

RIFF Submission Service

RIFF S3 and S4

Peter Raftos and Scott Yeadon, 2007

Context

There are a large number of ways in which content can be submitted to a digital repository: most repository applications come with their own submission workflows; there are standalone tools; and there are existing collections which need to be imported from older repositories.

All of these have their own ways of working and their own software. To complicate matters, each repository tool has its own internal workings and requirements. However, there is a common point for all applications: the actual object, with its associated metadata -- mainly content and preservation metadata.

The object might consist of one or many files, and it might consist of a still image, film, sound, text, tables of data. So there are lots of possible content types and file formats, as well as lots of possible metadata regimes. However, the basic model, that commonality of object-plus-metadata, remains the same.

This is where the Submission Service comes in. If it were possible to export a standardised object and metadata package from all submission applications, and ingest it with all repository applications, this would simplify the problem enormously: anyone could use a given spreadsheet or database or specialised application, and then export their collections into any repository.

The service describes a set of common, basic characteristics (using XML notation) to which the Submission Interface Packages (or SIPs) must conform. Additionally, it provides a lightweight application that can be incorporated into any workflow. The tool was written in Java and is invoked as an external service (a Java servlet), so it can be deployed on any platform.

Approach

Develop a service-oriented application providing a framework to support packaging and routing content and metadata from a source application to a target repository. In an APSR context the service should provide the framework plus default implementations of the RIFF workflow projects which map from targetted source content to Australian METS packages and subsequent routing to available repositories.

Essentially, develop both a conceptual and a technical framework to enable work around standardised Submission Interface Packages (SIPs), a la OAI-PMH. Thus, the Submission Service would provide:

1. The rules for what constitutes a SIP;
2. A framework for operating on that SIP;
3. Example applications that would perform those operations.

The SIP would be a “base model” that could handle generic needs: a file or file(s), some metadata. More specific medium or genre requirements were the responsibility of related APSR projects. These projects would extend the base model as needed.

A DRS staff member worked almost full-time on this project for the first half of 2007, with additional staff time allocated on an ad hoc basis for the rest of the year.

The application would be a web service, written in Java and XSLT. It was to be lightweight, robust and deployable on any platform. It must conform to OAI-PMH requirements. It must also be usable with DSpace, Fez/Fedora.

Outcomes

The Submission Service distribution comprises a number of default services including submission, status reporting and job management. From a software perspective the goals of the SS development were:

1. To make the Submission Service lightweight but flexible;
2. Not to be a burden for IT support staff (and other developers) to maintain;
3. To use common and stable technologies;
4. To be platform-independent.

The software was written in Java and XML(XSLT, specifically), languages which are popular, flexible and well-understood. This ensures that it can run independent of repository software. Interoperability is achieved partly through the use of the Australian METS Profile as a content format and partly through the ability to create custom tasks within a submission. Further, as the service can operate in various contexts, COSI interfaces that invoke the SS services were implemented for demonstration.

Further requirements that were met:

- deployable within the proposed National federated AAA service (e.g. Shibboleth);
- The SS makes no assumptions on the deployment context, authentication is a layer above the software and if required at a lower level, custom tasks are able to be developed.
- packaged and documented suitable for release in the public domain (conditional on the identification of a IP license suitable for the higher education sector); and
- The software is released under the Apache 2.0 license. Third party software is also included covering various licenses (details included in distribution) and is open source.

- deployable using common COSI Framework GUI components.
 - The APSR COSI demonstration deployment contains a Submission Service demonstrator that invokes SS services.

The relationship between the RIFF SS and RIFF workflow projects is quite tight. The service provides only a framework. It is only when tasks are implemented and jobs based around those tasks are configured the service can deliver any function. At the time of writing only the Journals RIFF workflow project has been completed and has supporting classes available within the service. As the partners complete their workflow projects, supporting classes will be added to the service.

As an overview for implementing a workflow using the SS, the journals workflow will be used as an example. The scope for this project was to route Open Access (OA) PDF journals published in OJS to both DSpace and Fedora repositories. In order for the service to transfer content the following was required:

- Development of an OJS Gateway Plugin which exposes the existing OJS markup as a web service
- Development of the Australian METS Profile and an Australian METS Journal Profile
- Development of various tasks within the service including transforms and transfers
- Development of DSpace hooks for submission service and Fedora transforms

Using the service with the journal implementation allows the configuring of periodic (batch) imports of journals and/or journal issues from OJS to DSpace or Fedora. It is also possible to set up a form-based submit for journals to DSpace/Fedora for ad-hoc submissions.

The remaining workflow projects will require similar treatment in that new tasks will have to be developed within the submission service to satisfy the requirements of each workflow.

Further work

As mentioned, the submission service does not, on its own, do anything. Further implementation work for specific submission tasks is needed.