Software Architectures Expressly Designed to Promote Open Source Development: Using the Hyrax Data Server as a Case Study

James Gallagher*, Patrick West**, Nathan Potter* and Michael Johnson*
*OPeNDAP, Inc. **Rensselaer Polytechnic Institute

Abstract

Data providers are continually looking for new, faster, and more functional ways of providing data to researchers in varying scientific communities. To help achieve this, OPeNDAP has developed a modular framework that provides the ability to pick and choose existing module plug-ins, as well as develop new module plug-ins, to construct customizable data servers. The data server framework uses the Data Access Protocol as the basis of its network interface, so any client application that can read data from one of these servers. In this paper/presentation we explore three new capabilities recently developed using new plug-in modules and how the framework's architecture enables considerable economy of design and implementation for those plug-in modules. These capabilities are to return data packaged in a specific file format, regardless of the original format in which the data were stored; combining an existing data set with new metadata information without modifying the original data; and building and returning an RDF representation for data. In all cases these new features are independent of the data's native storage format, meaning that they will work both with all of the existing formats as well as modules as yet undeveloped. In addition, we discuss how this architecture has characteristics that are very desirable for a highly distributed open source project where individual developers have minimal (or no) person-to-person contact. Such a design enables a project to make the most of open source development's strengths.

Open Source Software

✦ Open Source Software (OSS) is almost always the product of a collaborative and distributed group
✦ The bulk of most OSS projects' software is written by a small group of individuals
✦ The most common means of communication used by group members is email and/or text chat
✦ Awareness of project components and interactions is often maintained by 'listening in' on other conversations
✦ Interfaces within a project can provide reference points that improve awareness and simplify communication
✦ Interfaces reduce the need to communicate high levels of detail
✦ Interfaces are an explicit and formal description of core capabilities

About Hyrax

✦ Hyrax is a data server that supports access to scientific data using the Data Access Protocol (DAP)
✦ DAP was developed by OPeNDAP with sponsorship from NASA, NOAA and NSF
✦ Hyrax also supports the THREDDS protocol for cataloging data sources
✦ Hyrax also supports common return formats like ASCII and NetCDF, XML and RDF using transports such as HTTP

Public Interfaces in Hyrax

✦ Three public interfaces define the Hyrax server's customization capabilities: BES XML commands; the module interface; and DAP itself
✦ The BES commands used between Hyrax front end and the BES define how transport protocols can use the BES's capabilities as well as how some other response can be built
✦ The BES module interface defines how new modules can use the existing modules' capabilities
✦ DAP, because it is used to represent all data, provides a common language for data sources, providing one way to connect modules together within the BES framework

Hyrax uses three public interfaces to support basic feature development and customization.

Using both the DAP and Module interfaces, the NcML handler has been developed without detailed knowledge of the modules that actually read data sets.