

Final Report

Project name/acronym:	FROCKLE
Project website address:	http://frockle.oucs.ox.ac.uk/ http://wiki.oucs.ox.ac.uk/ltg-public/Frockle/FinalReport
Report author(s):	Colin Tatham
Date:	09/11/2007

Methodology

Following the project plan, we adopted an incremental and iterative approach to development. We produced an initial user interface within Bodington with place holders for the intended functionality. This was demonstrated to a small group of users to gauge requirements and initial response. We then enhanced the user interface and added further functionality, which we demonstrated to the WebLearn User Group.

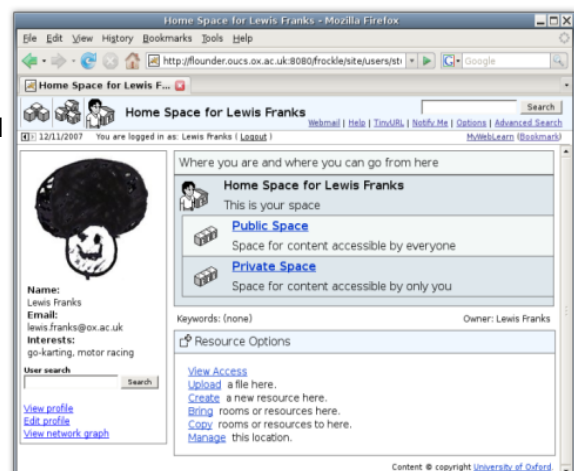
Throughout the development process we used a standardised development environment consisting of Eclipse, Ant and SVN, in order to make development collaboration within the team and with external developers easier. We assessed the eProfile toolkit code, build process, repository and documentation on an ongoing basis and gave feedback to the original developers where appropriate. We also modified the Bodington initialisation code as development progressed, in order to produce an easily deployable demo of the new functionality, including dummy data. We anticipated incorporating code into the production VLE service progressively during the project.

Implementation

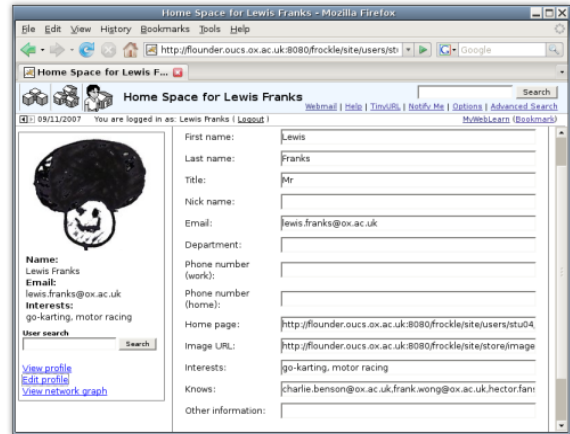
Initial work was to install and experiment with the eProfile toolkit components (which proved to be more time-consuming than we had planned for). We then looked at how the eProfile web service and TouchGraph applet could be incorporated into both Bodington and Sakai, and decided to mirror the class hierarchy of Sakai's existing profile tool, in order to ease later integration into Sakai. We developed the initial user interface in Bodington, following the Bodington community's decision to write all new tools using Spring MVC and JSPs instead of either of the Bodington custom templating schemes (thereby also improving the possibility of re-use). We also researched the possibility of using Ruby on Rails for prototyping, and RSF (a templating scheme based on JSF, in use in the Sakai community) but decided that there wasn't sufficient time in the project.

Description of user interface.

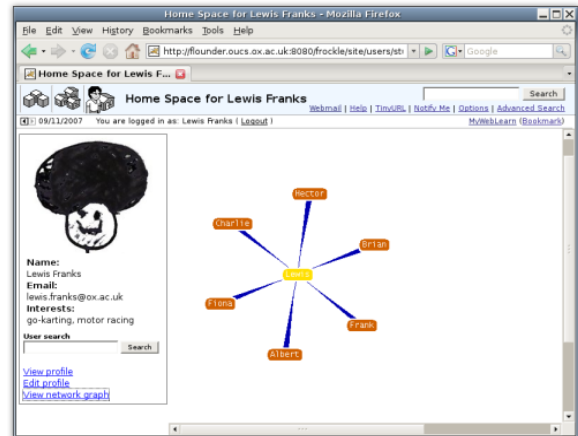
- The user's personal space now contains a modified left hand panel, containing a photograph of the user or other image and a summary of their details and research interests.



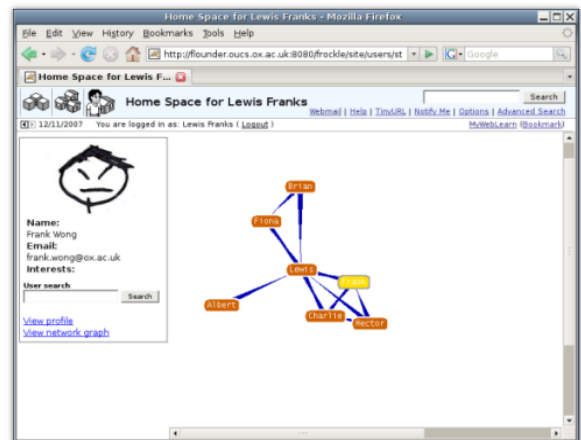
- Links are provided to view the user's full profile details, or edit their profile if they are logged in.



- Another link loads the TouchGraph applet which displays a visual representation of a network of linked users.



- Right clicking on a node in the network displays that user's profile summary in left hand panel
- Clicking on the linked user's image displays their full profile.



We set up a publicly accessible SVN repository at the start of the project, and made use of SVN externals to pull the eProfile toolkit and Bodington (WebLearn) code in to the project, in order to ensure the code was up to date, and to avoid duplicating code held elsewhere. We produced Ant build files not only for the FROCKLE code, but also for the eProfile web service and eProfile applet, as well as JUnit tests for the eProfile web service, and contributed them along with any bugfixes and repository rationalisations we thought necessary. It became apparent that we'd like to make quite a few changes to the way the toolkit functioned, as well as needing a few modifications in order to make it suit our needs better. In agreement with the eProfile developers, we gained additional funding and an extension from JISC in order for the eProfile developers to do the work. The additional work ⁽¹⁾ included modifying the web service to deliver FOAF profiles in an XML format, adding more FOAF fields to the profiles, some enhancements to the applet, and some suggestions for re-factoring. The additional work has been completed, and updated versions of the toolkit components will be integrated into the demo installation in due course.

(1) <http://wiki.oucs.ox.ac.uk/ltg-public/Frockle/ToolkitEnhancementRequirements>

Outputs and Results

- Modified Bodington code.
- Spring MVC based code and JSP pages suitable for re-use.
- JUnit tests and Ant build files for eProfile toolkit.
- Contributions to eProfile SVN, and feedback to developers.
- Dissemination:
 - Website: <http://frockle.oucs.ox.ac.uk/>
 - Wiki: <http://wiki.oucs.ox.ac.uk/ltg-public/Frockle/>
 - Blog: <http://frockle.blogspot.com/>
 - Demonstrated to users, WebLearn User Group.
 - Summary in Computing Services annual report.
 - Described in pre-conference seminar at Educause 2007.

All code produced during the project is held in the following repositories:

FROCKLE: <https://svn.oucs.ox.ac.uk/projects/vle/frockle>

eProfile client: <svn://chimera70.essex.ac.uk/eProfileClient/branches/frockle>

eProfile web service: <svn://chimera70.essex.ac.uk/eProfileWebService/trunk>

Demo available at: <http://flounder.oucs.ox.ac.uk:8080/frockle/>

Download available at: <http://downloads.oucs.ox.ac.uk/frockle/>

Implications

All of the code produced is available in the FROCKLE and eProfile repositories. The Frockle repository does depend on the eProfile repository being available, so it is arguable that we should keep a local snapshot as a backup. The software developed was not integrated into the production VLE service during the life of the project, due to concerns about its maturity, but the work will be taken up again during the current migration of the institutional VLE from Bodington to Sakai.

Conclusions

- Applets seem to be problematic in general. We experienced problems with loading and running correctly in some browsers. Hopefully the forthcoming 'Consumer JRE' project will address these issues: <http://java.sun.com/developer/technicalArticles/javase/consumerjre/index.html>
- It seems that FOAF is not particularly widely used.
- Gauging by initial user reaction, uptake would probably be slow, as the potential benefits aren't yet obvious.
- Some kind of integration/interaction with other profiling applications, like FaceBook, would produce benefits, but there wasn't sufficient time for implementation. TouchGraph has recently completed a Facebook integration: <http://www.touchgraph.com/TGFacebookBrowser.html>
- In retrospect, it wasn't realistic to plan integration into the production VLE service during the project, as it hampered development progress. It would probably have been better to concentrate on a simple demonstrator to be enhanced after the project ended.
- We made use of the opportunity to research multiple topics that will be of benefit in future work and collaborations.