

Like a version? No thanks, I'll have a revision.

Tom Worthington has been dropping names again. Mine this time, and Martin Dougiamas'.

Tom's thinking about version control for academia, [wondering if it's a killer app](#). He tells a story about meeting a professor who is swamped with documents managed in email, of all places:

During the World Computer Congress in Brisbane I talked to a university professor who was [having problems with email](#). They were keeping copies of all messages from students and book collaborators in the university mail system. As a result their mail box was clogged with copies of draft papers and books. This was partly a failure of the university's administration in not providing suitable online tools for handling large documents which will be required for several years. However, it was also a conceptual failure. When I asked them about version control, they looked at me blankly. I had assumed that this would be something familiar to any professional handling large amounts of information, but it appears to be confined to engineers and IT professionals.

I don't think this is just about version/revision control. It's multidimensional. And while version control looks like a no-brainer to a luminary of the Australian Computer Society, just like XML publishing or TeX, look like great ideas to various computer scientists who have seen the benefits, none of those things have really stuck with general users. My approach is to look at the consumer-level tools that are successful at building a user-base without training courses or coercion, and think about how we can apply the same ideas. I'm not saying that Tom's wrong, just that we need to look at solutions that help people transcend the limitations of their current systems without them having to learn complicated new stuff, 'cos most of them just won't.

We all need:

1. **Collaborative storage spaces** for workgroups who need to work on the same files and might all need to edit the same thing.
2. **Collaborative systems** for seeking input or comment from people who aren't in our immediate cohort:
 - i. Document review by teams, committees.
 - ii. Comments on draft documents such as theses, by supervisors.
 - iii. Marking systems for assignment assessment.
 - iv. Review systems for peer-review or formal document approval processes.
3. **Publishing systems and channels** which increasingly need to cater for web and mobile-web consumption, and ideally should have a feedback/discussion channel so people can engage with your stuff.
 - i. To course participants, often in a Learning Management System such as Moodle.
 - ii. To the world, via blogs etc.
 - iii. Via publishers and conferences.
 - iv. To our university administration for compliance reasons, we need to be able to deposit data for retention, and published works in the repository, etc.

Version control is pertinent in all of these contexts but it has different flavours and requirements. Tom's right, email is not the way to do any of these effectively, except in its role as a notifying service, but at least it does capture snapshots of file versions as-mailed, which is more than most file-systems do.

There are lots of systems that academics currently have to use to do the above content management tasks, which may or may not keep track of 'versions' in their own way, more or less effectively. Here at USQ SharePoint is used by a lot of workgroups for collaboration, for example. It has some kind of version control built in. Then there are systems like OJS for managing journals where there is a form of version control, in that one version of paper is sent to reviewers and things move through the journal publishing lifecycle. And

if you post something to a blog, that creates a kind of point-in-time snapshot of a document (as long as you don't go changing it later).

We're working on range of systems at ADFI, using a mixture of building things that don't exist and interfacing with things that do.

My current thoughts follow. Just thoughts.

Collaborative services

For local collaboration [DropBox](#) is a killer app. You can see and use the same files across multiple devices (for me that's 3 Linux computers, an iPad and an Android phone) via miraculous file replication. It does version control in the sense that you can go back to an old version of something (revision control, really). It stays out of the way until two people edit the same thing at the same time, then it gracefully creates a new version of one of the files, labelled as 'conflicted', and the authors can sort out the mixup. We need a version of this that can be run at the institutional or workgroup level. Seriously. (And don't talk to me about the ARCS data fabric, WebDAV does not work. It doesn't. Not like DropBox).

When we built [ICE](#), using the Subversion system as a replication back end, it was something like DropBox we were really striving for. Now that it's arrived, running ICE over the top of DropBox is a really good way for teams to collaborate – everyone can access shared files, they automatically turn into web pages, and groups can comment on a documents via the web-view. We do this to maintain the CAIRSS document intranet, and publish the [website](#). Give or take a few ICE bugs that works really well. This is a design pattern we'll definitely pursue with our work on The Fascinator repository toolkit – having a system sit on top of a shared file store and provide a web view of content that you can use to organize, annotate and publish that content.

We're looking at other design patterns, to help with annotation/discussion and web conversion services for people who might be committed-to or stuck-with systems like SharePoint, Sakai, Drupal et al.

Publishing

One of the projects ADFI is working on is a repository to sit behind the Moodle LMS, to manage learning and teaching content of all kinds. This could well end up providing:

- A quality controlled repository of courseware, including Open Access courseware that is flagged as such.
- A place for teachers to collaborate on course materials prior to publishing.
- An engine for learners and teachers to discuss and engage with content and each other, via an interface that appears in Moodle.
- A system to automatically convert stuff into the formats required for delivery on various platforms including mobile, and paper.
- A way to bring all this above together, to allow (say) collaborative annotation on phones and PCs.

In the academic publishing world things are a bit harder to influence, but I am still trying to find a way to make progress on the idea of [Scholarly HTML](#) - there's an upcoming workshop in the USA I'd like to attend, and some promising discussion with JISC in the UK about the potential for ePub publishing.

Some final thoughts on version control

Finally, Tom's post has prompted me to try to write down some of what I've been working on re version

control. What's a model that will work for your general academic?

For close collaboration with colleagues I think we need to supply something like DropBox, a magic file system that makes sharing automatic. Version control in this environment would be simply the ability to go back to previous versions if needed. (This is really what I think most theorists would call 'revision' control'), I have used this a few times in DropBox to get myself out of trouble – but I'm not sure its anywhere near as important as the magical ability to have your files available anywhere, including via the web.

When do you need an actual version of a file, rather than a revision?

I think it's basically when you share something – and you need to be able to agree/prove that a particular version of a file is what you're both working with. Sharing is a kind of publishing.

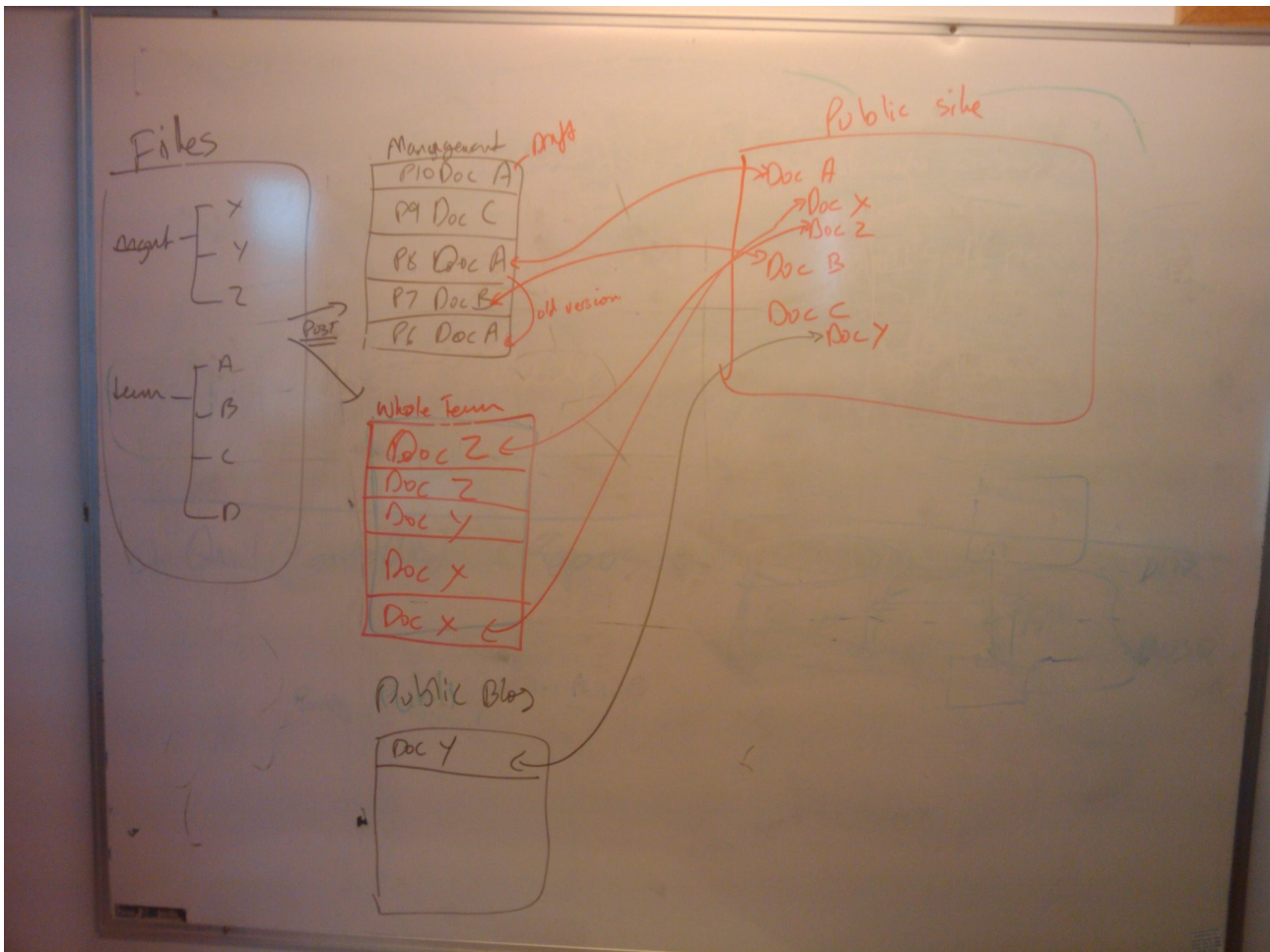
While DropBox is good for pre-publication sharing, I think it would be good to add to the features in DropBox to handle the act of making a published version of something. Sure, I can use it to share a folder, but what I want to be able to do is a kind of publishing – to create a stable version of a resource that's URL addressable, and that others can comment on and discuss, but not necessarily edit. The idea would be to take a file, a document, image, dataset, whatever and make a blogable version of it – DropBox can't do this, but the ADFI software stack we've been building for years **can**.

I've been wondering if the blog publishing model might not be a good way to do this in a number of contexts. After all, the simplest version of a weblog is just a series of time-ordered resources. You could post the same item multiple times, and get snapshot versions of it that persist over time. Remember, blog is short for **weblog**. You get a **log** of document publishing events.

Want to share something for comment? 'Blog' it.

The act of publishing gives your document/file a URL that locks it in time. You could have a 'blog' channel set up for a student-plus-supervisor, for example, and others for team-collaboration and sub teams. Add-in sophisticated commenting features and you have a collaboration environment. We have been (slowly) working on this with our Anotar project, [adding paragraph level annotations to WordPress](#) – which would make this a fairly simple thing to set up for anyone with a server (or you could use <http://digress.it>).

I've got a complicated diagram on my whiteboard, which tries to show an architecture for how a set of blogs might sit in between files storage (a la DropBox) and things like public sites, or learning management systems. The idea would be that you would push 'files' to different public 'blog' spaces for discussion/approval, and from there to things like public websites, or the LMS etc. It's probably a stupid architecture, but I do think that we need to look carefully at how we can let people have 'version control' without them having to learn about it. I mean, I work in a team of people who can all use Subversion to manage their code, but nobody is really inclined to use it for our main file-sharing / management.



Copyright Peter Sefton, 2010. Licensed under Creative Commons Attribution-Share Alike 2.5 Australia.
<http://creativecommons.org/licenses/by-sa/2.5/au/>



This post was written in OpenOffice.org, using templates and tools provided by the [Integrated Content Environment](#) project and published to WordPress using [The Fascinator](#).