Astronomical Catalogues - Simultaneous Querying and Matching

H.-M. Adorf, G. Lemson, W. Voges *Max-Planck-Institut für extraterrestrische Physik, Garching, Germany*

H. Enke, M. Steinmetz *Astrophysikalisches Institut Potsdam, Germany*

Abstract: *We report on our experience in trying to execute multiple simple cone searches on a variety of published astronomical catalogues. The individual search results are fed into a catalogue matcher developed by GAVO. The matcher attempts to perform a probabilistic "fuzzy join" based on sky positions and their uncertainties. We describe current features of the GAVO architecture that support such simultaneous queries, and outline some requirements for future versions.*

Goals

The German Astrophysical Virtual Observatory (GAVO) is pursuing a project with the following goals:
- In the long-term to construct a multi-band spectral energy distribution (SED) from various catalogues, useful of source identification and classification purposes;
- In the medium term to search for exotic objects like isolated neutron stars, brown and white dwarfs;
- In the short-term to set up an infrastructure that allows exercising the existing simple cone search services (and thereby to find out what works and what not yet).

To this end GAVO is developing a multi-catalogue multi-cone (MCMC) search service feeding a probabilistic source matcher.

Architecture

The overall architecture of the MCMC search and matching service is depicted in Fig. 1. There are three major building blocks:
- the multi-catalogue multi-cone search "download manager",
- the VOTable processor, and
- the probabilistic matcher.

The MCMCS download manager

The MCMCS application (Fig. 2) is similar in spirit to the IVOA "VODownload" manager [7]. It permits to query an on-line registry [2, 3] using a SOAP/WSDL-based Web-service in order to retrieve the base URLs of available simple cone searches. Alternatively, it may use a (cached) table stored on disk. The MCMCS download manager passes the incoming VOTables to one or more registered "result handlers" for further processing. The default result handler stores the VOTables on disk in different directories, one per simple cone search query.

The download manager is a multi-threaded Java application, designed to minimize the latency between query start and retrieval of the last result. It uses an event-based notification mechanism to inform any registered result handler about the arrival of a new dataset.

GAVO intends to offer the MCMCS download manager as a component within its services. In addition, GAVO plans to make this tool generally available for standalone use as well as a plug-in usable by other software systems.

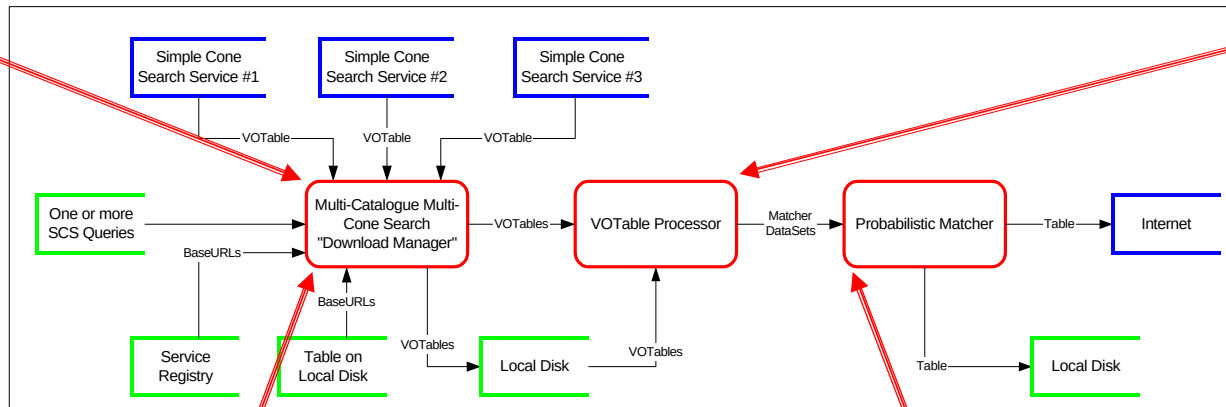

Fig. 1*: Dataflow through GAVO's multi-catalogue multi-cone (MCMC) search and matching service: an astronomer starts a query process by specifying one or more simple cone searches. A registry of available cone search services [2, 3] is used to build a table of available catalogues, from which the astronomer selects the catalogues of interest. Using this selection the MCMCS "download manager" queries the services and retrieves catalogue subsets in "VOTable" XML-format [4-6]. Each data set is pre-processed to extract the information required by the probabilistic matcher application. The latter cross-matches the entries from all datasets pertaining to the same simple query, and produces the final cross-match list.*

The VOTable processor

We are experimenting with different approaches for pre-processing the VOTables, in order to extract the data needed by the matcher:
- XSLT translation into tabular formats, e.g. comma-separated value (CSV) files, and
- XML-parsing using a JAXB parser compiled from the VOTable schema.

XSLT-processing is rather ro-bust; however it requires a reader to read in the resulting data tables. While JAXB-based VOTable parsing is elegant and the way of the future, right now the approach is hampered by the fact that many VOTables received do not validate against the VOTable XML-schema, thus causing the JAXB-parser to abort the parsing attempt with an error.

Once the VOTables have been processed, the extracted data are passed on to the probabilistic cross-matcher.

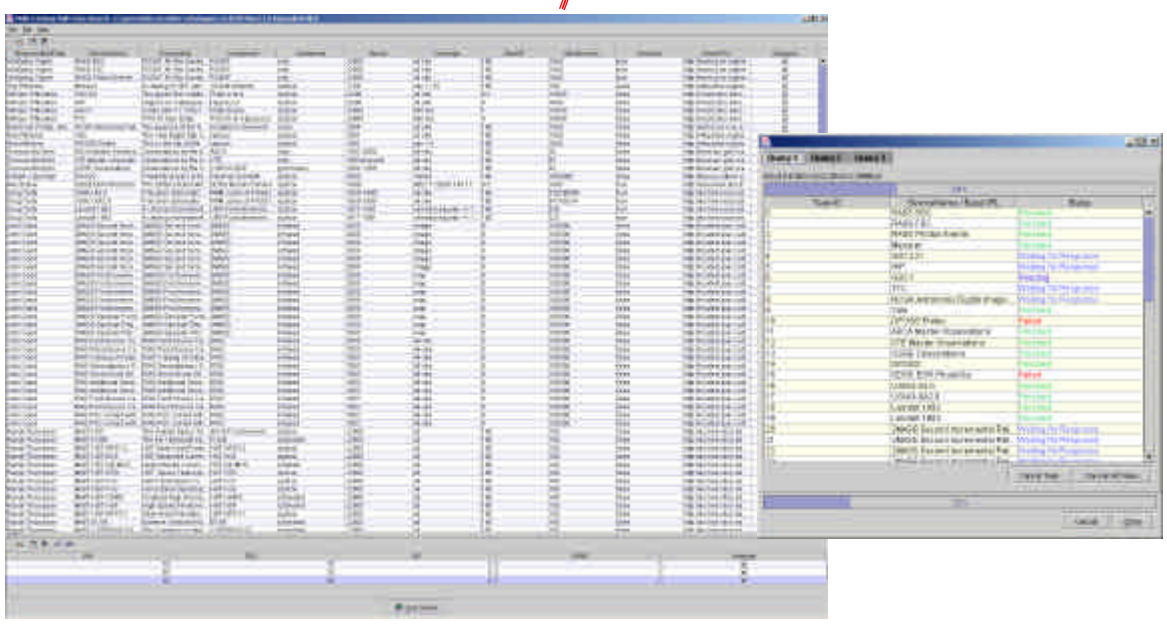

Fig. 2: *Screenshot of the multi-catalogue multi-cone search (MCMCS) download manager at work. A table (on the left) lists the available simple cone search services. The user selects the archives to be queried, and specifies one or more simple cone searches. The download manager retrieves the corresponding VOTables and passes them on to a result handler for further processing. A control panel (on the right) allows the user to monitor the progress of the multiple queries*

The probabilistic cross-matcher

GAVO's matcher is designed to perform a symmetric probabilistic match of the sources in the primary datasets from the different catalogues. Candidates are selected from each dataset, and are successively matched in a pair-wise fashion; intermediate datasets are matched with further primary datasets or with other intermediate datasets. We are using a maximum-likelihood-based approach, assuming multivariate Gaussian error distributions of the sky-positions. For each candidate match a "current-best" joint position is computed.

In essence we are pursuing similar goals as the SkyNode/SkyQuery project [8, 9]. Our matcher differs from the SkyNode/SkyQuery approach in that we attempt to use individual positional uncertainties on a per-object basis. This means it is necessary to obtain the positional errors from the catalogues.

There are different statistical measures useful for assessing the quality of a candidate match. We are exploring the use of the average squared Mahalanobis distance (see e.g. [10]) measuring the scatter of scaled distances from the sources to the best joint position. This is a generalization of the well-known chi-square statistics used in the SkyNode/SkyQuery project. Inferior matches are discriminated by applying a threshold to the average distance computed.

There are several ways *positional errors* can be specified. So far we have identified four cases:

- Type 0: no error information specified in the dataset;
- Type 1: a single error column specifying an isotropic positional error;
- Type 2: two error columns specifying two uncorrelated errors, one in the direction of the right ascension and the other in the direction of the declination;
- Type 3: a general error ellipse specified by its major and minor axis, and a position angle;

The pre-processor must be able to identify and handle these different kinds of error specifications. Internally the matcher is using a general 2D variance-covariance matrix to represent the positional error.

Observations and Issues

Overall we found most advertised SCS services operational, with a failure rate at the 5% level. However, the results returned vary syntactically and semantically to a degree that currently prevents a fully automated search and matching service. Some problems are in the data, others arise when trying to understand the schema/DTD of the VOTable itself. Here is a preliminary list of our findings:

1. Many VOTables received do not validate.
2. The service name is not unique (e.g. 2MASS-PSC is used by Vizier and Irsatest).
3. There is no standard for determining which columns are returned with which verbosity. Also, some services return errors, other return an empty VOTable, when no object was found.
4. It is difficult to automatically detect which right ascension and declination columns to use. There are VOTables that have more than one field description with a POS_EQ_RA_MAIN (or POS_EQ_DEC_MAIN) Unified Content Descriptor (UCD).
5. There is practically no way to automatically detect the type of the positional error information. Likewise, even if the type were known, it is not normally possible to automatically find which columns contain the error information, since the field descriptors are unrelated.
6. The positional error information may not be available at SCS verbosity level one (although it always returns the positional information). Thus different verbosity levels have to be tried, or one has to resort to always using verbosity = 3.
7. It is unclear whether the ID or the NAME attribute contains the "official" name of a data column. Some VOTables use both attributes.
8. The angular units are not homogeneously specified; mostly "deg" is used for the position, but we also found "degrees". The units of the positional errors are usually not "deg", but "arcsec", so a unit conversion needs to be performed somewhere in the dataflow.
9. We assume that the error in the right ascension always specifies the error on a circle in the direction of the right ascension (implicit multiplication with cos(declination)). It is unclear whether this assumption can be relied upon, or whether sometimes people might specify the error of the right ascension coordinate itself. The difference would be most notable near the poles.

Some of the issues mentioned above, e.g. the NAME or ID problem [12] have been noted before. Others are addressed in the proposed extension to the VOTable 1.0 standard [11]. E.g. column grouping is proposed in [13].

Suggestions

Here is a list of suggestions for improving the content and format of VOTables, so that a fully automated search and match process will be possible in the future:

1. Use unique service names and include them in the VOTable.
2. Replicate the SCS query in the VOTable.
3. Standardize a mechanism that allows retrieving just the field descriptions, e.g. by issuing a SCS with a negative search radius.
4. Always return the positional error information along with the positions.
5. Specify and implement a unique mechanism that allows an automatic identification of the position and error fields.
6. Support groupings of VOTable fields.
7. Indicate the type of the positional error specification (0 to 3 error columns).
8. Standardize on how angular units are specified. Perhaps, always use decimal degrees, also for the positional errors.
9. Include positional errors in the SCS service, if they are present in the original catalogue, but so far absent in the VOTables returned.
10. As a stop-gap measure, include extensive comments in the field descriptions (following Vizier's practice is to be commended) so that at least humans can find out what the fields are.

Conclusion

It is certainly an impressive accomplishment of the VO community that, with rather modest effort, it is possible to invoke a simultaneous search on 60+ services on the Internet. It is likewise impressive that the resulting datasets are available in "almost" the same data format.

In order to fully automate the search and matcher service, the VO community probably needs to spend some further work on harmonizing the deficiencies in implementing the VOTable standards, on straightening out the different interpretation of the existing standards, and on augmenting the existing standards in light of the needs of probabilistic source matching.

Acknowledgement: The core of the MCMCS download manager was implemented by Julius E. Adorf, and a pre-release was kindly made available to GAVO for use within this project.

References

1. Anonymous, *NVO compliance - Simple Cone Search*. 2002, National Virtual Observatory (NVO). p. 3. http://us-vo.org/metadata/conesearch/.
2. Anonymous, *VO Conesearch Profile Services*. 2002, NVO. http://voservices.org/conej/.
3. Anonymous, *Virtual Observatory Registry Prototype*. 2003, NVO/Johns Hopkins University. http://skyservice.pha.jhu.edu/devel/registry.
4. Ochsenbein, F., et al., *VOTable: Tabular Data for Virtual Observatory*. 2002. http://www.eso.org/gen-fac/meetings/vo2002/up/talks/ochsenbein/Ochsenbein.ppt.
5. Ochsenbein, F., *VOTable Documentation*. 2002, The VizieR Catalogue Service, Centre de Données astronomiques de Strasbourg (CDS). http://vizier.u-strasbg.fr/doc/VOTable/.
6. Williams, R., et al., *VOTable: A Proposed XML Format for Astronomical Tables*. 2002, CDS: Strasbourg. p. 28. http://cdsweb.u-strasbg.fr/doc/VOTable/VOTable-1-0.pdf.
7. Anonymous, *About the IVOA Client*. 2003, National Virtual Observatory (NVO). http://skyservice.pha.jhu.edu/develop/vo/ivoa/default.aspx.
8. Thakar, A.R., et al. *SkyQuery - A Prototype Distributed Query and Cross-Matching Web Service for the Virtual Observatory*. in *AAS 201st Meeting, January, 2003, Session 105. Mapping the Cosmos: A Variety of Surveys, Oral, Wednesday, January 8, 2003, 2:00-3:30pm, 606-607*. 2003. http://www.aas.org/publications/baas/v34n4/aas201/1137.htm.
9. Malik, T., et al., *SkyQuery - A distributed Web-based Query Service for Astronomy*. 2002, The Johns Hopkins University: Baltimore. http://www.skyquery.net/images/skyquery.doc.
10. Hsu, S.-Y., *The Mahalanobis Classifier with the Generalized Inverse* Comp. Graph. Image Process., 1979. **9**: p. 117--134.
11. Ochsenbein, F., *Proposed Extensions to VOTable 1.0*. 2003, Observatoire Astronomique de Strasbourg, France. http://www.ivoa.net/internal/IVOA/IvoaVOTable/votable-1x.html.
12. Page, C., *NAME or ID*. 2003. http://www.ivoa.net/forum/votable/0250.htm.
13. Ochsenbein, F., *Column Groups in VOTable*. 2003. http://www.ivoa.net/forum/votable/0190.htm.