Managing a Distributed Development Project: The Subject Portals Project

The Project

The RDN's Subject Portals Project (SPP) is funded under the JISC's DNER Development Programme. There were two proposals, SAD I (Subject Access to the DNER) and SAD II. The original SAD I proposal was part of a closed JISC DNER call, 'Enhancing JISC Services to take part in the DNER'. The SAD II proposal was successful under the JISC 5/99 call, 'Enhancing the DNER for Teaching and Learning'. The original project proposals are available [1].

The aim of the project is to improve the functionality of five of the RDN hub sites to develop them into subject portals. Subject portals are filters of Web content that present end users with a tailored view of the Web within a particular subject area. In order to design software tools that simultaneously satisfy the needs of a variety of different sites and make it easier for institutional portals to embed our services in the future, we are designing a series of Web "portlets". One portlet will be built for each of the key portal functions required, focussing initially on authorisation and authentication (account management); cross-searching; and user profiling; but including eventually a range of "additional services" such as news feeds, jobs information, and details of courses and conferences. The project is committed to using open source software wherever possible.

The hub sites involved in the SPP are EEVL (based at Heriot Watt University, Edinburgh), SOSIG (University of Bristol), HUMBUL (University of Oxford); BIOME (University of Nottingham) and PSIGate (University of Manchester). The project is managed from UKOLN based at the University of Bath, and the technical development is led from ILRT at the University of Bristol.

Distributed Development: The Problems

The fact that the SPP partners are geographically dispersed has posed a number of challenges. Since the objective of the SPP is the enhancement of the existing hub sites, hub representatives have naturally wished to be closely involved, both on the technical and on the content management sides of the project. At the last count, 38 people are involved in the project, devoting to it varying percentages of their time. But this means that physical meetings are difficult to organise and costly: since work began in December 1999 on the SAD II project, only two full project meetings have been held, with another planned for the beginning of 2003.

Smaller physical meetings have been held by the technical developers at ILRT and the five hubs, but these again are extremely time-consuming.

We also faced the problem that many of the project partners had never worked together before. Not only was this a challenge on a social level, it was also likely to prove difficult to find where the skills and experience (and software preferences) of the developers overlapped, and at the beginning of 2002, the then project manager Julie Stuckes commissioned a skills audit to discover the range and extent of these skills and where the disparities lay. It was also likely to be hard to keep track at the project centre of the different development activities taking place in order to produce a single product, and to reduce the risk of duplicating effort, or worse, producing incompatible work. We also thought moreover that it was desirable to develop a method of describing the technical work involved in the project in a way easily understood by the content managers and non-technical people outside of the project.

The Solutions

We tackled the problem of communication across the project by the use of a project JISCmail mailing list [2]. The list is archived on the private version of the SPP Web site [3] where other internal documents are also posted.

The developers have their own list (spp-dev@dev.portal.ac.uk) and their own private Web site [4] which is stored in a versioning system (CVS - Concurrent Version System [5]) which gives any authenticated user the ability to update the site remotely.

In addition the developers hold weekly live chat meetings using IRC (Internet Relay Chat [6]) software (as shown in Figure 1), the transcripts of which are logged and archived on the developers' Web site.

Using IRC means the developers are able to keep each other informed of their activities in a relaxed and informal manner; this has aided closer working relationships.

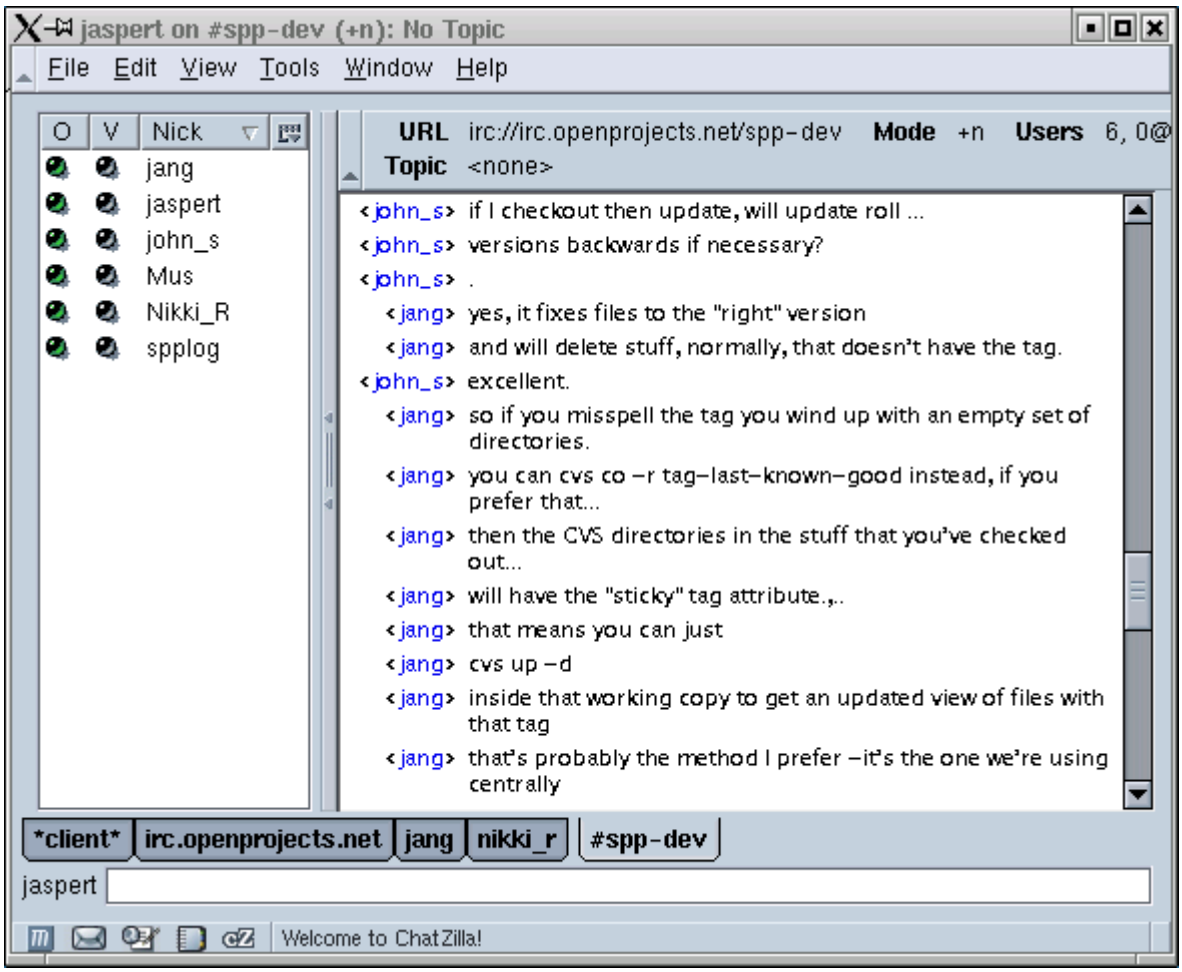

Figure 1: Example IRC Session

As well as holding the developers' Web site, CVS also contains the project's source code and build environment. This takes the form of a central repository into and out of which developers check code remotely, ensuring that their local development environments are kept in step. A Web interface also provides the option of browsing the code, as well as reviewing change histories. Automatic e-mail notification alerts the developers to updates checked into the CVS repository, and all changes are also logged. This has proved an essential tool when co-ordinating distributed code development.

The other part of the software development infrastructure is providing a build environment that takes care of standard tasks, allowing the team members to concentrate on their coding. Using a combination of open-source tools (e.g. **ant** [7] and **junit** [8]) a system has been created that allows the developer to build their code automatically, run tests against it, and then configure and deploy it into their test server. As well as this, the build system will also check for new versions of third-party packages used by the project, updating them automatically if

necessary. This system is also managed by current project down from the central repository, build, configure and deploy it, having it running in a matter of minutes.

Because of the widely dispersed team, the difference in software preferences and the mixed technical ability across the project, we looked around for a design process that would best record and standardise our requirements. UML (Unified Modelling Language [9]) is now a widely accepted standard for object oriented modeling, and we chose it because we felt it produced a design that is clear and precise, so making it easy to understand for technical and non-technical minds alike. UML gave us a means to visualise and integrate use cases, integration diagrams and class models. Moreover using UML modelling tools, it was possible to generate code from the model or update the model whenever the code was further developed.

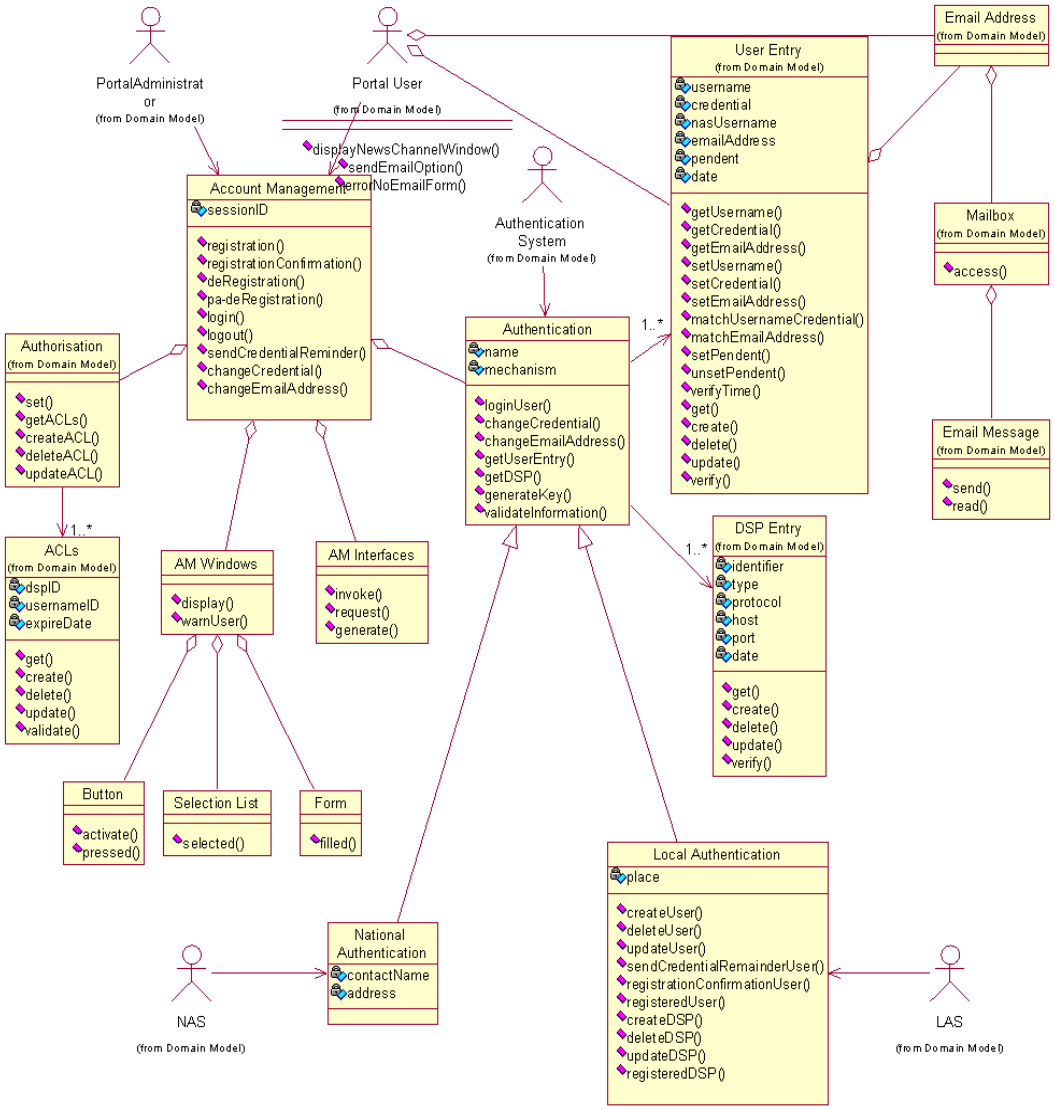

Figure 2: Example UML Diagram

Finding UML software that had all the features needed was a problem: there are plenty of products available but none quite met all our requirements, especially when it came to synchronising the work being done by different authors. Eventually we opted to use the **ICONIX** process [10]. This is a simplified approach to UML modeling, which uses a core subset of diagrams. This enabled us to move from use cases to code quickly and efficiently using a minimum number of steps, thus giving the technical side of the project a manageable coding cycle.

Additional funding was obtained from the JISC in order to bring one of the authors of the ICONIX process (Doug Rosenberg) over from California to run a three day UML training course. Although this course was specially designed for SPP, places were offered to other 5/99 projects in order to promote wider use of this methodology across the JISC community. Unfortunately, despite early interest, no other project was represented at the training, although Andy Powell, the technical co-ordinator for the RDN, attended the course. Additional funding was also received from the JISC to purchase licences for **Rational Rose** [11], which we had identified as the most effective software available to produce the design diagrams

Finally, to provide greater structure to the project, a timetable of activities produced using MS Project is posted on the private project Web site and is kept continually up to date. A message is posted to the project mailing list to alert partners of any major changes to the timetable.

What We Would Have Done Differently

It would have been sensible for us to have adopted a process for software development at an earlier stage in the project: it was perhaps a need that we could have anticipated during the SAD I project phase. Also, it is worth noting from our experiences that getting the communications and technical support infrastructure in place is a job in itself, and should be built into the initial planning stage of any large and dispersed project.

Continuing Problems

Electronic communication is still no substitute for face-to-face meetings so the SPP development team continue to try to meet as regularly as possible. Time is inevitably a major problem wherever project partners have other work commitments: all the project partners based at the hub sites have to juggle SPP work which is for the project as a whole, with that which relates particularly to their own hub's adoption of the project's outcomes. Increasingly, as the project develops, less work will be required from the project "centre" and more at the hubs, leading to an eventual handover of the subject portal developments to the hubs for future management.

The Future

It is our plan to make use of UML diagrams in the final project documentation to describe the design and development process. They will offer a detailed explanation of our decision making throughout the project and will give future projects an insight into our methodology. Andy Powell was also so impressed with UML that he is planning to use it across development work for the RDN in the future.

The future development of the SPP beyond the end of the project is likely to be led by the technical development partners, for instance in the continued development of the portlets to enable them to be installed into alternative open source software platforms to make the technology as compatible with existing systems as possible. It is therefore greatly to the benefit of the project that they have become such an effective and close working team.

Contacts:

Ruth Martin, SPP Project Manager
UKOLN
University of Bath
Bath
BA2 7AY

Email: r.martin@ukoln.ac.uk

Jasper Tredgold, SPP Technical Co-ordinator
ILRT
University of Bristol
10 Berkeley Square
Bristol
BS8 1HH

Email: jasper.tredgold@bris.ac.uk

References:

1. **Subject Portals Project**, RDN, <http://www.portal.ac.uk/spp/documents/>
2. **Archives of SPP Mailing List**, JISCMail, <http://www.jiscmail.ac.uk/archives/spp.html>
3. **RDN portal development project**, RDN, <http://www.portal.ac.uk/spp/private/> (Note username and password required)

4. **dev.portal.ac.uk**,
   <http://dev.portal.ac.uk/>
5. **CVS - Concurrent Versions System**,
   <http://www.gnu.org/software/cvs/>
6. **Internet Relay Chat (IRC)**,
   <http://www.irchelp.org/>
7. **ant**, Apache,
   <http://jakarta.apache.org/ant/>
8. **junit - Testing Resources for Extreme Programming**,
   <http://www.junit.org/>
9. **UML**,
   <http://www.uml.org/>
10. **ICONIX**,
    <http://www.iconixsw.com/>
11. **Rational Rose**,
    <http://www.rational.com/products/rose/>

QA Focus Comments

The SPP project (initially known as SAD I and then SAD II) was funded by the JISC's 5/99 programme.

Brian Kelly, QA Focus, 4 November 2002

Citation Details

Managing a Distributed Development Project: The Subject Portals Project, Martin, R. and Tredgold, J., QA Focus case study 03, UKOLN, <http://www.ukoln.ac.uk/qa-focus/documents/case-studies/case-study-03/>

First published 4th November 2002.

Changes

20 May 2004

Minor changes made to format of QA Focus Comments. Brian Kelly.