

Interactive Multi-Submission Deposit Workflows for Desktop Applications

David Tarrant¹, Les Carr¹, Alex Wade², and Simeon Warner³

¹ School of Electronics and Computer Science, University of Southampton
davetaz, lac @ecs.soton.ac.uk

² Scholarly Communications, Microsoft Research
Alex.Wade@microsoft.com

³ arXiv.org, Cornell University
simeon@cs.cornell.edu

Abstract. Online submission and publishing is the norm for academic researchers. With the pressure on these authors to submit their work to conferences, journals and Institutional Repositories, this leads to demands on the author to go through multiple web based interfaces, filling in forms with the same information multiple times before they can submit. At the same time, each of these services in turn will have made policy decisions on what types of format they allow and what templates the content has to conform to. The amount of work expected of the author does not add up to the potential gain, thus most authors will only submit into the repository or publication where they foresee the most benefit. In this paper we propose a solution to this problem that embeds the workflow for multiple submissions into the desktop application of the author, most commonly Microsoft Word. We also propose extending the work done on the Microsoft Word Author Add-in tool to allow two-way negotiation between each repository and the desktop application.

SWORD (1) has been a significant addition to repository functionality, allowing program-controlled deposit rather than requiring author effort for completion of multiple forms via a Web browser. It has opened up direct submission from the desktop to the repository (2). With most major repositories now adopting this standard, the effort required for submission through complex web based workflows has been significantly reduced. However this had the unfortunate side effect of loading a much greater amount of work onto the repository editors who have to maintain the high quality of metadata within a repository to which a significant amount of design effort has been applied. Each repository has to consider its dissemination mechanisms and authenticity of documents which it provides. This leads to the majority of repositories demanding a PDF version, which demands extra time and effort on the authors part to generate and upload. The inverse situation is also true where a user doesn't realise they can submit an editable copy (such as OpenXML, Open Document Format or TeX etc) to the repository alongside the PDF for preservation purposes.

From the point of view of the repository editor, the biggest problem is obtaining full and correct metadata from the author. Currently SWORD is a one-way protocol, meaning that a repository can either accept a record, or reject it; there is no middle ground. Adding a lightweight mechanism to desktop applications to enable negotiation on what is sent in a SWORD package would go some way to bridging this gap.

On the desktop a user may be using a contacts management system from which their authors are included and a reference manager which handles all the citations and citation style for a document. At the point the document is submitted all this valuable information (such as author identities disambiguated by email address and structured citation listing) is lost.

Our proposal is to enable a simple yet powerful set of negotiations to occur between the desktop application and multiple repositories such that a single familiar submission workflow (in the style of the author's application) can be presented to the user. At the same time the application and repository can negotiate the packaging format and items which should be included in the package. Figure 1 shows an example interaction between two repositories, the desktop application (Microsoft Word) and the user. In this example our institutional repository has detected that there are no subjects defined in the metadata, thus it is requesting these alongside a PDF version of the submission. Our National Archive (as depicted in figure 1) is providing a more advanced service where it takes a full text copy of the document, processes it for keywords and then feeds these back to the user as recommendations as well as asking for any others.

Figure 1 also demonstrates the asynchronous nature of the system. Once the initial communication with the repositories is complete, the desktop application can be requesting additional metadata from the user whilst also generating any alternate versions of the document the repository has requested. Finally the existing SWORD implementation is used to submit all this data to the repositories and present the user with their receipts.

As part of this work we would also like to advocate that repositories advertise their endpoints (SWORD included) via their front page as meta-markup. The Registry of Open Access Repositories (ROAR)⁴ has already been extended to support this. This enables the author to submit to a repository by copying the front page URL into their desktop application. This meta-markup would also point to the processor used to initiate and control the transaction between the desktop application and the repository. We envisage that all of this process would be performed via a RESTful interface which pushes back and forth XML/RDF data conforming to a lightweight specification.

⁴ ROAR - <http://roar.eprints.org>

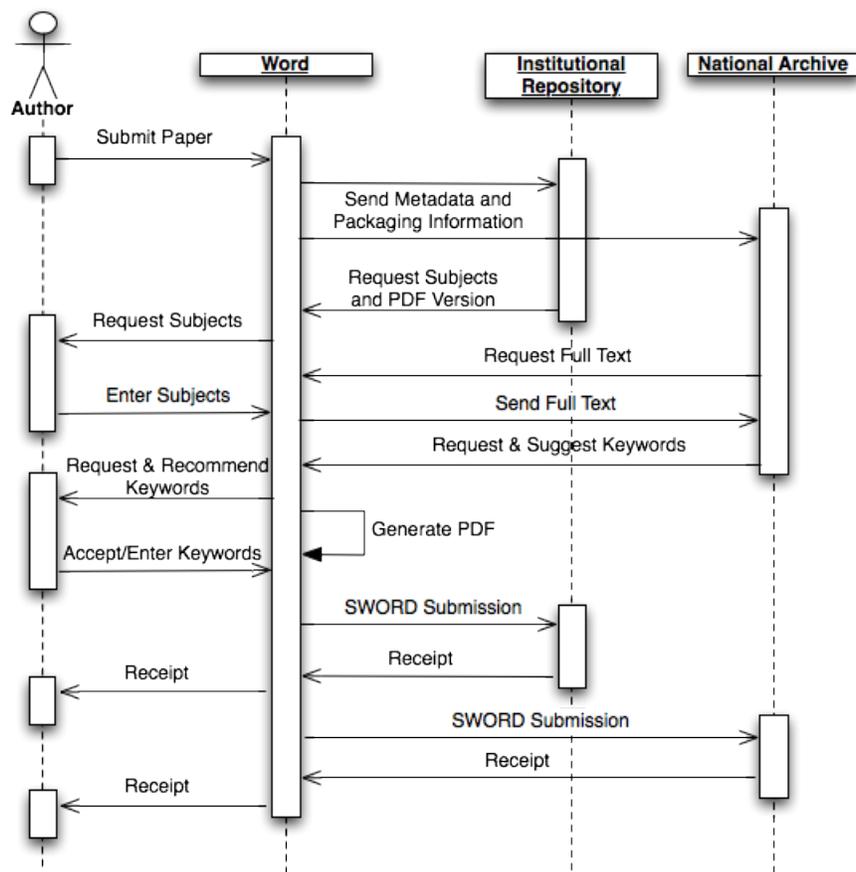


Fig. 1. Sample sequence diagram for communication between Author and Multiple Repositories

With this work in its early stages we are open to discussion with the community on this work which we believe to be an enabler of not only single click submission of an author's work into multiple repositories, but also of a much higher quality of metadata. The JISC Citation Services project is funding the first step of this architecture; the ability to package up and deposit explicit structured citation information for storage as harvestable metadata in the repository. A demonstration of this work will be available at the Open Repositories conference. Overall, providing a workflow to the author via a tool which is already familiar to them will increase the size of a repository's self-depositing community and opens up the opportunity to obtain research data from a wider suite of desktop applications.

Bibliography

- [1] Allinson, J., François, S., Lewis, S.: Sword: Simple web-service offering repository deposit. *Ariadne* **54** (2008)
- [2] Fericola, P.: Connecting authors and repositories through sword. In *Proceeding of Open Repositories 2009*, Atlanta, GA (2009)