## Conference or Workshop Item:

Allinson, Julie (2008) SWORD : simple web service offering repository deposit. In: Third International Conference on Open Repositories 2008, 01-04 Apr 2008.

# SWORD : simple web service offering repository deposit

## A proposal for Open Repositories 2008

## Julie Allinson

## Abstract

This paper presents an overview of a JISC (Joint Information Systems Committee) activity to scope, define and develop a deposit specification for use across the repositories space, which has come to fruition within the SWORD (Simple Web service Offering Repository Deposit) project [1].  It will look both at the background and how this piece of work came to pass, the movement from informal working group to funded project, the lightweight project construction and the resulting protocol and technical outputs.  The paper will also consider the future of SWORD and look at some of the activity which has already galvanised around the project outputs.

## Before SWORD there was the Deposit API

Discussion around the lack of a standard mechanism for deposit began at the JISC CETIS Conference in 2005 in a session led by Lorna Campbell and Phil Barker.  At around the same time Andy Powell, in his 'A service-oriented view of the JISC Information Environment', identified a set of discreet services that a repository should offer, among them 'deposit'.  Falling out of these discussions and aligning with the newly formed Repositories Research Team, a collaboration between UKOLN and CETIS, led by Rachel Heery, a group of repository developers were contacted and asked firstly whether they agreed that there was an identifiable need for a standard deposit protocol and secondly whether they would be willing to join a small working group of repository developers to kick-off discussion in this area.

Among those who responded were Thomas Place (ARNO), Leah Houser (OCLC), Ben Ryan (KaiNao), Jon Bell (Harvestroad), Martin Morrey (Intrallect), Chris Gutteridge and Tim Miles-Board (EPrints, University of Southampton), Richard Green (Fedora, University of Hull), Jim Downing (DSpace, University of Cambridge) and Herbert van de Sompel (OAI-PMH and aDORE, Los Alamos National Laboratory).  All but the last of these were able to attend a one day meeting in March 2006.

Largely a scoping exercise, this initial meeting was followed in July with a two-day event focussed on firming up on the requirements for a deposit service and working on possible serialisations.  From this the beginnings of an implementable serialisation was forming. But, as is often the case with distributed activities that demand the gratis input of busy individuals, the work began to slow down.  Despite that, interest remained high and in January 2007 the work to date was presented to the Open Repositories 2007

---

[1] http://www.ukoln.ac.uk/repositories/digirep/index/SWORD/

conference and generated further international interest. One concern expressed at OR2007, though, was that this activity was too parochial, focussing on the UK-only and on developing a new standard or specification where others already existed, and could be leveraged. Throughout, international collaboration and the potential for re-using existing standards had been very much within scope for the deposit API exercise but since few repositories had made headway in this area, extra effort was needed to review the deposit specification potentials.

## JISC to the rescue …

Under the Repositories and Preservation Programme in March 2007, JISC funded a six-month project to take the deposit API activity into a more formally funded project. Led by UKOLN, the project was a partnership between CASIS at the University of Aberystwyth, the University of Southampton and Intrallect. The project aims were simple – to agree on a protocol or specification for deposit, to implement a deposit interface into DSpace, Fedora, EPrints and IntraLibrary and to produce a prototype 'smart' deposit client for testing the implementations.

## But why? … slicing up the scenarios

There are a number of reasons why a standard deposit mechanism would be useful. Various manifestations of multiple deposit, such as deposit into institutional and subject repository, mandated deposit into a funder repository, transfer from repository to repository and mediated deposit can all be supported by SWORD.



SWORD scenario illustrated
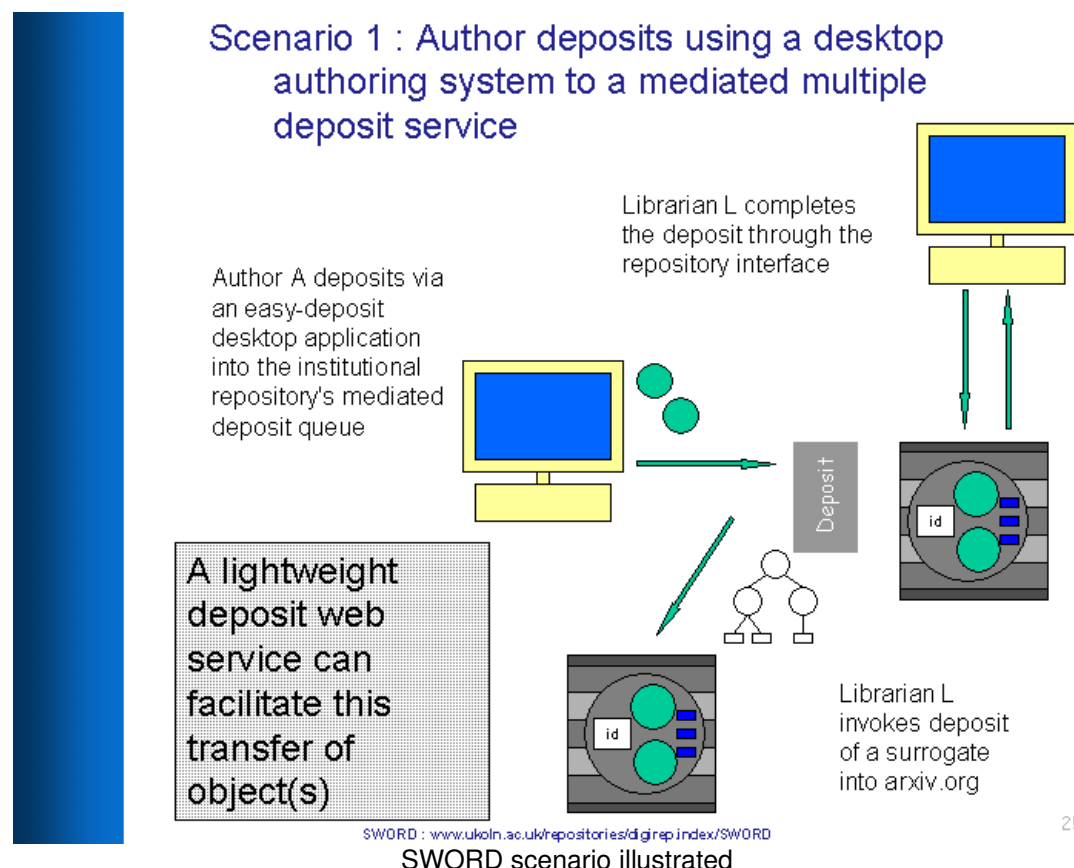
## Making scenarios happen

Once we knew just what it was we were trying to do in our scenarios, the next obvious step was to boil these down into a list of clear requirements or parameters.  A shortlist of candidate deposit specifications was drawn up, including WebDav, the OKI OSID and SRU/W.  It was the Atom Publishing Protocol (APP or ATOMPUB) that captured the interest of the group.  Lightweight, relatively easy to implement, already growing in use for remote blog posts and closely tied to the Atom Syndication format (ATOM), widely used across the Web, APP had much in its favour.  Would this lightweight standard be sufficient to fulfil the needs of repositories?

## Talking technical

Within the boundaries of the SWORD project we have produced demo SWORD deposit interfaces in EPrints, Fedora, DSpace and IntraLibrary.  Prototype clients in desktop, command-line and web form allow for testing of SWORD interfaces, and sourceforge code for all of these enable re-use and wider dissemination.  Case studies about on-the-ground implementations will help us to understand where SWORD does and doesn't fulfil its primary goals.

## The future … where seamless deposit actually happens

SWORD has achieved what it set out to do.  It facilitates deposit into repositories from remote locations in a standard way.  Now is the time to encourage uptake and implementation.  This is perhaps the most difficult part, but beginning from the foundations outlined above, with the backing and input of repository developers we hope that SWORD really can help interoperability of deposit become a reality.  This is already happening with a variety of implementations.  Not only that, but the process that has led us to where we are now can hopefully help future development activity funded by JISC.  Having presented this work at Open Repositories 2007, it seems only fitting that we return to Open Repositories to report on its progress and potential.