Astronomical Data Analysis Software and Systems XII
ASP Conference Series, Vol. 295, 2003
H. E. Payne, R. I. Jedrzejewski, and R. N. Hook, eds.

Web Services and Their Use in Starlink Software

Mark Taylor, Roy Platon, Alan Chipperfield, Peter Draper, Brian McIlwrath, David Giaretta

Starlink Project, UK

Abstract. Web Services are gaining great popularity in the Grid community, and with good reason. The Starlink project is adopting Web Services as the method of interapplication communication. This is being done natively in new Java-based applications while older applications are being wrapped to provide Web Service interfaces. We are in this way providing interoperability between the generations of software in a heterogeneous, distributed manner and allowing the software to be usable in a distributed environment such as the GRID.

1. Introduction

Starlink applications have until now used a dedicated messaging system. This gives a closely coupled command interface to applications. In addition, data is accessed via the NDF data access layer. Figure 1 illustrates the overall architecture of a typical application.

As part of a new phase of developments we are starting to use Web Services as our messaging system. This not only bases our work on Open Standards, allowing us greater opportunities for interoperability, but also gives us access to a variety of tools. It allows us to work naturally in a distributed environment, and in particular positions the applications to play a natural role in the Virtual Observatory.

2. New Application Architecture

The new architecture, shown in Figure 2, must allow us to work transparently with old as well as new applications. Clients communicate with a server (for example a TOMCAT/AXIS or embedded server) via SOAP messages, usually over HTTP. The server then redirects the messages to either a JNI interface for a non-Java application, OR to a native Java application.

Data access is via the network enabled NDF data access layer for the older, non-Java applications and via the new HDX layer (Giaretta et al. 2003) for the pure-Java application. Data access could include appropriate authentication and authorisation, for example using GLOBUS-type certificates and fitting in to the Open Grid Services Architecture[1] (OGSA).

[1]`http://www.globus.org/ogsa`

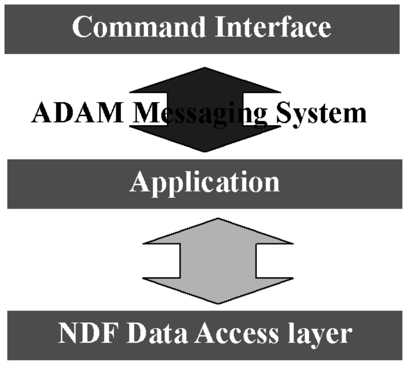

Figure 1. Starlink classic application architecture.

When defining interfaces to existing applications a number of options are available. These are not mutually exclusive, but too many interfaces can cause maintenance problems. The simplest choices include (a) complete command lines as a single string and (b) making each parameter separately available. Option (a) is simpler but less easy to use and to validate. However it may provide an easier transition for pipeline processing systems such as ORAC-DR.[2] Option (b) on the other hand is better suited to an application which is being run effectively interactively with user input.

3. Web Services Approach

Based on our investigations at the time of writing, a number of problems and advantages have been identified. Problems:

- Robustness: if the server and application are part of the same JVM then a fatal exception in the latter can kill the former.
- Interactivity: the loss of close coupling between the command interface and the application makes interactive work difficult.
- Interface definitions: any plans for replacing "classic applications" by pure Java ones will require the older interfaces to be supported, although they can be extended. Web Service interfaces to non-Java applications should therefore be chosen carefully.
- Error handling: requires special care.
- Image display: While display via X is possible, it is an extra complication.

Advantages:

- Distributable: tasks can be distributed to appropriate servers for example co-located with the data or to use available CPU power.
- Replaceable components: applications can be transparently replaced, as long as the interfaces are replaced by single or multiple new components.
- Available infrastructure software: UDDI servers to advertise information, SOAP servers.

[2] http://www.jach.hawaii.edu/JACpublic/UKIRT/software/oracdr/

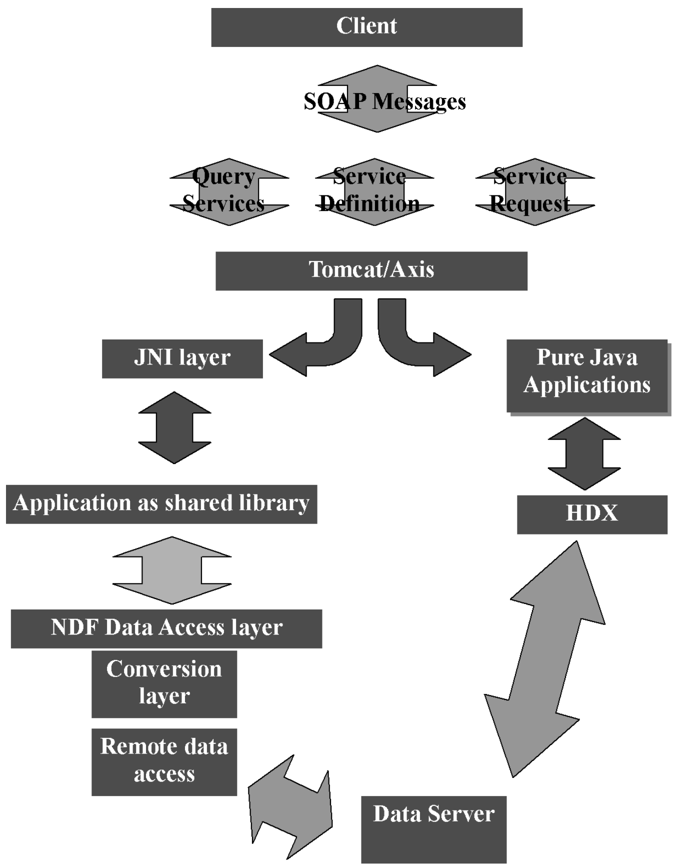

Figure 2. New application architecture.

- Compatibility with GRID: OGSA is built on Web Services. Publishing astronomical applications as Web Services is the first step in making these available via OGSA.
- Flexible command interface implementation: SOAP-enabled clients can be produced in a variety of languages using easily available support libraries, e.g., Java, Perl, Python, etc.

4. Conclusions

Use of Web Services as a new messaging system will allow Starlink applications increased flexibility. Old and new applications will be able to interoperate easily and the software will be suitable to use in the Virtual Observatory.

References

Giaretta, D., Taylor, M., Draper, P., Gray, N., & McIlwrath, B. 2003, this volume, 221