

Making a Home for a Family of Online Journals

Robert Forkel

Heinz Nixdorf Centre for Information Management, Max Planck Society
Boltzmannstr. 2, D-85748 Garching, Germany
email: forkel@zim.mpg.de

Abstract

In this paper we present the e-publishing platform created by and for the Living Reviews project [1]. In particular, we give an overview of the Living Reviews project, concept and publishing model, compare our project to related projects, and present the technical infrastructure used to run the "Living Reviews" family of online journals. We will give an overview of the publishing environment, point out the challenges inherent in distributed editorial offices, and describe the interaction between various user groups and the infrastructure. Based on this description of the environment, we will explain the constituents of the infrastructure and how they evolved. Following the history and detailed description of the constituents we give an overview of further plans within our project and applicability of our particular solutions to other publishing projects. Again on this level of detail, we will compare our solutions to related efforts. We will conclude the paper listing lessons learned from our project. In particular, we will describe the challenge of reconciling the technical and the non-technical divisions of the publishing project and how it was met for the Living Reviews family. While mainly providing a case study, we also hope to point out solutions to publishing problems which may be of general interest, and more importantly make a case for necessary standardization of e-publishing software components. We don't see much scientific value in this venture, but value for the scientific community since we present another e-publishing platform to use, provide another set of tools to choose from, give another best-practices description when rolling your own. For the organisational and interactional aspects see [2].

Keywords: e-publishing software; e-publishing platform; review journal; e-publishing services; open source

1 Introduction

The Living Reviews project started out as a venture to establish an online journal with a particular concept: Publish solicited review articles and keep them up-to-date. Slowly this idea was spreading. By now the Living Reviews publishing platform is used to run – i.e. create, maintain, host – three scientific online journals, with a fourth one in preparation. The term platform can be understood in a wide sense. So it comprises:

- the concept (see "What is Published?"),
- the project (i.e. - basically - the people, see "Who are the family members?"),
- the technical platform (see "The Technical Infrastructure").

1.1 What is Published?

Living Reviews journals publish *solicited*, peer-refereed *review* articles that are being *updated* by their authors. This means:

- The invitation process to find authors is a substantial part of the editorial back office work.
- Articles are big - 100+ print pages, 300+ references.
- Publication rate on the other hand is rather low (for Living Reviews in Relativity around 10 per year).
- Published articles must be amenable for authors to update.

All these issues set us quite apart from typical electronic publishing ventures. Providing a niche product as ours does also help establishing a good standing. But obviously, relying on solicited articles means the editorial board must be of a very high scientific authority.

1.2 Who are the Family Members?

The family members in the narrow sense are the journals published under the Living Reviews umbrella. But there's more structure and diversity to the people involved. Project management and technical development for

the family is located in Garching: 1 project manager; 1,5 programmers. The back offices for the journals are established locally:

- For LR in Solar Physics and LR in Relativity a shared back office with a staff of 2 is located in Golm at the MPI for Gravitational Physics.
- For LR in European Governance, a one-person back office is situated in Wien.
- For LR in Landscape Research it will be established in Muencheberg.

Editorial Boards are international for all journals. Again, since our publications are solicited, we better treat our authors as family members as well.

1.3 Any Legacy?

From the history of the project one could see all but LR in Relativity as second generation Living Reviews journals. And of course there was quite a legacy. But starting a second journal was about the best that could happen because it forced rethinking of practices and processes, redesigning software and a re-evaluation of the Living Reviews concept. The result of coping with the legacy is the current Living Reviews publishing platform; and since it was born from need to evolve, we hope it will support further evolution well.

2 Related Projects

While above it was said that quite a few characteristics of Living Reviews set us apart from other publishing ventures, in this section we will look at similarities with such.

2.1 GAP – The German Academic Publishers

The GAP project [3] is one we have been in contact with for a long time. While the general aim is comparable to ours – namely to further open access to peer refereed scholarly works – the concrete activities are more geared towards the needs of publishing houses. GAPworks, the online publication system developed by GAP, is comparable to the EIMS component of our epublishing toolkit. Thus, standardization in electronic publishing – another goal of GAP – may lead to interoperability/interchangeability of infrastructure components developed by GAP or us.

2.2 DiPP – Digital Peer Publishing

DiPP [4] is another – rather young – German initiative which shares our goals pretty closely. This initiative vastly encourages the foundation and expansion of scientific eJournals. But again, their focus is more on journals publishing original research articles. The scope of services – in terms of infrastructure and concept – is pretty much the same as for our project – i.e. DiPP also provides a complete publishing platform and offers infrastructure for the publication of eJournals. While the organization and publication of a journal are at the journal editorships, DiPP takes over central and spanning tasks to hbz:

- hosting of journal content
- technical support and consulting of journal editorships
- guarantee of a safe and long-term archiving as well as a durable citation
- drafting templates and style sheets for publication and presentation of articles
- configuration, advancement and optimization of the technical basis
- statistical evaluations
- distribution and indexing (common), that is reference of journals and articles in search engines and databases
- sharing metadata by OAI-PMH2.0 compliant interface
- right management, license questions in co-operation with ifrOSS
- co-ordination, contracts, co-operation with third
- development of operating and business models
- representation of DiPP; presentation on relevant meetings and congresses

In the publication initiative DiPP are offered modular service packages cut to the individual case, which cover the range from simple basis to a comprehensive services. A particularly interesting part of the DiPP activities is their work towards implementing a publishing platform on top of the repository software Fedora.

2.3 DPUBS – Digital Publishing Systems

Cornell's Digital Publishing System project [5] started out with a particular publishing project (project Euclid), expanding to more journals, and redefining the development project in the process. So they do share some characteristics with the Living Reviews project. DPUBS is mainly a software project; their aim is not only to provide components for publishing platforms in form of web services, but also to standardize the interfaces between these services. This effort may well be the most advanced in terms of clean interface design between publishing components: "The DPubS architecture supports and coordinates distinct services, which are implemented as separate software modules. [...] For example, the Index Service indexes metadata or full-text from a repository, or set of repositories, queries these indexes when requested, and returns search results. The Subscription Service manages subscription data and answers access rights questions when controlled content is requested. Other services include Repository, User Interface, and Registry Services. Each DPubS service has a well-defined interface. The syntax for making every request of a particular service and the expected format of each response are documented. The Dienst protocol formed the basis of this documentation, and subsequent extensions (new requests and responses) are documented in the DPubS code. This protocol is based on HTTP verb requests. A similar idea was employed for the OAI protocol for metadata harvesting, which borrowed directly from Dienst and resembles it closely in design. The advantage of having such clearly articulated interfaces is that services can be extended (the addition of new functionality, with corresponding new requests and responses), without breaking existing dependencies. This flexible and extensible modular design makes the DPubS system well-suited for open source development, since existing functionality need never be disrupted by another developer's desire to enhance that functionality."

2.4 ESPERE

The ESPERE project [6] had an influence on Living Reviews in terms of the publishing model. While at first, our software was thought to be the good to share with other journals, it turned out – with ESPERE as a precedent – that hosted solutions are much more attractive.

2.5 Other Online Journal Systems

Various projects develop complete online journal systems (for example the Open Journal System <http://pkp.sfu.ca/ojs/> and HyperJournal <http://hjournal.org/>). Most of them address the same kind of journal: journals publishing non-solicited, one-file, original research articles.

2.6 Summary

Learning from other projects is definitely sensible. Reusing software on the other hand is hard, in particular with our special set of requirements. Still, the market of publishing platforms is not so crammed yet, to make creating a new platform a vain attempt. But it also seems to be the right time, to think about standards. OAI-PMH to disseminate metadata is one example. An RDF vocabulary for journal metadata might be useful as well, to foster interoperability of epublishing software (in this case, metadata storage backends and dissemination or web front ends). Another interface which might be worth standardizing is the one between publishing tools and content repositories. DiPP and DPUBS both target Fedora (<http://fedora.info/>) as repository backend. This goal is also very explicit in the DPUBS project (see <http://dpubs.org/development.html>), since they aim to develop software for epublishing as collection of fine grained web services, which can be accessed/integrated via REST/SOAP interfaces.

3 The Living Reviews Publishing Platform

The Living Reviews publishing platform – as described above – is a multi-faceted thing. Different people see different aspects of the Living Reviews platform:

- The prospective journal founder sees the concept (<http://www.livingreviews.org/faq.html>).
- The netizen sees the web content (i.e. the outside of the journal).
- The editorial back office sees additional applications: A subversion/trac instance for manuscript handling, the editorial information management system and various ePubTk applications running centrally or decentrally.
- The developer sees the epubtk software project [7].

While this section may look like a hodgepodge of technologies, tools, best practices, it should still look familiar to anyone responsible for the technical aspects of an online publishing venture.

3.1 The Production Environment

The production environment for Living Reviews runs on two servers located in the RZG (Rechenzentrum Garching). These machines run apache web servers, webkit application servers, postgresql database servers, and several additional services. Important tools used throughout our project are the version control system *subversion* [8] and the lightware collaboration software *trac* [9]. The web content is kept in subversion repositories (and served from local checkouts of these). This mechanism has several advantages (as compared to – for example – FTP):

- Secure (HTTPS), reliable (transactional) upload mechanism for web content produced in distributed back offices.
- Versioned web content.

Subversion is also used to:

- store sources (i.e. the manuscript and additional files) of each publication,
- allow concurrent editing of sources.

On top of these repositories we use trac to:

- browse the subversion repository (see revision log and diff screenshot),
- track the editing process via tickets and milestones (see roadmap and tickets screenshot),
- document editing know-how in the wiki.

logged in as robert

Logout

Settings

Help/Guide

About Trac

	Wiki	Timeline	Roadmap	Browse Source	View Tickets	New Ticket	Search
--	------	----------	---------	---------------	--------------	------------	--------

View Latest Revision

root / 2005_armstrong / article.tex /

View revision: 590

Date	Rev	Chgset	Author	Log Message
01/23/06 13:32:07	590	590	fschulz	as published
01/19/06 18:42:49	589	589	harrydee	Last changes after author feedback. ...
01/04/06 10:46:34	558	558	harrydee	Article is now in LivRev? style; some feedback from author needed. ...
11/29/05 12:05:00	488	488	fschulz	final version as submitted Nov 29
09/26/05 12:13:36	361	361	fschulz	changes for new referee version with pubtool
08/18/05 16:00:39	163	163	fschulz	harrys changes to create referee version, committed by christina on turtle
08/09/05 12:51:08	65	65	christina	setting more svn:keywords
08/09/05 12:49:20	64	64	christina	setting svn:keywords property
08/08/05 19:04:12	63	63	fschulz	removed clearpage
08/08/05 17:12:57	62	62	fschulz	edited figure positioning and footnote
08/08/05 17:05:09	61	61	fschulz	first edited version that runs through epubtk
08/08/05 15:12:23	60	60	fschulz	moved submitted files to edited
08/08/05 12:10:24	39	39	fschulz	sources as submitted by author Armstrong

Figure 1: Revision history of a manuscript viewed with *trac*

Ideally, also the authors would use a checkout of their article from our subversion repository while revising it after submission. At least in the physics community this might even happen. As an additional service, a weekly analysis of the web server logfiles from the production machine is run and the reports provided online (see <http://www.livingreviews.org/journalUsage.html>)

3.2 The Test Environment

A Test server at the Albert Einstein Institute (where the back office for two of our journals is located) serves the nice-to-have functionality, like:

- more realistic testing of web content (than pointing your browsers at files on your hard-disk);
- server to provide publication previews for authors.

3.3 ePubTk – The epubublishing Toolkit

During the lifetime of the project, quite a bit of software has been developed, automating many aspects of publishing. This software is available as python package, comprising components as sub packages which we will describe in the sequel.

The *pubBuilder* component is the oldest part of the ePubTk software. It is used to create HTML and PDF presentation versions from manuscript sources in LaTeX and BibTeX. For an example of a publication created with *pubBuilder* (using the support for updates) see <http://relativity.livingreviews.org/Articles/lrr-2003-2/>

This component can be used on its own and is actually a good fit if:

- the document input is LaTeX+BibTeX anyway;
- high-quality HTML *and* PDF need to be generated from the same source;
- publications should support some sort of 'update' functionality.

The *jrnBuilder* component provides functionality to create various journal pages using the standard ePubTk page creation mechanism, i.e. XSLT, thereby providing access to all the journal's configuration data. This component may be used separately. alas, the main reason for its existence is the integration of the configuration data we already need for other components in the creation of the rest of the journal.

The *register* component provides the functionality to manage public metadata of publications. This includes a GUI application to edit this metadata (see Figure 3), an RDF triple store and functionality to create views of the metadata via XSLT. This component can be used separately and may be extended easily, since the triple store is semantic agnostic (with exception of update information) and the GUI is created from a pretty simple description of the metadata. The Living Reviews journals use *register* to create publication indexes, author pages, RSS feeds, OAI-PMH static repository data, etc.

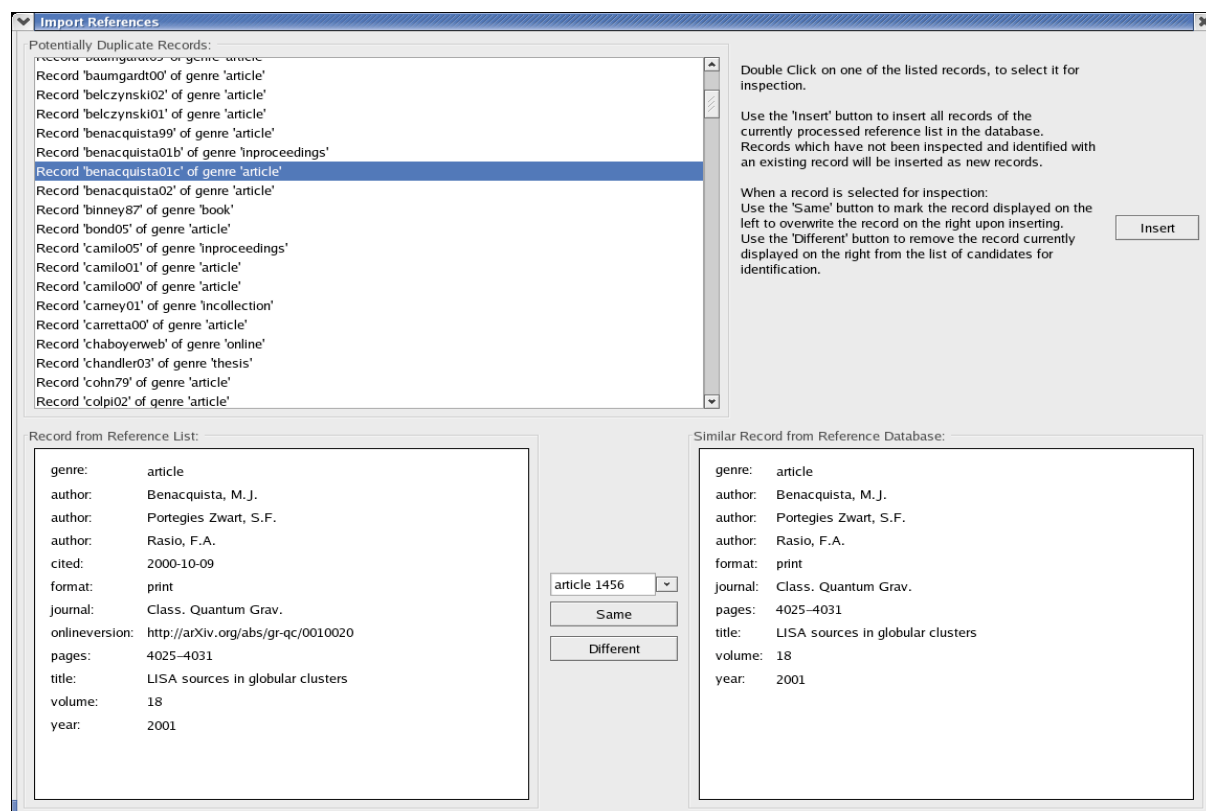


Figure 2: Import application for the reference database

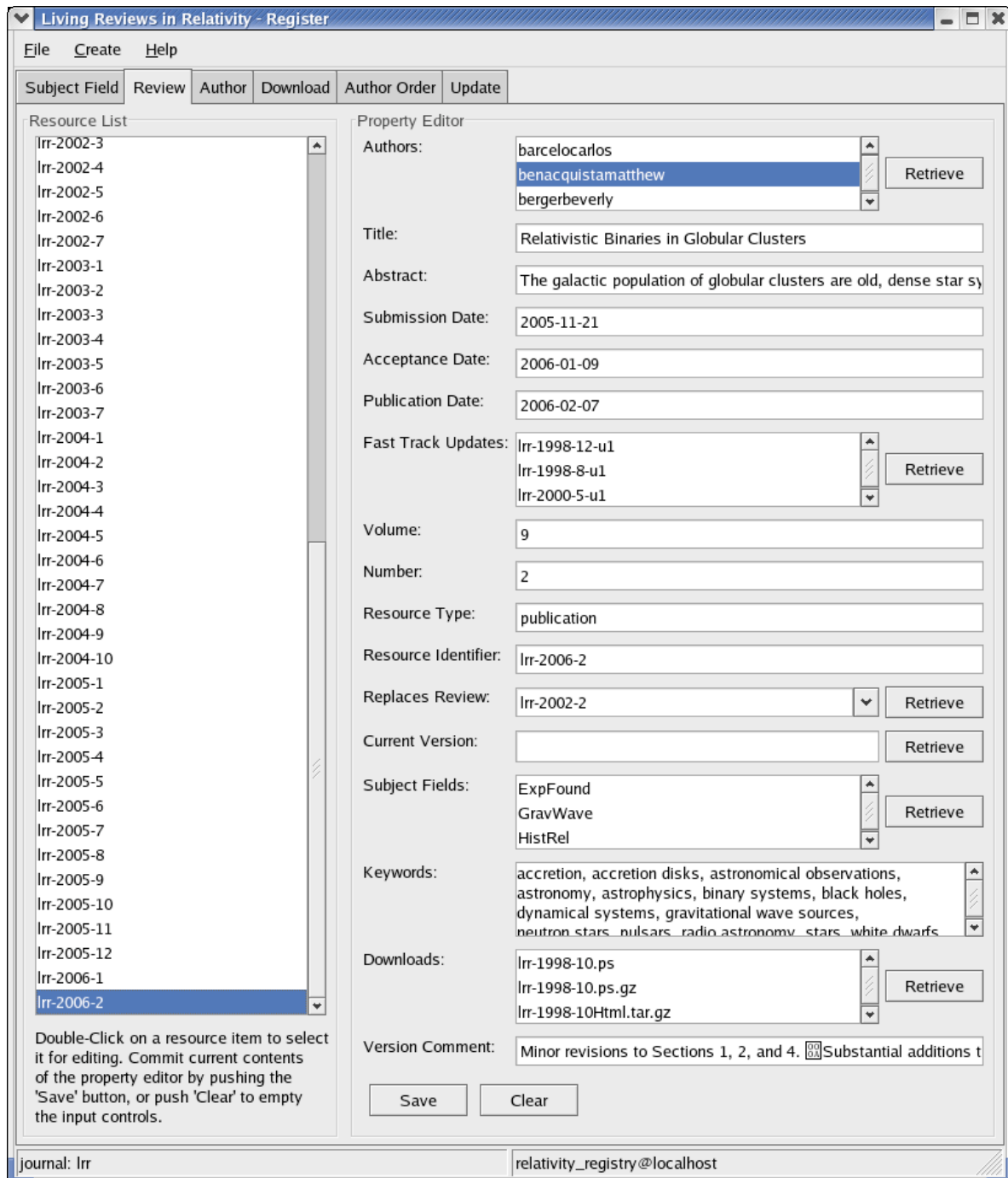


Figure 3: The register editor application

Since references are a very important part of review articles, we keep them in a database. Functionality to manage this database is provided in the *refdb* component. The import format of this database is a custom XML-based format described in an XML schema (link to schema and documentation!). The functionality of *refdb* comprises:

- a BibTeX import filter application (see Fig. 4),
- a general import/export application (see Fig. 2),
- functionality to detect and merge duplicate entries,
- support for publication updates (new references or no longer cited references are detected),
- a web application to search the database (e.g. <http://relativity.livingreviews.org/refdb/search>).

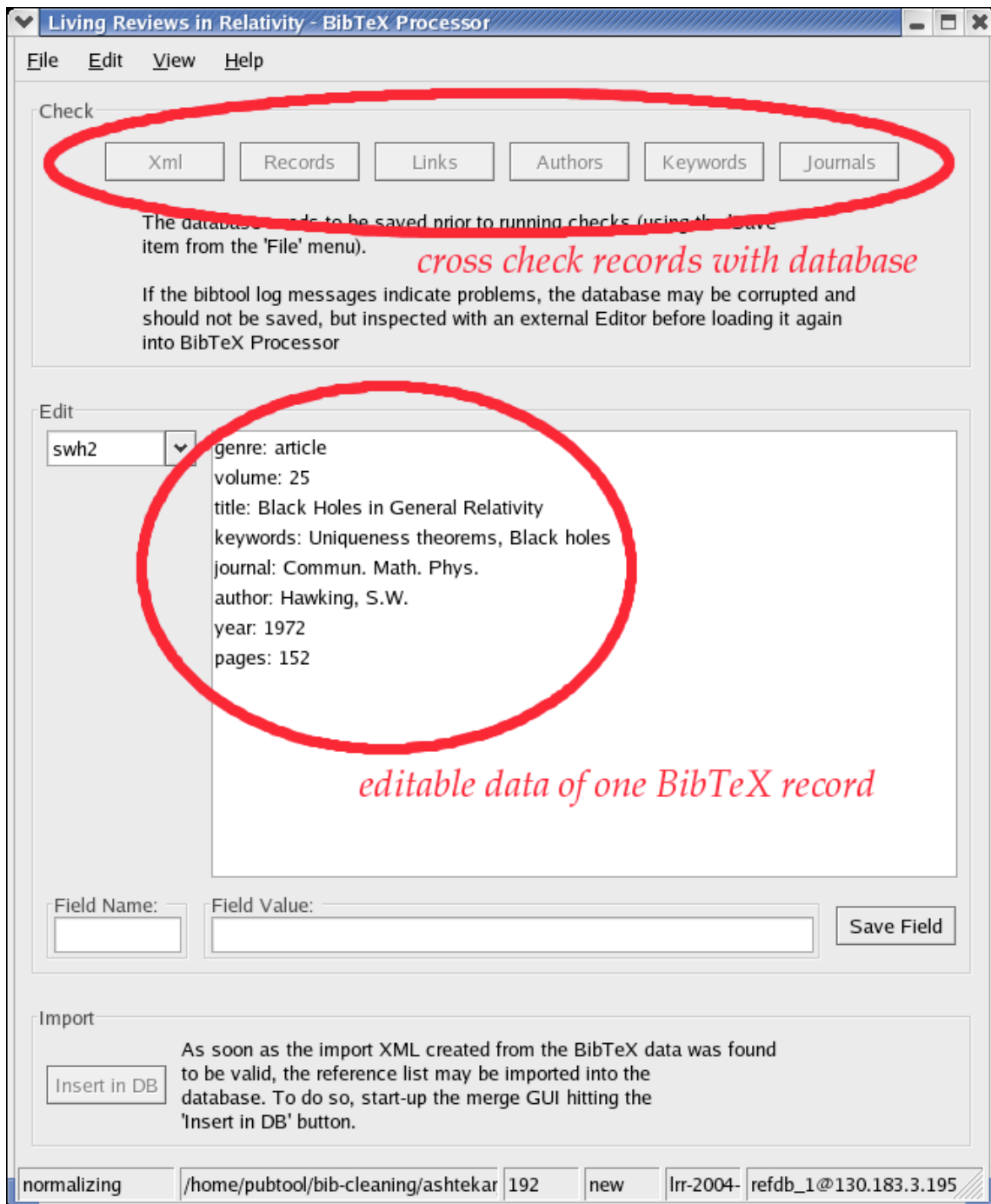


Figure 4: BibTeX import filter application for the reference database

The *eims* component provides an “Editorial Information Management System” - basically a web application supporting our publication workflow, which includes a peer-review system. Why did we roll our own online peer-review system? Again, this is mostly due to what we publish:

- The invitation process is a big part of the workflow - and not supported at all in typical document tracking solutions which start with submission (see Fig. 5).

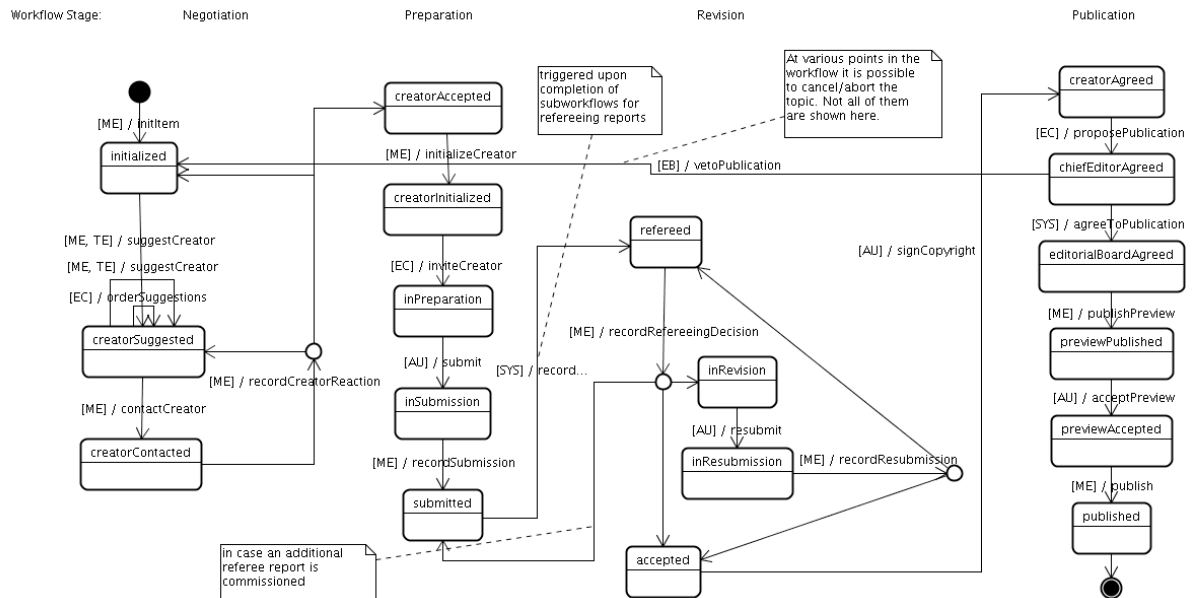


Figure 5: Publication workflow for Living Reviews journals

- Our online publications are rich web applications of their own (as an example see <http://solarphysics.livingreviews.org/Articles/lrsp-2005-3/resources/tadpolemovie/index.html>) which doesn't fit well with the model of most publishing systems where a publication is typically *one* PDF file or one HTML page.
- Our low publication rate doesn't make automation as pressing as not to be able to do it by ourselves.
- The revisions/updates of publications must be supported, i.e. these relations need to be modelled in the system.
- Migration of our legacy data is important!

eims can be used separately as peer-review or document-tracking system and as such is probably the component with the most competing products. For an example of the web interface see Fig. 6. The history of a publication reflects the stages of the workflow the publication has already passed.

Editorial Information Management System

Living Reviews in Relativity

Navigation in three Layers: Meta, Main and Context navigation, as known from trac.

logged in as robert | [Logout](#) | [Settings](#) | [Help](#) | [Admin](#)

Home	Tasks	Reports	Messages
new message	edit	revise	recordRevision

Topic

Updated by [redacted] (Update)

by [redacted] ([redacted])

Topic Editor: Beig, Robert ([redacted]) Priority: normal

Stage: online Status: published

Submission Date: 2003-08-07 00:00:00 Item ID: 213

File: [item.txt](#) URI: [http://www.livingreviews.org/\[redacted\]](http://www.livingreviews.org/[redacted])

Subject Fields:

- Mathematical Relativity

Link to report on updated version of the topic

Links to reports on the Referee Reports for the topic (Referee Reports are handled as first class items in the systems, just like topics.)

Referee Reports:

- Item #214, [redacted] (Referee Report 1)
- Item #215, [redacted] (Referee Report 2)

History:

Action	Actor	Executed	Recorded	Public Comment	Backoffice Comment	
Initialize Workflow Item	system	2002-11-25	2006-02-13			Attachment
suggestCreator	system	2002-11-25	2006-02-13		[redacted] (#659) suggested;	
contactCreator	system	2002-11-25	2006-02-13			
recordCreatorReaction	system	2002-11-25	2006-02-13		[redacted] accepted.	
initializeCreator	system	2002-12-02	2006-02-13			
inviteCreator	system	2004-03-15	2006-02-13			
editItem	system	2004-03-15	2006-02-13		old submission date: 2003-04-30 00:00:00.	
editItem	system	2004-03-15	2006-02-13		old submission date: 2003-08-31 00:00:00.	
editItem	system	2004-03-15	2006-02-13		Submission was delayed 1 time(s).	
recordSubmission	system	2003-09-15	2006-02-13			
recordRefereeReports	system	2006-02-13	2006-02-13	automatic		
recordRefereeingDecision	system	2004-01-16	2006-02-13		toBeRevised	
resubmit	[redacted]	2004-02-11	2006-02-13			
recordResubmission	system	2004-02-11	2006-02-13			
recordRefereeingDecision	system	2004-02-27	2006-02-13		accepted	
signCopyright	[redacted]	2004-02-27	2006-02-13			
publishPreview	system	2004-03-15	2006-02-13			
acceptPreview	[redacted]	2004-03-16	2006-02-13			
publish	system	2004-03-16	2006-02-13			
recordRevision	system	2005-06-01	2006-02-13			Attachment

Workflow actions in the invitation stage

Between recordSubmission and recordRefereeReports the report items complete their own workflow!

Note: Sensitive data like topic title and author name have been blacked out.

Powered by ePubTK

Figure 6: Screenshot of a detailed report on a publication item from EIMS

The *webstats* component of ePubTk is a thin layer integrating the log analysis software analog and ReportMagic with the configuration mechanisms of ePubTk. To allow publication specific feedback, a *blog* component has been adopted recently. Again the approach was to integrate existing software – in this case pyblosxom – with ePubTk. The posts in the are generated from publication metadata exported by *register* to RDF+XML. For an example see <http://solarphysics.livingreviews.org/discussion/>

3.4 Interfaces

Components of Living Reviews publishing platform interface with each other mostly via ad-hoc file formats, exported by one and imported by another component (see Fig. 7). While this approach prevents entangling the functionality of different components too much, it also highlights the summary of our comparison with other projects above. They clearly call for standards, and possibly for integration via web services.

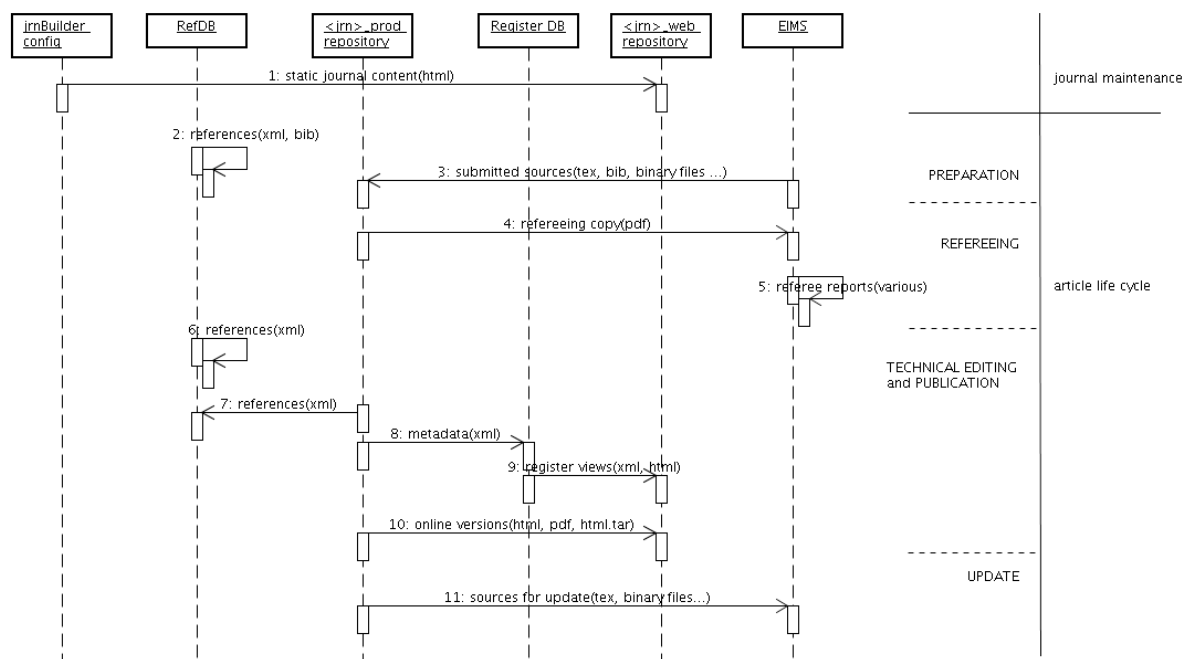


Figure 7: Interfaces between components of the Living Reviews publishing platform

4 Lessons Learnt

1. *Keep the tool zoo small* - The central tools we use at Living Reviews are subversion, trac and ePubTk. While none of these three (except maybe trac) is particularly easy to use, the fact that each of them is used for multiple tasks helps getting up the learning curve. In particular the fact that different user groups use the same tools helps building up a common knowledge base.
The prime example for this is subversion. Managing editors use it to maintain the web content, technical editors use it to handle the manuscripts, developers use it for version control of the software.
2. *Don't reinvent the wheel – unless ...* - While we try to make use of existing tools (possibly using them in a novel way), a big part of our toolset is the ePublishingToolkit, which does reinvent some wheels to make sure they fit.
3. *Embrace diversity* - Very diverse people need to collaborate in our project: linux, mac and windows people; scientists and non-scientists; non-collocated people; developers and users. While this demands a lot of attention, it also makes for good experiences: windows people attacking command lines, linux people looking for /etc/hosts on windows machines – in general, tackling steep learning curves with a team. You better like communicating (digitally).
4. *Open Source* - Simply is the way to go! It makes experimenting easy – since you don't have to buy licenses first; it's fun – and possible - to get involved; you can learn by examples and actually use what you learned. The prime example here is trac. We liked the user interface. So we just plucked it and put it on our EIMS – thus keeping user experience uniform across applications.
5. *Technical staff is necessary...* - ... because software is never really finished, automation is always wanted, maintenance just has to be done (server updates, migration of legacy data). On the other hand, we found that

people who want to start a journal and people who can support the technical infrastructure for an online journal are rarely the same. So to keep the barrier for new journals low, we provide a hosted solution.

6. *Plan for change* - Migration issues have been a constant companion in our project: We migrated several GB of web content (where tidy proved very helpful, but xhtml or clean html4 from the outset would have been better). We are just in the middle of migrating the old editorial information database (and it turns out, that an API for the DB would be way better than dumping CSV files). The easiest migration task so far has been migrating the register. Since the data was already in XML it took only one XSL transform. Hence:
 - put emphasis on portability/migratability of data,
 - avoid any dead-end formats,
 - put apis on applications or even better, store in format which is easy to export right away.

Again, change would be a lot easier, if standards existed.

7. *Find a niche* - For our journals so far, the niche created by our special form of content proved helpful. In particular, the advantage of online publishing with a mechanism to update publications seems more clear cut in the case of review articles, which follow the developments of a research field.

References

- [1] Living Reviews Portal. <http://www.livingreviews.org>
- [2] DALCHOW, C.; NENTWICH, M.; SCHERHAUFER, P.; WEYHER, C. "Living Reviews – Innovative Resources for Scholarly Communication bridging diverse spheres of disciplines and organisational structures". *Accepted contribution for elPub2006 conference, "Digital Spectrum – Integrating Technology and Culture", Bansko, Bulgaria, 14 – 16 Jun 2006*. In preparation.
- [3] GAP. <http://www.gap-portal.de/>
- [4] Digital Peer Publishing. <http://www.dipp.nrw.de/>
- [5] Digital Publishing Systems. <http://dpubs.org/>
- [6] ESPERE. <http://www.espere.org/>
- [7] ePubTk. <http://dev.livingreviews.org/sec-cgi-bin/epubtktrac>
- [8] subversion. <http://subversion.tigris.org/>
- [9] trac. <http://projects.edgewall.com/trac/>