USING ESO REFLEX WITH WEB SERVICES

The Sampo project is a contribution from Finland when it joined ESO. Sampo is looking into future options for ESO data reduction and has developed a graphical workflow system called ESO Reflex[1], based on Taverna[7], and primarily intended to be a flexible way of running ESO data reduction recipes along with other legacy applications and user-written tools. As a separate small side-project the team has been looking at the use of ESO Reflex with astronomical Web Services, primarily those developed within the auspices of the Virtual Observatory (VO).

ESO Reflex can use the Taverna Web Services features that are based on the Apache Axis SOAP implementation. Taverna is a general purpose web service client, and requires no programming to use such services. However, Taverna also has some restrictions: for example, Taverna has no numerical types such integers. The Taverna preferred binding style is document/literal wrapped, but most astronomical services publish the Axis default WSDL using RPC/encoded style. Despite these limitations we have created simple but very promising test VO workflow using the Sesame name resolver service at CDS Strasbourg, the Hubble SIAP server at the Multi-Mission Archive at Space Telescope (MAST) and the WESIX image cataloging and catalogue cross-referencing service at the University of Pittsburgh. ESO Reflex can also pass files and URIs via the PLASTIC protocol to visualisation tools and has its own viewer for VOTables.

We picked these three Web Services to try to set up a realistic and useful ESO Reflex workflow. They also demonstrate ESO Reflex abilities to use many kind of Web Services because each of them requires a different interface. We describe each of these services in turn and comment on how it was used.

SESAME

Sesame[2] is a name resolver which returns, from a string representing the designation of an astronomical object outside the Solar System, the position of that object in the sky and a few other details.

The original
http://cdsws.u-strasbg.fr/axis/services/Sesame?wsdl
had to be modified by adding an empty soapaction definition to our version
http://www.eso.org/sampo/reflex/instructions/workflows/Sesame.wsdl

After tuning the Taverna WSDL processor it was possible to use the Sesame service. In order to extract the appropriate fields we used XPath[3] (XML Path Language): a terse (non-XML) syntax for addressing portions of an XML document. Taverna has an XPath processor which is very convenient to parse XML responses returned from the VO.

In our example workflow Sesame is simply used to convert on object name to a sky position for the next stage.

MAST

The Multi-Mission Archive at Space Telescope (MAST)[4] uses the SIAP (Simple Image Access Protocol) and follows the REST architecture approach. It allows the use of a simple Get_web_page_from_URL processor instead of the WSDL processor. The MAST query is handled by several processors that are combined into a Taverna sub-workflow. The first is a Beanshell processor combining the query string from coordinates. BeanShell[5] is a small, free, embeddable Java source interpreter with object scripting language features, written in Java. The query returns a VOTable, which is showed to user through a VOTableviewer processor. This allows the section of files for further processing.

WESIX

WESIX (Web Enabled Source Identification and Cross-matching)[6] provides a web service interface to the standard astronomical image analysis package SExtractor.

To use the Web Service in a Java application requires a service object generated with the WSDL-2Java program of the Axis library. All changes to the service currently require new code generation and very often also updates to the client. Complex types generate lots of code which the client programmer must understand. We hope that in the future the WSDL could hold links to the public domain libraries which can be used instead of the generated code.

Our test workflow passes the URLs of the Hubble images provided by MAST to WESIX and the resultant image and catalogues are passed (through independent paths) back to a local Aladin using PLASTIC.

Fig. 1

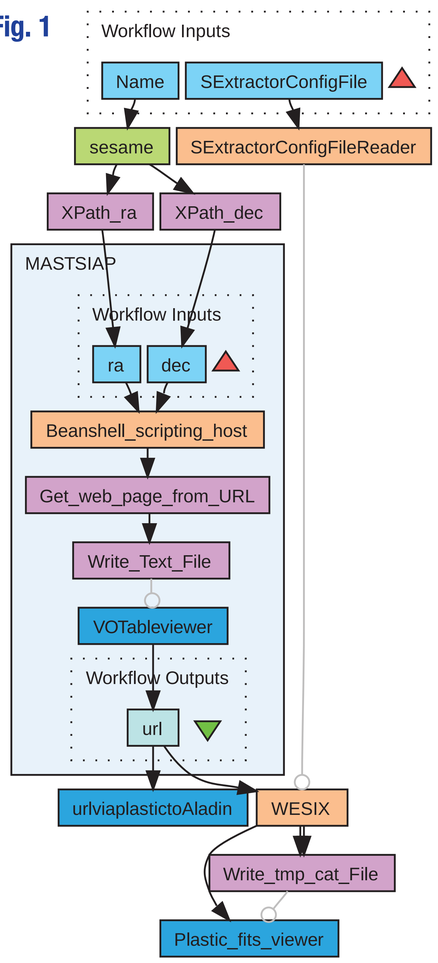
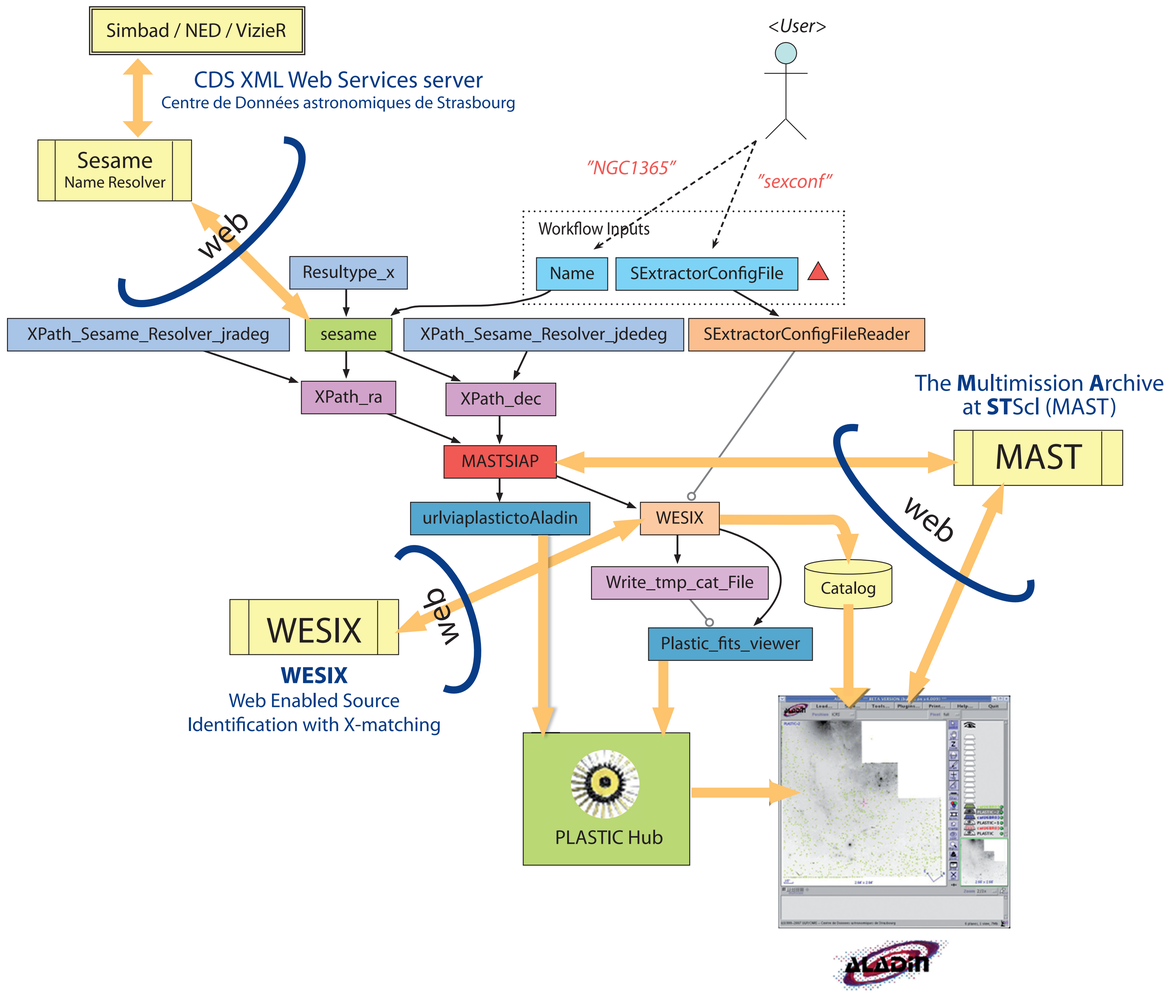


Fig. 2



The workflow is shown in Figure 1 and the relationships between the three Web Services are illustrated by the diagram in Figure 2.

CONCLUSIONS

ESO Reflex, based on Taverna, is a very effective workflow engine to utilise VO Web Services. We have prepared a simple experimental workflow using three such services. Using ESO Reflex as a more general VO client would require the addition of support for numerical types. Services using the REST architecture have proved to be easy to use.

REFERENCES

1 http://www.eso.org/sampo/reflex/
2 http://cdsweb.u-strasbg.fr/doc/sesame.htx
3 http://www.w3.org/TR/xpath20/
4 http://archive.stsci.edu/
5 http://www.beanshell.org/intro.html
6 http://nvogre.phyast.pitt.edu:8080/wesix/index.jsp
7 http://taverna.sourceforge.net

AUTHORS

Pekka Järveläinen (CSC - Scientific Computing Ltd)
Tero Oittinen (Observatory, University of Helsinki)
Richard Hook (ST-ECF / ESO)
Sami Maisala (Observatory, University of Helsinki)
Marko Ullgrén (Observatory, University of Helsinki)
Ville Savolainen (CSC - Scientific Computing Ltd)