Systems support for collaborative research projects: position statement and discussion

David Aspinall

24th February 2007

This is a note about desirable systems support for collaborative research projects, especially collaboration is with users external to the School of Informatics, and especially where the collaboration includes developing software. I consider "infrastructure" type of support here, rather than support for research application software (which I would dearly prefer CO effort to be spent on). I explain what infrastructure has been needed for the Proof General project and how it is provided, and discuss some possibilities for future provision.

1 Background

I started discussing these issues with various people around December 2006, particularly in connection with the ISDD and the reasons that I do not use it for Proof General. I was interviewed by Morna Findlay as as a precursor to a survey of research support requirements, which I believe, has not yet taken place. There has been some recent discussion by COs about to what extent we are actually allowed to grant users outside the University access to our systems (see `http://www.inf.ed.ac.uk/systems/policies/AccountsForNonEd.html`). And there is also an obvious connection to the discussions about what the baseline support support could be for research projects in the new resource model for research computing support, and how the new resource model accommodates funds spent outside the School/University.

These issues deserve to be discussed more widely and some sensible policies proposed.

Instead of trying to be comprehensive and talk about what other projects need or may need, I'll just mention what I have needed so far for the Proof General project; to what extent these needs are currently met (part of the baseline research support, I believe); the alternatives I've thought of, and some of the issues. Hopefully these points can be discussed at the relevant committees, perhaps in conjunction with a more comprehensive document prepared by someone else.

2 Proof General needs

My basic needs at the moment are:

Backups : Reliable and accessible backups of all project data.

Web : A web server to serve up web pages with a modest amount of dynamic content in standard scripting languages (not full Web 2.0: I am not using a database backend at the moment, although it is a future possible need). Content written by researchers. Low traffic volume.

Downloads : Web/ftp space to provide downloads, ideally with download counting and possibly click-through license agreement and registration. Total download data up to about 1G.

Mailing lists : mailing lists which can be subscribed to by internal and external users (two lists: one for users, one for developers; both very small volume).

SCM : A standard source code management system which allows external read-only anonymous access as well as read-write access by both local and external named collaborators. Repository tree size about 600M (with history, probably double). Ideally both CVS and its emerging replacement, Subversion.

Wiki : A wiki space which can be edited by external and local users, used for collaborating on research ideas, research software usage and development. Small traffic, small amount of content.

Tracking : A project management/bug tracking system (ideally Trac, otherwise Bugzilla) which can be used to provide more organised planning and support of the research project and its software.

It is truly impressive and a credit to our systems support that most of these needs are met, or almost met, already. The current status is:

Backups : Fully provided by standard DICE provision.

Web Fully provided by the project web server hosting `proofgeneral.inf`, with a similar configuration to `homepages.inf`. There are some issues over the security, exact configuration and facilities provided by this server, but these could be addressed if it were given a configuration more like `homepages.inf`. (This might require pseudo-users on that server for projects, however.)

Downloads Provided by the ISDD, but I haven't used it so far. The reasons are that (a) it requires separate and manual upload for each download provided and (b) each different version (platform, distribution, daily snapshot release, etc) requires a separate entry and upload to the database.

I could provide stable versions of Proof General on particular platforms, but these would not give representative download figures. I think the ISDD should be provided with ways to aggregate different versions of the same program, and upload software automatically.

Mailing lists : Fully provided by `lists.inf`.

SCM Fully provided by `cvs.inf` for CVS.

Partly provided by `svn.inf` for Subversion. Unfortunately Subversion has not been configured for anonymous access and does not allow external access except for users with a DICE account (this is an issue with how it has been set up: anonymous access is possible to configure with direct http connection, at least).

Wiki Partially provided. Project areas are available on `wiki.inf` but write access is restricted to internal users (more specifically, users with DICE accounts). Apparently there are ideas for a UCS provided University-wide wiki system but it is likely that this would be similarly restricted.

Tracking Not provided.

In the last few months I've spent time installing my own version of TWiki (`http://proofgeneral.inf.ed.ac.uk/wiki`) and the Trac bug-tracking system (`http://proofgeneral.inf.ed.ac.uk/trac/`). This is difficult within the limitations of the project server configuration, and without access to standard install locations for additional libraries, etc. I'm acutely aware that the resulting installations pose security risks both to my data and to the School's systems, more than they ought. The installations also don't work as well as they should (part of the appeal of Trac is an intimate connection to Subversion, but it must be hosted on the same server to work). So they should be be improved and ideally officially supported, or they should be moved elsewhere in the near future.

3 Alternatives

Of course needs evolve continually and different projects may want to use different basic mechanisms. It's unreasonable to suppose that baseline research support (or even, project-specific support) can provide everything in-house. So a sensible way forward might be to provide mix of research collaboration provision in-house, but also to formulate a policy on outsourcing support by using hosted solutions.

3.1 In house

How much of this should we aim to supply in-house in Informatics? Are we making the provision in the right way? Some possibilities are:

- Continue as we do now: provide piecemeal and (probably costly) bespoke DICE-based installation and customization of collaboration tools, as resource allows. Some of this may be baseline, some perhaps paid for from research grants.

  Paid-for facilities may need to be supported beyond the life of the project[1] and there should be possibilities for sharing costs, or re-deploying infrastructure provision from one project to another at lower cost (since the initial effort has already been paid for). I suppose this has been discussed already in the context of the new resource model.

- Possible alternative: move to an integrated project management resource which would (presumably) be easier to deploy. INRIA, for example, uses the GForge system (`http://gforge.org`) to host its collaborative projects. Collaborative users have to click through an agreement.

- Possible alternative: provide our own virtualised hosting services and devolve management of virtual machines to research projects (non-baseline). This has the enormous advantage that research software can be given a longer shelf-life by taking a snapshot of particular operating system and software versions.

  At first sight this appears to leave a much higher burden to the research users (or the costed CO support) and ignore the advantages of aggregated support. But basic operating system installations (even standard versions of DICE) are quite easy, and with greater access privileges it is much easier to install and configure bespoke applications.

[1] Loss or decay of research prototypes at the end of a project impedes future research; there many cases where this has happened. Virtualisation is probably the best way forward to take snapshots of installed research prototypes. But some collaborative projects may anticipate continued development beyond the life of the project, so SCMs, etc, may need to be maintained.

Providing such a service would probably cost a lot of CO effort to use open-source solutions (Xen or UML), but providing a basis using paid-for solutions (e.g. VMWare or similar) might be relatively easy.

3.2 Out of house

Perhaps research projects should be allowed, encouraged and even supported in their use of external services. Already there is a lot of take up by individuals in Informatics of externally provided services (including Google Calendar, Google Analytics, Flickr, Blogger, etc).

For project infrastructure management we may chose:

- A free service: for example, Sourceforge `http://sf.net`.
  Sourceforge is enormous; it provides web space, database, shell hosts, CVS, Subversion, bug-tracking, mailing lists, bulletin boards. However, it has a number of disadvantages: software licenses must be open source (perhaps non-exclusively, I haven't checked); free access is cluttered with advertising, and the long term survival of the company supporting the site (VA Software) has been called into question repeatedly.

- A commercial service: for example, `http://www.hosted-projects.com`.
  This is a German company that provides hosted Subversion and Bugzilla/Trac installations. They say "using hosted-projects.com has many advantages, which, in most cases, outweigh the disadvantages by far."[2] Pros:

  - Cost-effective. A similar hardware and software setup would cost a lot more when you host your projects yourself.
  - You can concentrate on your actual work instead of having to do administrative work
  - Ideal for geographically spread out teams who need fast network access from anywhere in the world
  - Fast and competent support available to help you with any issues you may have
  - We handle backups for you

  Cons:

  - Source code leaves your office
  - Access through the Internet is inherently slower than LAN access

  Their current charge for 1000MB SVN disk space 100MB WebDAV disk space with unlimited developers and repositories is $15/month.

[2] Pasted from: `http://www.hosted-projects.com`

The legalities and contingencies of using external services need to be considered in detail. Our research collaborators may have their own rules about what external systems they are allowed to use and where they can submit content. Even if we let source code outside the University we should probably have our own mirroring and back-up provision.

A concern was put to me by a computing officer that this idea would mean that if this was taken to the extreme, our own computing officers would be reduced to providing the most basic (and therefore boring) infrastructure. But I don't think this should be the view at all: rather it is more likely that we could enable more CO resource for applications-level support on research projects, really capitalising on the highly-skilled CO staff that we have. This kind of support is dearly needed, difficult to obtain from funding agencies, but could make a real difference to the output of our research.

I've lately benefited from this kind of help, thanks to some funds from Alan Bundy's platform grant which paid for time of Graham Dutton. Graham helped me with application development and debugging, maintenance for API changes in Eclipse, packaging and building the distribution, and managing the web pages and Trac bug database. The experience was excellent, but unfortunately all too short. A project like Proof General requires continued support help of this kind.