Geophysical Research Abstracts
Vol. 13, EGU2011-13050, 2011
EGU General Assembly 2011

The reality of deploying models in service oriented architectures

Richard Jones (1), Dan Cornford (1), Christoph Stasch (2), John Paul Gosling (3), and Jill Johnson (3)

(1) Computer Science, Aston University, Birmingham, United Kingdom (d.cornford@aston.ac.uk), (2) Institute for geoinformatics, University of Muenster, Muenster, Germany, (3) The Food and Environment Research Agency, York, United Kingdom

The increasing popularity of service-oriented and distributed computing has created opportunities for the integration of models in service workflows to create the “model web”. The model web is a concept based on the publishing, discovery and consumption of models over the web using standard interfaces. Exposing models in standard ways creates exciting opportunities for model reuse and sharing between various organisations. If these models represent granular functionality, it is possible to create complex workflows from several components. The FP7 UncertWeb project (http://www.uncertweb.org/) extends the model web concept by accounting for uncertainty in model web inputs and outputs.

Within the UncertWeb project, we aim to demonstrate the potential of the uncertainty-enabled model web by creating workflows based on several existing application domain use cases. One of these use cases, working with the Food and Environment Research Agency (FERA) in the UK, is composed of a set of models for predicting land-use and crop yield response to climatic and economic change. In this paper we will explore the challenges faced when applying model web concepts to the FERA models. In particular, this includes examples of how existing, often proprietary, data structures can be mapped to interoperable standards, such as Geography Markup Language (GML) and Observations and Measurements (O&M), in particular its XML encoding. In particular we show that it is not always simple to map all inputs / outputs into profiles of recognised XML encoding standards, and discuss some possible solutions to the problem of complex information transfer requirements.

We also discuss the practical issue of connecting existing model code to web service interfaces. It is often infeasible to modify the model implementation to support a web service interface. To ‘web enable’ a model thus requires a wrapper to written providing the web service interface and connecting to the model inputs and outputs. This approach relies on an additional layer of communication from the interface to the model. Further challenges are faced if the models have a custom interface, for example when the model has a graphical user interface and accepts inputs as files in a particular format. In this setting generic solutions will not be possible. We discuss our approach to this communication layer for both commonly used technical computing environments such as Matlab and R, and a custom model example from the FERA use case.

Finally we address issues of standards based service interoperability in model web workflows. The term ‘interoperability’ can be defined on several levels, ranging from communication within a group of services from one provider, or in a much wider domain involving several providers, indeed several application domains. When interoperability is discussed, it is often unclear what level of interoperability people are targeting.

We have found that both web service interfaces (in particular the Web Processing Service interface) and encoding standards defined by the Open Geospatial Consortium (OGC) are difficult to use, as they are attempting to target a level of interoperability which is infeasible on an implementation level. For example, the Web Processing Service interface is sufficiently general to describe any process and in an attempt to remain generic, the O&M specification states that an observation can have a result of any type. Thus if a service interface describes an input or output as an O&M observation, the user is unaware of the exact nature of the result of that observation and thus implementing a program that can use this is almost impossible. While some argue that profiles are a possible solution there is insufficient guidance or examples of such profiles in practice. Our approach is to focus on the domains of our use cases within the UncertWeb project, and create a set of profiles to restrict input and output data types. We describe the rationale and methodology for the creation of the profiles and show examples of the restrictions and how these enable us to support a wide range of applications in practice.

Finally the paper describes how we are supporting a broader community beyond the relatively small and specialised OGC adopters by embracing popular web technologies, such as JSON, SOAP, and WSDL. These technologies, often only partially supported by the OGC, have a wide array of tool and software support in the mainstream web development community. These technologies and associated tools enable rapid application development and have lead to widespread adoption across the web. We describe our approach to using JSON, SOAP, and WSDL in addition to the encoding profiles to further increase the level of interoperability in UncertWeb services.