RePub, the IR of the EUR: "It's alive"[1]

Peter van Huisstede & Jasper Op de Coul

October 1, 2012

[1] Presentation at Emtacl12, Trondheim, Norway, 1–10–2012. RePub website: http://repub.eur.nl Contact: Jasper e–mail: opdecoul@ubib.eur.nl; Peter e–mail: vanhuisstede@ubib.eur.nl

RePub is the institutional repository of the EUR. It is part of the University Library. It is a cornerstone of the EUR's Open Access policy. Within the University Library RePub functions as an innovation hub for new library services. We develop the software to run the repository and build services upon it.

Developing our software we use a model of parallel tracks. Both the research workflows and the university library workflows are – turning more and more into– digital workflows that process data into information. In between the two parallel tracks we place projects as "sleepers".

So, right from the start, RePub has been very much alive; not just a box one drops PDF files into, closes the lid and that is it, but developing software in close contact with researchers and based upon ideas about library innovation.

In the following we will present three "sleepers", projects that join the two university digital workflows: research and university library.

Introduction: Parallel tracks

RePub is the institutional repository of the EUR; it is part of the University Library. It is a cornerstone of the EUR's Open Access policy. Within the University Library RePub functions as an innovation hub for new library services.

RePub started with custom software called "webdoc". Then switched to DSpace, only to find out that a RDBMS containing a lot of text–strings is perhaps not the best datamodel for an institutional repository.

We are currently working on the 4th iteration of our repository software. The first iteration, RePub 1.0, was modelled after the way programmers would work with their program files: Always run under version control, it is all about the programs, good clean metadata. So RePub 1.0 was build on top of Subversion, a backend with XML files and assets, workflows and a web front–end.

We kept developing repository sofware. RePub 2.0 never came off the drawing board, although we learned a lot. RePub 3.0 is the current production system (DVCS (Mercurial), RDF & JSON, editor, web front–end) we will show parts of today. RePub 4.0 will essentially be RePub 3.0 on the Google App Engine (GAE).

Developing our software we use the model of parallel tracks. Both the research workflows and the university library workflows are digital workflows that turn data into information. In between the two

Figure 1: Parallel tracks: Research and library research workflows are digital and aim to process data into information

Figure 2: Wrong track: Dead–end

parallel tracks we place projects as "sleepers".

So, from the start, RePub has been very much alive. Not just a box to drop PDF files into after which the lid is closed.

The editor

In order to be of any use, a repository should contain "all" the (scientific) publications by EUR staff. So, we are not going to wait until researcher such and such decides to put something in, we getting in all stuff that is associated with at least one EUR author. Our staff will concentrate on reviewing the data; authors fix the bloopers: "No that is not me, that is my nephew".

Figure 3: First sleeper: Getting things in

We use RDF (ID's, types, SKOS concepts, values and relations) to keep track of things.

Using the PDF itself or, preferably, the DOI, we fill the repository.

Once the data is in, we process the data in various ways to produce useful information.

Presentations & Visualizations

We produce authorpages with publication lists. We pull in citation scores from other parties, we produce bibliographies, in LaTeX for example.

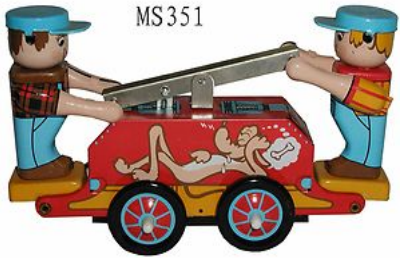

Figure 4: Second sleeper: Visualizations

Authorgraphs are quite popular: Who is related with whom through authorship of publications.

Another kind of information we present are the animated graphs on the journals articles EUR authors publish their work in.

Future developments: Where do we go from here?

One issue we are contemplating for some time now is how to distribute our software (assuming that somewhere out there wants to use it). Use the model of open source? Or set it up as a Software as a service (SAAS) in th so–called cloud?

Figure 5: Future developments

At the moment we are moving in the direction of the latter. Our Apache log analysis using the infrastructure of Google's BigQuery. Since Google Analytics does not provide us with much information about the actual use of Open Access full-text files, we use our Apache server logs, add a lot of repository metadata and sent the lot to Google's BigQuery.

An environment for contextual reading, hooking up the results of full–text analysis with outside sources like DBPedia, STW–thesaurus, etc., etc.