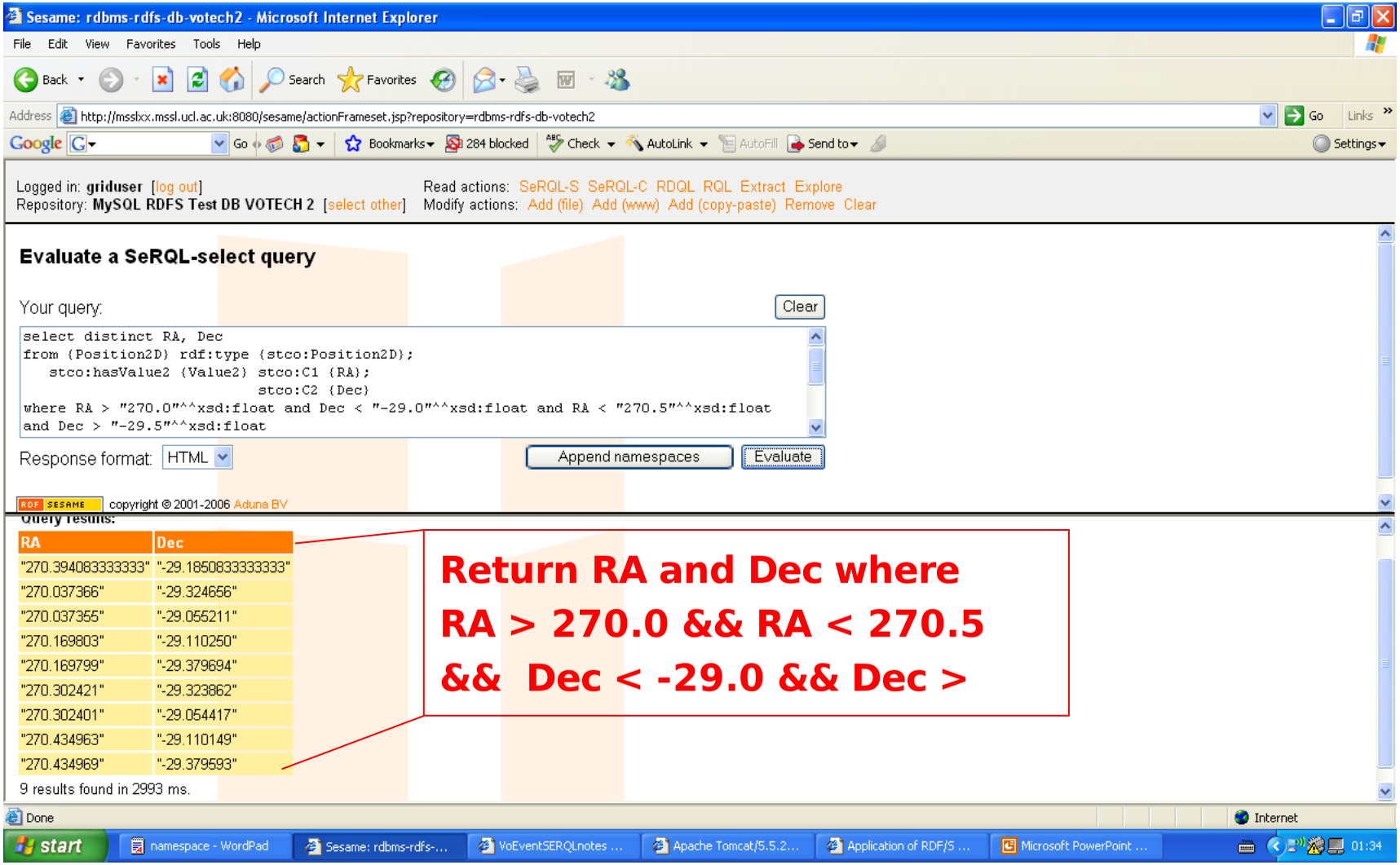
2.2.Related Work

Maintaining the STC, VOEvent, and Characterisation ontologies requires continuous effort as new versions of the corresponding schemas are released. Future VOTECH DS5 ontology work will concentrate on discovering new relationships between datasets. One method of evaluating the usefulness of such developments is to determine which queries can and cannot be replicated with SQL queries to the same data stored in a relational database instead of a knowledge base.

In order to replicate the example SPARQL and SeRQL queries with SQL, all astronomical and solar VOEvent packets described in section 2 were stored in a MySQL database. The database holds six tables, one each for the six test VOEvent feeds: OGLE, SDSS, GCN, SOHO–LASCO, NOAA–GOES, and BATSE. These tables store metadata for one event per row: IVORN, RA and Dec, start time, stop time, peak time, event concept, event name, packet author's name, packet author's email, a comma–separated list of parameters, and a comma–separated list of reference URI's.

Although the VOEvent database was initially created to serve as an SQL control group for semantic knowledge base queries, storage of start and stop times meant that the event data was suitable for remote searches through AstroGrid's Simple Time Access Protocol (STAP) web services [14]. STAP services were first deployed for each of the six static VOEvent sample sets. Following interest at the IVOA VOEvent working group's "Hot–Wiring the Transient Universe" workshop in June 2007, work was undertaken with Alasdair Allan to enhance parsing code in the eSTAR project's VOEvent perl client [15]. A client was installed at MSSL to listen for new VOEvent packets published by the eSTAR VOEvent broker, parse data from each XML packet, and load the event data into the appropriate MySQL table. The VOEvent STAP services at MSSL now provide searchable access to live VOEvent feeds in addition to historic VOEvents. A further two VOEvent STAP services were deployed at MSSL: the ROBONET feed of microlensing anomalies and the ESSENCE feed of supernova candidates.

Figure 3: Screenshot of MSSL Sesame knowledge base showing results of an RA and Dec box query submitted as a SeRQL select statement against VOEvents formatted as RDF inside the knowledge base



The VOEvent STAP feeds were registered with the central AstroGrid registry, and time–based queries to the HelioScope tool can return VOEvent packets. With development of the AstroGrid VOExplorer application, display of VOEvent STAP results was customized to show database values for curation metadata and event parameters along with further links to any reference URI's inside the packet [16]. Using the PLASTIC protocol, VOEvent reference files such as images, time series, or FITS files can be opened in a web browser, Aladin, TOPCAT, and other virtual observatory tools [17]. A tutorial for integration of time–based event data with VOExplorer, including deployment of the eSTAR VOEvent client, loading event data into MySQL, and deployment and registration of a STAP service, can be followed at http://wiki.astrogrid.org/bin/view/Astrogrid/VoEventSTAPtutorial. Alasdair Allan presented the work at a Euro–VO workshop on data publishing in the virtual observatory in July 2007.

A serendipitous opportunity for joint astronomical and solar use cases of VOEvents came with the spring 2007 announcement of the Heliophysics Knowledge Base (HPKB) hosted by Lockheed Martin's Solar and Astrophysics Laboratory (LMSAL). The goal of HPKB is to collect thirteen categories of solar events recognized by humans and automated detection algorithms [18]. Each event is encoded as a modified VOEvent packet using the IDL SolarSoft vobs:ontology package and then ingested to the HPKB database. The solar event knowledge base is being prepared in advance of the January 2008 launch of the Solar Dynamics Observatory mission, but it is currently being populated with events discovered from existing mission data and historical solar event catalogues. The production knowledge base will allow searches of solar, magnetosphere, and even astrophysical events.

Examination of the vobs:ontology modules revealed that the resulting solar VOEvents did not conform to the IVOA VOEvent standard. Therefore, work was undertaken to first develop a new XML schema encapsulating the mandatory and optional solar event parameters delineated at http://www.lmsal.com/helio–informatics/hpkb/VOEvent_Spec.html inside a "SolarEvent" element. Next, the VOEvent schema was redrafted to incorporate an optional "ExternalEventMetadata" element that can inherit the LMSAL "SolarEvent" element inside the VOEvent "What" element. The new VOEvent schema draft and LMSAL schema are currently under discussion by the HPKB project and the IVOA VOEvent working group. This work, along with links to the new schemas and example event packets, can be found at http://wiki.astrogrid.org/bin/view/Astrogrid/VoEventLMSAL.

2.3.Registry-related work

Ontologies and knowledge bases have also been built to test VO access control, using SPARQL queries and digital certificates for access control to VO resources[18].

An ontology of registry medata has also been developed to test SPARQL queries on VO registry browsing.

18 http://wiki.eurovotech.org/twiki/bin/view/VOTech/AccessControlUseCases

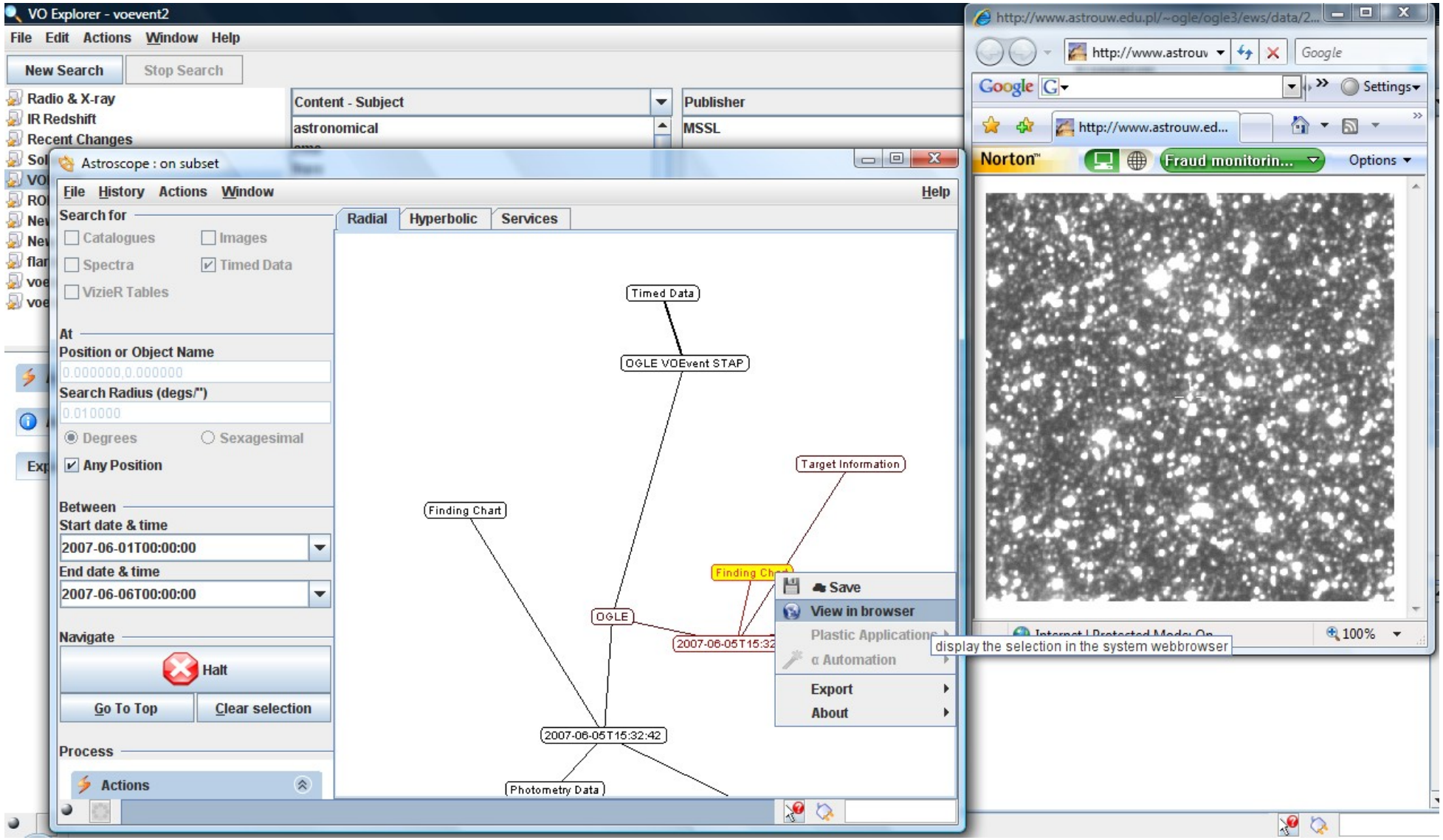


Figure 4: Screenshot of AstroGrid VOExplorer demonstrating results of a time-based STAP search for VOEvent packets. The reference files for an OGLE microlensing event include a finding chart .jpg image, which has been loaded into a web browser.

3. Data ingestion and homogenisation

MEx is a tool that has been developed to populate metadata repositories from the contents of FITS headers.

3.1.Motivation

Astronomy data products are typically stored in FITS format which also serves as container for meta data. Data and meta data generated by different detectors and processed by different software pipelines follow different conventions. Taxonomies or physical units are heterogeneous and incompatible.

In order to search and find data of interest it is necessary to describe them and store them in an homogeneous way. Intelligent resource discovery in particular can only be as good as the quality of the metadata on which it is performed. Moving from a technical observation log to an archive warehouse which characterizes observations in an instrument and observatory independent way is a demanding process. It requires machinery that homogenizes heterogeneous data sets from varying origin and at different reduction levels into an integrated search engine [19,20].

3.2.Existing tools

a)SAADA

SAADA (an existing tool available at http://amwdb.u-strasbg.fr/saada) version 1.1.1 was evaluated in 2005 to assert its applicability to the task. A detailed report was produced [23]

SAADA is a tool that aims at creating astronomical databases as easily as possible. It is a tool for the user who wants to build a repository for individual data products, without having to invest too much time and resources in setting up or even developing a new one.

Using the tool, users can:

- Create a database to store metadata describing those data.
- Ingest their products.
- Build relationships to link the data.
- Create a web interface to browse as well as publish data through a web server.
- Visualise data sets using Aladin.

The evaluated version of SAADA is a one–stop tool that can be used to quickly create an archive and to publish its content on the web. This makes it very convenient for individual projects to make their data available.

It lacks support for existing data models, and requires that all files are stored under its own repository. This can be problematic for data centres where an archiving system is already in place. It would be useful to be able to reuse parts of the tool, e.g. the metadata extraction, without having to include the whole application.

b)Recent developments

At present other tools exist that address the same issues, notably ESAVO's DALToolKit & DMMapper[19] and China–VO FitHAS[20], with varying degree of applicability.

DALToolKit is a one–stop tool specialised in SIA and SSA data ingesting and publishing. DMMapper enables mapping of existing database data into other conceptual data models, enabling a uniform access to heterogeneous data. FitHAS is a direct FITS header–to–database extraction tool. All mentioned tools are open source except for DALToolKit and DMMapper.

At the time of the study SAADA was the only available tool. Its strength is to provide an interface to small collections of homogeneous data sets and so MEx came about.

3.3.MEx

The MEx prototype was developed to solve the issues outlined. It supports astronomy data products like images and spectra that are stored in FITS format. This prototype was developed under the VOTECH project and made available to the community. The tool is currently being used as part of the ongoing data curation effort at ESO [21].

Data product description is defined independently of FITS keywords and instrument conventions by a dictionary of concepts (utype) and standard vocabularies (UCD). FITS values are extracted, transformed and mapped to those concepts by means of user–defined mappings, thereby homogenising the instrument and observatory incompatible conventions. Values are converted to the physical units (using CDS's unit conversion library – http://cdsweb.u–strasbg.fr/cdsdevcorner/units.gml) as defined by the concepts dictionary. Special purpose software modules can be hooked in where the mapping expressions are not sufficient to compute the desired values. Finally, values are stored in user–defined data models (not necessarily databases).

19 http://wiki.eurovotech.org/twiki/pub/VOTech/DS5PlanningStage06/DALToolKit__DMMapper.pdf

20 http://services.china–vo.org/fithas/

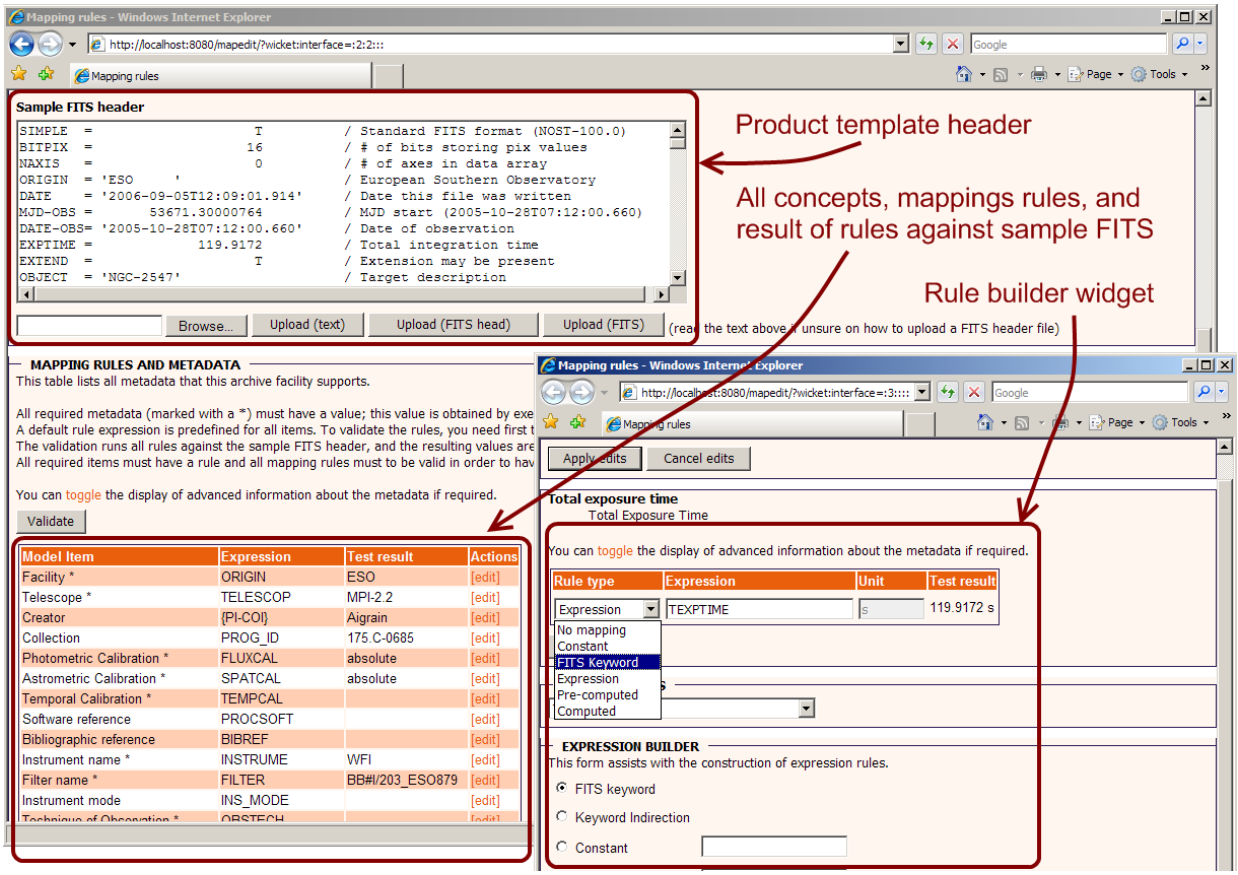


Figure 5: Mapping Editor showing a mapping for WFI/2.2 m data

A mapping editor graphical user interface was developed to assist users in defining mappings (Fig. 5). This interface enables a user to write mappings without requiring prior knowledge of the mapping rule syntax in most cases, and test the mappings against a sample FITS file provided by the user. It can be integrated into an existing web application through the HTTP POST method.

During development and in the scope of VOTECH collaboration, datasets provided by Anita Richards, Guy Rixon and Robert Mann from ASTROGRID were used for requirement definition and testing.

a)Typical use case

The typical use case for MEx is to populate metadata repositories of SIA and SSA compliant services. Dictionaries of concepts were defined to meet SIA and SSA required and optional metadata requirements, effectively populating those services repositories with data from several sources in a homogeneous way. To achieve this homogenisation, only the mappings for each data product must be edited.

MEx was bundled with independently developed SIA and SSA servers for the “EURO-VO Workshop on how to publish data in the VO” on June 2007, offering a one-stop solution for ingesting data and publishing it in a VO compliant way.

b)Architecture

To achieve the proposed goal in a generic way, usable by any data centre, MEx is split in two components, executed in sequence: keyword mapping and persistence (Fig. 6).

The keyword mapping component processes the FITS files, applies the mapping definition to them, and produces an in–memory list of all files and normalised keyword values.

A mapping definition is validated against a dictionary of concepts provided by the data centre, where required concepts and requirements for acceptable physical values must be met.

The persistence component stores this list in whatever format/database/etc a specific application might require; a particular data centre will customise this component to meet its data model. Some possible uses are:

- persisting to a VOTable
- persisting to an existing database structure
- persisting into the original FITS files

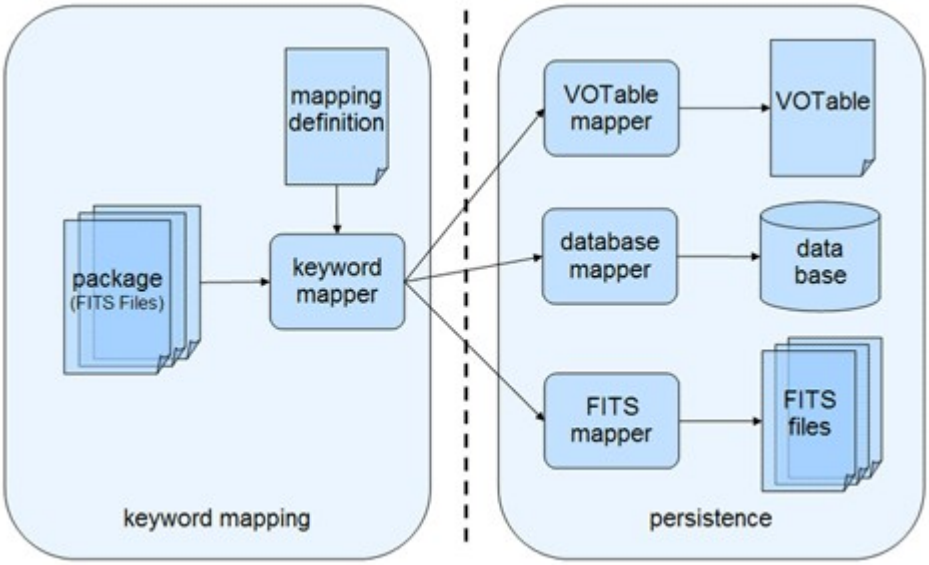


Figure 6: MEx architecture

Further technical details can be found on MEx page: http://wiki.eurovotech.org/twiki/bin/view/VOTech/MEx

c)MEx Features

I/O formats

- Input files are in FITS format, as it is the standard file format for astronomy data.
- Metadata persistence procedure is not dependent on database vendor or design (it might not even be a database).

Standard compliance and adaptability to existing systems

- IVOA efforts of metadata description (UCD, utype) are supported.
- Required metadata can be defined for each supported data product type (e.g. images, spectra).
- Metadata is persisted into a fixed database structure, but easily adaptable to existing data models defined by data centres

Metadata curation

- Support for physical unit definition by data centres.
- Values for each concept are stored under a unique physical unit, applying unit conversion where necessary.

Mappings

- Metadata present in FITS files must be ingested into an homogeneous representation, hence a mapping between the FITS metadata and the homogeneous metadata must be defined
- It is easy to define simple mappings where a FITS keyword maps to a concept, but that does not sacrifice the flexibility of defining complex mappings.
- The following mapping types are implemented:
    - Simple (keyword/value): The value is the content of a known FITS keyword.
    - Constant: A value is fixed across all the data to ingest
    - Unit Conversion: A value is present, but in different unit/formatting than the one expected
    - Arithmetic expressions, string concatenation
    - Choice: A value exists in one of several keywords
    - Keyword Indirection: A keyword contains a keyword name to lookup
    - FITS Extension Indirection: A keyword resides on an extension other than the current one.
    - File Indirection: A keyword resides on an external FITS file
    - Standard computations: values are absent from the header, but can be silently computed from the data itself and/or other values.
    - Pre–computed table: Fallback solution for when values are missing or too hard to get directly.

d)MEx Summary

To give easy access to archived data is one of the driving goals of the Virtual Observatory. The more data there is, the better, but also, the more data there is, the harder it is to find. Intelligent resource discovery tools will make data easier to find, but they need a homogeneous data repository to work effectively.

Hence the role of data homogenisation: large data holdings, especially in legacy archives, are heterogeneous by nature. The archival process keeps the observatory and instrument imprints contained in the data itself, hampering the capacity to discover and compare data, and ultimately slowing down the science.

Homogenization is a complex task requiring domain experts with access to the respective documentation. It is a task for data centres and required for enabling state of the art archival research. Doing it for all the different data in large legacy archives is a major challenge.

Tools such as MEx play a vital role in this process. Defining science–enabling curation metadata independently of the data and capturing the mapping with the existing metadata are the two required steps to create such high quality archives where scientists can find valuable data for their research.

4. Homogeneous data retrieval

Two different tools have been developed to locate relevant resources in the Registry, and to extract from these resources data in an homogeneous format. These tools require extensive usage of metadata in order to perform properly, and have been fine–tuned in a first step to catalogues from the VizieR collection, because they have all the required metadata.

4.1.Registry Query tool

The Registry Query Tool is used to search for specific resources in a Registry, based on their contents. The UCD that can be associated to tabular data description in the registry are used to locate relevant datasets.

The tool allows a user to find relevant VO resources from a registry thanks to UCDs, publishers and date of creation. A list of resources can be selected and saved for further processing with the Data Extraction Tool. To help the user in the selection of the resources, their metadata can be retrieved from the registry and visualized.

Fig. 7 shows the tool in action, with the query for UCDs that have to be present in the catalogue, and the corresponding list of resources from which the user can select a subset.

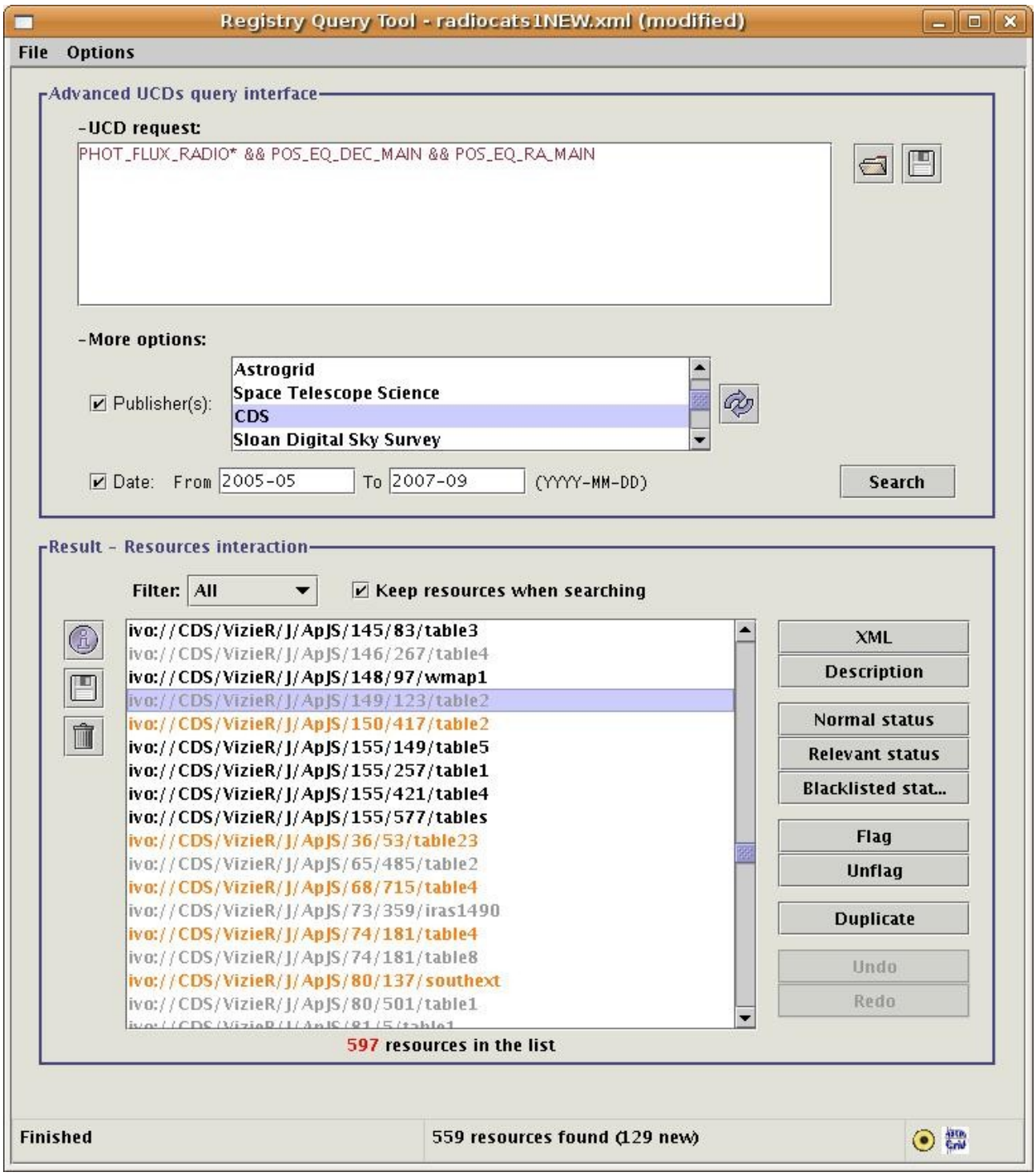


Figure 7: The Registry Query Tool: searching tables containing equatorial coordinates and a radio flux.

4.2.Data extraction tool

The data extraction tool can take as an input a list of resources, and helps extracting tabular data from those relevant resources, and transforming it in a uniform schema : same units, same columns names...

In addition, one can filter the sources one wants to keep in the output, one can generate new columns by combining input columns and one can define rules to

generate unique astronomical source identifiers. One can also choose the coordinate system one wants (B1950 or J2000), provided equatorial coordinates are available in the input resource.

The user first defines an output schema (Fig.8), with the desired column that have to be extracted, with the associated UCDs, units, format, etc...

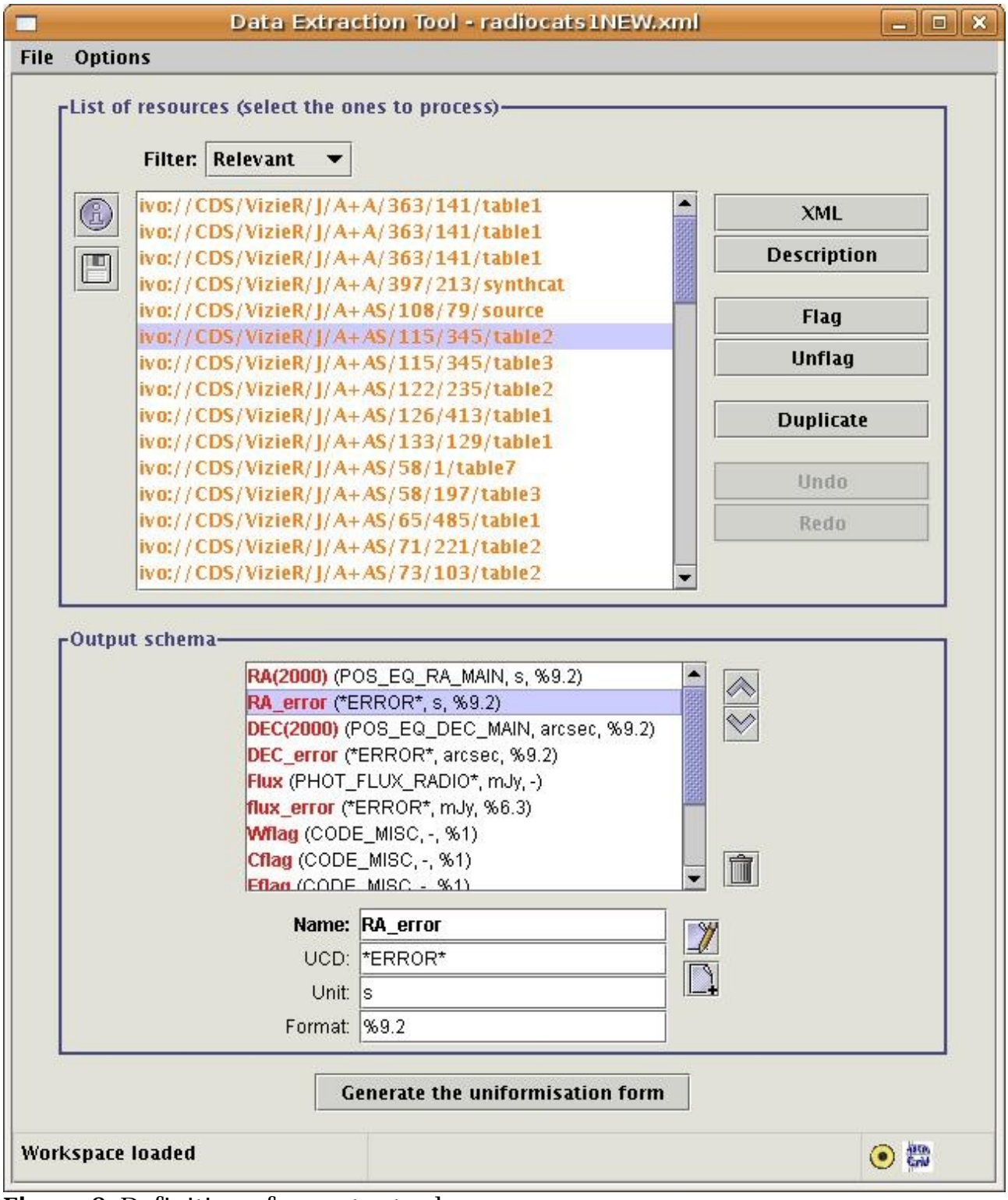


Figure 8: Definition of an output schema.

The tool attempts to find the best way to compute the desired quantities, perform unit conversion, etc... but the user can still customize the output (Fig. 9).

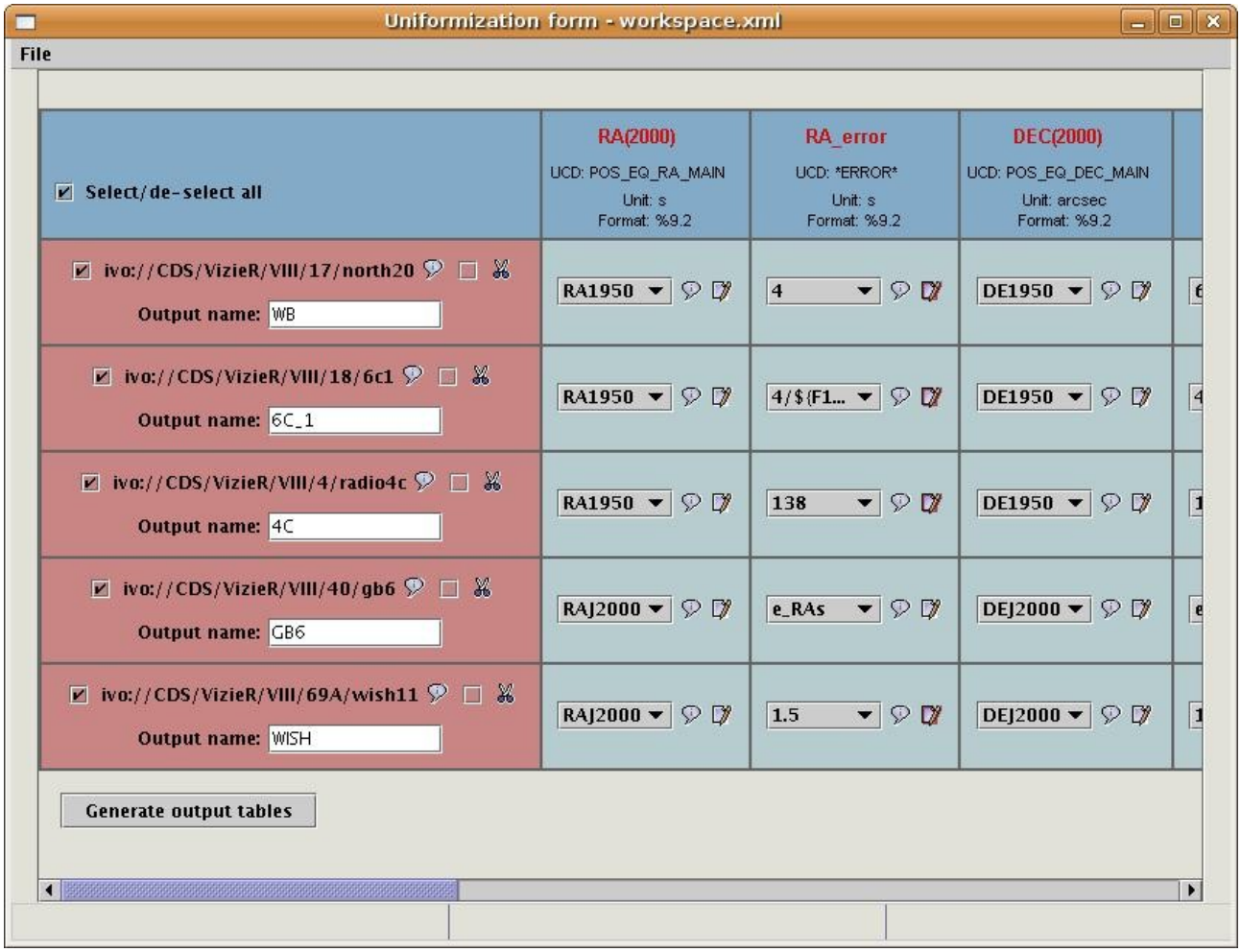


Figure 9: The Data Extraction Tool in action.

4.3. Application

Bernd Vollmer has applied the tools to the extraction of homogeneous sets of radio fluxes from VizieR radio catalogues. This is used as an input to the radio SED construction, and the SPECFIND[21] catalogue.

5. Object Names Recognition

The problem of automatically identifying astronomical object names in published papers is far from easy. There are thousands of different object names syntaxes in the Dictionary of Nomenclature of celestial objects, and potentially millions of valid identifiers.

A tool called DJIN has been developed to detect the object names in PDF versions of astronomical papers. The detected names are highlighted, and the annotated document helps librarians update the SIMBAD database with the correspondence between the object's identifiers and the bibliographic reference.

21 http://cdsarc.u-strasbg.fr/viz-bin/Cat?VIII/74A

DJIN includes a number of features:

- patterns available to process most journals (A&A, ApJ, AJ, MNRAS...) with different styles (depending on volume number)
- recognition of all formats in the Dictionary of Nomenclature
- identification of greek symbols as glyphs
- machine learning for improving performances (with the Weka software suite)
- detection of the location of the object name in the paper (title, abstract, text)

DJIN is now used routinely by the documentalists processing the journals for the SIMBAD database. They are now able to extract more information: the number of times, and location where an object name is cited in a paper can be computed, and therefore, a new piece of information becomes available. This should allow to estimate the importance of an object in a paper (if a name is cited in the title or the abstract, or if it is cited several times in the text, it is more important than if it is only cited once in a table).

This will open the path to some future intelligent bibliographic data mining.

6. Future work

6.1. VOEvent

During the next phase of VOTECH, Francois Bonnarel and Elizabeth Auden will conduct a joint DS5 / DS6 experiment with VOEvent and Characterisation data. The experiment's use case will identify astronomical datasets described by Characterisation that include serendipitous observations of events described by VOEvent. Two sets of test data will be uploaded to a Sesame knowledge base:

- VOEvent packets generated from historic catalogues of supernovae and gravitational microlensing events
- Characterisation files describing 2MASS, DENIS, and Schmidt plate observations taken during the same time frame as the test events

The historical supernovae and microlensing events will also be parsed into a MySQL database and deployed as AstroGrid STAP services. Semantic queries will be performed over the knowledge base to determine which events appear in one or more astronomical datasets using both VOEvent start and stop time metadata as well as the Characterisation "field–of–view" polygons rather than the simpler RA / Dec box searches executed during initial Sesame tests. When an event is matched with an astronomical dataset described by Characterisation, a new VOEvent packet with a "follow–up" attribute will be published that includes a reference URI citing the matching astronomical dataset.

6.2. Object types

During the next phase of VOTECH, the SIMBAD consistency checker will be improved by taking into account all the measurements that can be taken into account. The ontology has already been upgraded toward this end.

Since the ontology can be both used before and after a query to a database, experiments will be conducted to evaluate if and how it could help building advanced queries on object types, like it has already been successfully tested on the VizieR registry querying use-case.

Some work will be done on the RDF registry with Norman Gray since the prototype suggestions server requires interaction with an ontology-like structure of object types and database mappings to give out its full potential.

7. Dissemination

The work done in DS5 has led to some software releases, participation on some workshops, and publications.

7.1. Software

The ontologies that have been developed, the OWL files and the various pieces of software and their documentation are being made available through the project wiki pages:

http://wiki.eurovotech.org/twiki/bin/view/VOTech/ResourceDiscovery

7.2. Scientific events and workshops

DS5 co-workers participated in a number of events, either to learn new techniques, or to broadcast knowledge to the community. Reports from these events are available on the project wiki pages.

- The 8th international Protégé conference[22], Madrid, Spain (Jul 18-21, 2005) to learn about ontologies and the Protégé editor.
- The 4th International Semantic Web conference[23] (november 6-10, 2005), Galway, Ireland.
- European semantic web conference[24] (june 11-15, 2006), Budva, Montenegro
- The IAU meeting in Prague (august 17-18 2006), during the Special Session 3[25] (SPS3: The Virtual Observatory in action: new science, new technology, and next generation facilities), where the ontology of astronomical object types, and the registry query tool and data extraction tool were presented.
- The 5th International Semantic Web conference (november 5-9, 2006), Athens, Georgia, USA.

22 http://protege.stanford.edu/conference/2005/
23 http://iswc2005.semanticweb.org/
24 http://www.eswc2006.org/
25 http://www.ivoa.net/pub/VOScienceIAUPrague/

- European semantic web conference[26] (june 3–7, 2007), Innsbruck, Austria.
- The "*Hot–wiring the transient universe workshop*"[27] (june 4–7, 2007), Tucson, Arizona.
- The EURO–VO workshop on "*How to publish data to the VO*"[28] (ESAC, Madrid, June 25–29, 2007). A hands–on session using MEx was presented.
- The "*Practical Semantic Astronomy*"[29] (february 18–21, 2008) in Pasadena, California. Three VOTECH co–workers were involved in the Program Organizing committee and presented their work during the workshop.

7.3. Publications

Some activities in the Intelligent Resource Discovery study are closely related to some IVOA working groups. One of the first development was the production of a set of tools to handle UCDs, in relation with the UCD working group.

The ontology of astronomical object types was described in an IVOA note[30].

There were regular contributions to IVOA interoperability meetings, especially in the VOEvent group, the Data Model group (related to characterization), the UCD (which later became Semantics) group, with the production of a document related to vocabularies[31].

The MEx software was also the subject of a paper [21].

8. Conclusions

The two main axes that have been explored in the DS5 intelligent resource discovery study are related to ontologies and metadata. Ontologies have been built both from the conversion of XML schemas to OWL (section 1.2), and directly from scratch in the case of the ontology of astronomical object types (section 1.3). Converting XML schemas to OWL does not produce intelligent metadata relationships, but it allows to import them into knowledge bases or additional ontologies as has been shown in section 2. The combination of a knowledge base and a query mechanism (Quaestor and SPARQL, or Sesame and SeRQL) allows to answer some practical use cases.

Building an ontology from scratch allows to be very expressive in the definition of concepts. We have defined orignal methods to express constraints in the common case of missing measurements for some astronomical objects. The resulting ontology, coupled with a reasoner, can be applied to several use–cases where the development of alternate solutions (based for example on expert systems) would prove harder and less flexible.

The maintenance of an ontology on the long term is an important issue, for several reasons. For ontologies derived from XML schemas, the underlying standards are subject to evolutions, and therefore the corresponding ontology

26 http://www.eswc2007.org/
27 http://www.cacr.caltech.edu/hotwired/
28 http://cds.u–strasbg.fr/twikiDCA/bin/view/EuroVODCA/DcaMay2007Workshop
29 http://www.cacr.caltech.edu/semast/
30 http://www.ivoa.net/Documents/latest/AstrObjectOntology.html
31 http://www.ivoa.net/Documents/latest/vocabularies.html

would need to reflect these changes. Then, because ontologies are still an active research domain, the standards for the representation of ontologies are also evolving: the OWL reference is very likely to change in the future. And the existing tools or API are also evolving. Last but not least, the knowledge in a discipline like astronomy is also changing, and because ontologies are a representation of this knowledge, some curation has to be done to keep them up–to–date. But despite all these restrictions, the very nature of ontologies suggests that they can be very powerful in their application, because they can be used in very flexible systems, where changes in knowledge do not impact directly on the structure of the tools themselves.

Intelligent resource discovery requires good metadata. This means having the proper metadata assigned to the dataset, being able to convert a metadata format into another, with properly defined standards, and tools that are able to interpret and use these metadata as good as possible.

One of the first development in DS5 was the development of tools to help manipulation of the UCDs, an important piece of metadata standard of the IVOA. They can be used by data providers to assign relevant UCDs to their datasets, and curate these metadata, thereafter producing valid VOTable documents or Registry resource descriptions.

The MEx program also helps data providers homogenize the metadata, by automating the mapping from a set of FITS files to some accessible repository. This kind of tool helps bridging the gap between the wide variety and heterogeneity of basic data (here in the form of FITS files), and ideal homogenized (meta)data repositories with high level search and data mining capabilities.

The work on homogenized data retrieval, with the registry query tool and data extraction tool demonstrated the use of metadata (UCDs, but also units) to first locate relevant datasets in VO registries (based on the resource's contents), and then to extract data with an homogeneous format from potentially heterogeneous datasets. The tool uses metadata to free the user, as far as possible, from the painful task of explicitly stating every conversion from one parameter or unit into the desired one. Future tools should use such features as far as possible, so the user can focus on high–level requirements while the basic and repetitive tasks (units conversions, conversion from wavelength to frequency, coordinate conversions, etc...) are performed automatically by the system.

The tool for object names recognition, DJIN, does some advanced data mining in the published papers, and produces useful metadata relationships between bibliographic references and individual objects. This metadata will help searching for bibliographic references in a more precise way than was previously feasible.

Resource discovery is a challenge in a domain where the amount of data is growing exponentially. The transition from data to information, and from information to knowledge will require advanced techniques to explore, browse, organize... This will only be achieved if the data are described by proper metadata, and if users have a way to easily express their needs in terms of these metadata.

For the future of resource discovery, an important need will be the mappings between vocabularies. First, to allow an easy conversion from natural language to

standard sets of keywords or vocabularies, and second, to allow translation from one vocabulary to the other. It will then allow real efficient search amongst properly annotated datasets, using ontologies and knowledge bases to automatically suggest adaptative interpretations of the queries, and discover additional relationships.

9. References

[1] OWL – http://www.w3.org/TR/owl-guide/
[2] STC schema – http://www.ivoa.net/xml/STC/stc-v1.30.xsd
[3] Characterisation schema – http://www.ivoa.net/xml/Characterisation/Characterisation-v1.11.xsd
[4] VOEvent schema – http://www.ivoa.net/xml/VOEvent/VOEvent-v1.1.xsd
[5] Poseidon – http://www.gentleware.com/products.html
[6] Protégé – http://protege.stanford.edu/
[7] RDF – http://www.w3.org/RDF/
[8] SPARQL – http://www.w3.org/TR/rdf-sparql-query/
[9] Xsltproc – http://xmlsoft.org/XSLT/xsltproc2.html
[10] N3 – http://www.w3.org/DesignIssues/Notation3.html
[11] Quaestor – http://thor.roe.ac.uk/quaestor/http.html
[12] Sesame – http://www.openrdf.org/
[13] SeRQL – http://www.openrdf.org/doc/sesame/users/ch06.html
[14] STAP – http://wiki.astrogrid.org/bin/view/Astrogrid/SimpleTimeAccessProtocol
[15] eSTAR VOEvent module: http://sourceforge.net/projects/voevent/
[16] VOExplorer – http://wiki.astrogrid.org/bin/view/Astrogrid/VoExplorer
[17] PLASTIC – http://plastic.sourceforge.net/
[18] HPKB – http://www.lmsal.com/helio-informatics/hpkb/
[19] Leoni M., Dolensky M., et al. 2006, Multi-Purpose Metadata Repository for a Real and Virtual Observatory, ASP Conf. Ser., 351, 414
[20] Slijkhuis R., Delmotte N., et al. 2006, Feeding VO Data Products into the ESO Archive, ASP Conf. Ser., 351, 425
[21] Rite C, Slijkhuis R., et al. 2007, "Production of Previews and Advanced Data Products for the ESO Science Archive, ASP Conf. Ser. 2008 to be published.
[22] http://wiki.eurovotech.org/twiki/pub/VOTech/OntologyOfObjectTypes/ObjectType_TN_2007-03-08.pdf
[23] http://wiki.eurovotech.org/pub/VOTech/DS5Plan01/saada-report.pdf

10. Glossary

2MASS	Two-Microns All Sky Survey
BATSE	Burst And Transient Source Experiment
CDS	Centre de Données astronomiques de Strasbourg
China-VO	Chinese Virtual Observatory
DENIS	DEep Near-Infrared Southern Sky Survey
DJIN	Detection in Journals of Identifiers and Names

ESAVO	European Space Agency Virtual Observatory
ESO	European Southern Observatory
FITS	Flexible Image Transport System
GCN	Gamma–ray burst Coordination Network
GOES	Geostationary Operational Environmental Satellite
HPKB	HelioPhysics Knowledge Base
HTTP	HyperText Transfer Protocol
IVOA	International Virtual Observatory Alliance
IVORN	International Virtual Observatory Resource Name
LASCO	Large Angle and Spectrometric Coronagraph Experiment
MSSL	Mullard Space Science Laboratory
N3	Notation 3
OGLE	Optical Gravitational Lensing Experiment
OWL	Web Ontology Language
RDF	Resource Description Framework
SIA	Simple Image Access
SIMBAD	Set of Identifiers Measurements and Bibliography for Astronomical Data
SPARQL	SPARQL Protocol and RDF Query Language
SSA	Simple Spectral Access
STAP	Simple Time Access Protocol
STC	Space Time Coordinates
TOPCAT	Tool for OPerations on Catalogues And Tables
UCD	Unified Content Descriptors
UML	Unified Modeling Language
URI	Uniform Resource Identifier
VO	Virtual Observatory
W3C	World Wide Web Consortium
XMI	XML Metadata Interchange
XML	eXtended Markup Language
XSLT	eXtensible Stylesheet Language Transformation

Acknowledgments

Many thanks to all the co–workers of this Design Study. This includes Elizabeth Auden, Norman Gray, Alasdair Allan (Astrogrid), Alexandre Richard, Andrea Preite–Martinez (INAF), Soizick Lesteven, Christian Bonnin, Anais Oberto, Brice Gassmann, Bernd Vollmer (CDS), Nausicaa Delmotte, Marco Leoni, John Lockhart, Remco Slijkhuis, Dieter Suchar, Bruno Rino (ESO).