JISC

Project Information			
Project Acronym	SPP		
Project Title	Subject Portals Project (continuation)		
Start Date	1st May 2006	**End Date**	30th June 2007
Lead Institution	Institute for Learning and Research Technology University of Bristol		
Project Director	Lesly Huxley		
Project Manager & contact details	Jasper Tredgold jasper.tredgold@bris.ac.uk +44 (0) 117 928 7068		
Partner Institutions	None		
Project Web URL	http://www.spp.opensource.ac.uk/		
Programme Name (and number)	Portals programme		
Programme Manager	Balviar Notay		

Document Name			
Document Title	Final Report		
Reporting Period	2006-2007		
Author(s) & project role	Jasper Tredgold, project manager		
Date	12th December 2007	**Filename**	spp_final_report.pdf
URL	http://www.spp.opensource.ac.uk/wiki/spp/docs/spp_final_report.pdf		
Access	✔ Project and JISC internal	✔ General dissemination	

Document History		
Version	**Date**	**Comments**
0.9	2007-07-31	Internal draft for project team comments
1.0	2007-12-12	Circulated Proposed Final version
1.1	2008-01-22	Submitted to JISC

Subject Portals Project (continuation)

Final Report

Jasper Tredgold

jasper.tredgold@bris.ac.uk

12th December 2007

Table of Contents

1. Acknowledgements

This project gratefully acknowledges the funding provided by JISC and it was undertaken within their Portals Programme. The project team would also like to thank various projects and individuals for their time and effort spent in collaborations with this SPP work. This includes the team at the JISC VRE Programme's IBVRE project, Mark McLaren of the University of Bristol Portal team, Chris Awre from the CREE portal project and Matthew Dovey and the Jafer team. We'd also like to acknowledge the early involvement of the PerX project, although our differing timetables prevented us from working together more fully.

2. Executive Summary

After nearly a two year hiatus, the Subject Portal Project (SPP) received a final tranche of JISC funding with the over-arching aim of ensuring that some of the portlet [PORTLET] work done within the original SPP was successfully contributed into the community as Open Source software. Within this the objectives were to support users of the original software, build upon relevant developments since the end of the first SPP effort, evaluate and disseminate, and make software available via established Open Source repositories.

The project was hampered early on by a delayed start and the loss of a key partner, however new partners were located, the project objectives were altered accordingly and the focus of development was successfully shifted, such that, by the end of the project an Open Source portlet and supporting infrastructure had been delivered and housed on SourceForge.net, one of the primary Open Source software development repositories. In addition work had been contributed to enhance another portlet which will be used in the current JISC project CREE Extension.

The project managed this by using software development methods influence by agile practices. These helped endure the development stayed on course and that any potential problems were highlighted early enough to ease the process of dealing with them.

The outputs of the project included Open Source software, an article and various reports relating to the current portal and portlet landscape.

In conclusion, the project managed to achieve much of what it set out to, although with a different emphasis than anticipated. SPP portlet software was made available as Open Source to the community. One recommendation worth highlighting is that regarding the need for further exploration of successful methods of community building around Open Source projects.

3. Background

The Subject Portals Project (SPP) [SPP] was originally funded under the 5/99 Programme to improve the functionality of the Resource Discovery Network (RDN) subject hub sites. It built and packaged portal-based software using open source Java-based portal technologies. However, when the RDN

re-launched as Intute [INTUTE] and reviewed its technical infrastructure in light of a changed funding environment the decision was made not to pursue the use of portal technologies as the platform of delivery for their subject hubs.

However, a short inspection of the current portal landscape makes clear the continuing developments in both the technology and the take-up since the end of the previous SPP work. There are a raft of commercial and Open Source offerings, including those from the major software vendors [IBM, ORACLE]. The emergence of standards in this area (JSR 168 [JSR168] and WSRP [WSRP1]), their continued active development (JSR 286 [JSR286] and WSRP 2.0 [WSRP2]) and their successful adoption within the industry shows that the idea of a portal as a delivery platform for aggregations of content is still a live one, as does the newer non-standards based developments by the likes of Google [GOOGLE] and Netvibes [NETVIBES].

In addition, within UK Higher Education there are an increasing number of institutions choosing portal platforms to deliver customized institutional information to their staff and students, as well as using VLE platforms to manage the delivery of course content and VRE platforms to aid research communities. There are examples too of products in these latter categories that are, or plan to be, JSR 168 compliant [SAKAI]. There had also been more work and research done under the auspices of organisations like JISC, for example the Contextual Resource Evaluation Environment [CREE] project, conducted under JISC's Portal Programme.

So at the outset of this project it was clear that the portal landscape had developed and matured in the time since the end of the previous SPP work. As such there was still an argument that there was useful material within SPP that could be made more readily available via Open Source, specifically in the largely subject-based areas of Z39.50-based searching and literature resource alerting.

4. Aims and Objectives

The broad aim of the continuation project was one of ensuring that some of the software developments from the SPP were contributed effectively to the community. In more detail, this comprised:

- Contributing standards-compliant portlets to the HE and FE community (and beyond) and establishing the framework for their support within the open source community. It was envisaged that the portlets will be deposited in appropriate repositories (e.g. the JA-SIG Clearinghouse [JASIGCH]) while the source code will reside in an open source development repository (e.g. SourceForge.net).

- Responding to maintenance and development requests from key software users. In particular, the RDN service EEVL Xtra [XTRA] and the PerX [PERX] project both made use of SPP software at the time of the first inception of the project. As part of the original work plan they provided us with a list of suggested SPP maintenance and enhancement tasks. However, incompatibilities in

project timetables eventually prevented our project and the PerX project from being able to collaborate. In mitigation, a working relationship was formed with both the IBVRE project [IBVRE] (regarding the Alerting Portlet), and the CREE Extension project (regarding the Searching Portlet). The work plan was updated in places to reflect this change.

- Reviewing and building upon relevant research done elsewhere, for instance the CREE project, and exploring routes towards interoperability with frameworks other than portals. Of particular interest in this area is the SAKAI framework.

- Exploring and demonstrating potential routes for portlet deployment in production settings by developing case studies involving real-world institutional portal platforms. Disseminating project outcomes via appropriate community channels.

5. Methodology

As evident from the project plan the largest part of the effort spent during the project was concentrated in the area of software development. This comprises for the most part work package 3 and some of work package 4. For this reason much of the project methodology concerned software development best practices, and the approaches considered and adopted are discussed below.

The dissemination and evaluation work packages (5 and 6) required the involvement of users external to the project. The techniques applied are also discussed below.

In addition to these, various pieces of desk research were undertaken, including work package 2 and the reports on the impact of new portlet standards.

5.1. Software Development

Process

Within the development team there was some experience of differing software development techniques. However, recent experience of agile-based methodologies [AGILE] had produced favourable results and experiences. Having discussed the relative benefits of the various agile approaches and reviewed some of the literature available about them, the decision was taken to adopt a methodology that ensured key agile practices were supported:

- Involving users, as the key determinant of successful product 'fit' to requirements.

- Utilizing short iterative cycles of incremental development.

- Concentrating on face-to-face communication over heavy-weight process documentation.

- Using unit-testing as a measure of software compliance.

Although the team did not strictly adopt a single formal agile method, the techniques used were drawn from the various proponents of agile processes. In particular the technique of short two-week development cycles was adopted, with a team meetings marking both the end of a cycle and the beginning of the next. These were termed milestone meetings.

Legacy code

As well as considerations regarding new code development, the methodology had to accommodate the fact that project's starting point was a legacy code-base, the output of the original SPP work which ended mid-2004 [SPPARCHIVE]. This required that one of the initial project activities was to review and assess the legacy code in terms of its current applicability and the options for its re-factoring and re-use.

User involvement

One key part of the agile approach is the close involvement of product users as part of the development team. These users are deemed to be the authoritative source on requirements and their feedback is a key part in measuring whether software is successful in meeting a requirement or not. Effort was directed throughout the project in identifying and managing that relationship with those project stakeholders that took on that user role.

Supporting tools

As usual in a project of this type, it was important to ensure effective version controlling of the software during development. A standard system (Subversion [SUBV]) was adopted at the outset and integrated into the Java IDEs in use within the team. While the repository was obviously a key point of interoperability within the team (all the developers must be able to commit and check-out changes), the particular IDE platform a developer was not seen to be. As long as repository interoperability was maintained the development platform was not mandated but left to developers' preference. This resulted in the use of both Eclipse [ECLIPSE] and IntelliJ IDEA [INTELLIJ] within the team.

In order to manage the progress of work throughout the project a combined Wiki and ticketing system (Trac [TRAC]) was adopted. The Wiki allowed internal project documentation to be developed collaboratively, while the ticketing system allowed work tasks to be identified, specified and tracked against time-scales. This system was the basis for the measurement of progress against scheduled milestones, facilitating the easy review and planning of tasks at each milestone meeting.

The development strived to follow coding best practices. This include source code documentation, coding to interfaces, loosely-coupled class design and unit-testing. In addition the team utilised appropriate third-party Java libraries frameworks, most notably the Spring application development framework [SPRING], including the portlet components, and the Hibernate persistence framework [HIBERNATE].

Standards

The development addressed interoperability by ensuring adherence to the two key portlet standards: JSR-168 and WSRP. Being compliant with these two standards was seen as the best route to compatibility with the broadest array of possible portal platforms. All the main platforms are now JSR-168 compliant as are, or will soon be, some platforms not traditionally seen as being portals, e.g. Sakai. WSRP provides a standard for delivering portlets to portals remotely. This is an important standard to support as it provides a scalable model for portlet provision.

In addition the project considered the likely impact of the upcoming JSR 286 and WSRP 2.0 portlet standards, both currently in development.

Scalability

Scalability was addressed at the application level by ensuring that the software design allowed for the distribution of components across more than one hardware platform.

5.2. Testing and Evaluation

Testing was addressed at two broad levels: the testing of code and the testing of functionality against user requirements. For the former various techniques were used, including unit-testing and integration testing. As regards the latter another methodology was employed, that of recruiting example end-users, asking them to test the software and then gathering and analysing their feedback.

As well as helping the team judge whether functional requirements had been fulfilled, this user-based testing provided much data for the overall project evaluation, as well as material for the dissemination work package.

6. Implementation

Project start-up

Although the project formally started in May 2006 very little effort was available for the first few months. This was due in combination to a delayed start, a short funding lead-in and the fact that staff were already committed to finishing an existing JISC VRE project. Over the course of the year the total effort spent on the project was as planned, though the distribution was not. Most of the effort was expended from late Autumn 2006 onwards. By Autumn 2006 the team was fully in place.

Project partnership changes

The late project start in relation to its original inception resulted in problems with the originally planned collaborations however. The PerX project, who played the stakeholder role of chief end-user in the initial plan, had their own funding-related time constraints which became incompatible with the shifted

timings of this project. Although a couple of their maintenance issues were addressed very early on (relating to OAI harvesting and memory management) it became clear by the Autumn of 2006 that the partnership would have to end and the project plan be re-assessed. With agreement of the programme manager, the plan was altered to remove the specifically PerX-related activities.

To compensate for this change new partnerships were sought by the project team. This lead to a collaboration with the JISC-funded IBVRE project in Oxford around the development of the alerting portlet and a collaboration with the CREE Extension project around the search portlet.

The loss of PerX as a partner also changed the emphasis placed on the maintenance and re-use of the existing code base. PerX had been using the searching portlet, but without their involvement, and with the new involvement of CREE and IBVRE, the emphasis shifted in two directions.

Firstly the development of the alerting portlet became took increased prominence. This was the area IBVRE were especially interested in. IBVRE contacted the project regarding this functionality initially, and were happy to take on the role of user stakeholders with regards its development.

Secondly, the CREE Extension project was at the stage of making plans for the start of their work come mid-2007. The original CREE project had, amongst other things, produced a proof-of-concept portlet based on code from the Jafer [JAFER] project. Re-factoring the SPP search portlet to use Jafer code had been part of the original project plan, but, after discussions with Chris Awre (CREE) and Matthew Dovey (Jafer), it was decided a sensible approach would be to use SPP search portlet effort to contribute to the continued development of the Jafer portlet.

Desk research

The initial desk research work reviewing the current state of portal frameworks was completed by the end of 2006. This report can be found on the project's website [SPP].

Infrastructure

The infrastructure required for the development work was also put into place. This comprised a SVN repository, a Trac instance and an internal mailing list and file space. The Trac instance provided an internally-facing Wiki and task-management subsystem. Within this, the development roadmap was created, with face-to-face milestone meetings every two weeks. This managed the core work-flow for the development team over the following few months. At a milestone meeting the team would review progress on the tasks slated for completion at the milestone just ended. Those still outstanding would be discussed and re-scheduled as appropriate. Work would then proceed as laid out in the roadmap for two weeks when the team would then meet again and repeat the milestone process.

Software development

The SPP software existing at the end of the previous SPP work, late 2004, was reviewed in the early stages of the continuation project. The main developer on the continuation project had not been involved in the previous SPP work. The code areas that were focussed on in particular were those that complimented the work of the continuation project, i.e. the work related to the Alerting Portlet and the Search Portlet.

It was unfortunate that areas that much original SPP effort had gone into could not be re-used in this project. Worth particular mention was the work on authentication and authorisation and the work on a portlet-neutral data storage schema. The latter has been made somewhat redundant by the JSR 168 standard and its inclusion of interfaces to store and retrieve portlet data. The former was tied into both the portal framework (Apache's Jetspeed version 1 [JETSPEED1]) and the Athens authentication system [ATHENS]. Both of these have subsequently been or are in the process of being superseded.

Alerting Portlet

Development proceed on the alerting portlet during 2007. IBVRE provided an initial set of user requirements for the alerting functionality and this, and the above-mentioned review of the existing alerting code provided developers with a starting point for their new work. Low fidelity prototypes were provided to IBVRE to confirm the team's understanding of their requirements. Once these were agreed design and development started in earnest, following the two-weekly milestone cycle described in the methodology section above.

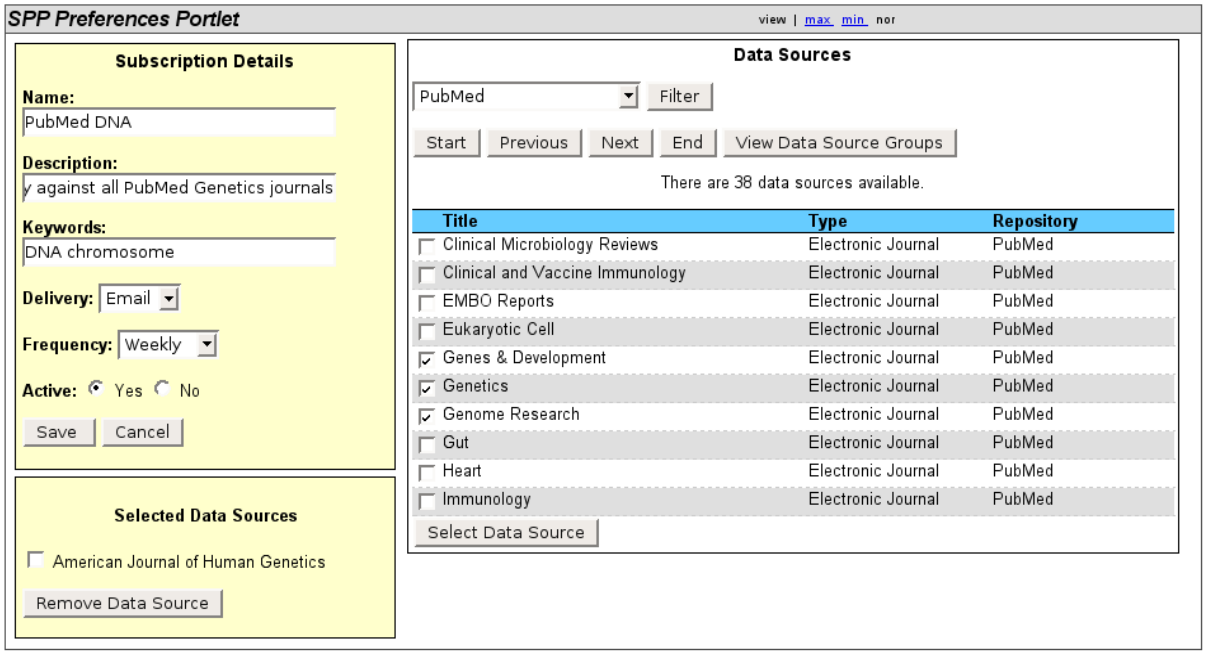

Figure 1: Alerting Portlet screenshot

The architecture approach adopted was designed to ease implementation and increase scalability by breaking the infrastructure into components that were functionally distinct and that communicated via

well-defined interfaces. An overview of the architecture is given in Figure 2. An advantage of this approach were that it allowed the two developers involved to concentrate on their components safe in the knowledge that so long as the component met the agreed interface requirements, interoperability would be ensured.

The communication vehicle adopted for traffic between the components was one of a message queue (JMS). This provided for asynchronous communications between components such as the harvester and the harvest-triggering application. An example of data flow between components in given in Figure 3.

Although this approach required the use of a third-party component, the message broker, it was felt that any cost involved in this was outweighed by the advantages of flexibility in component co-ordination that the solution offered. This could provide benefits as regards future configuration and scalability.

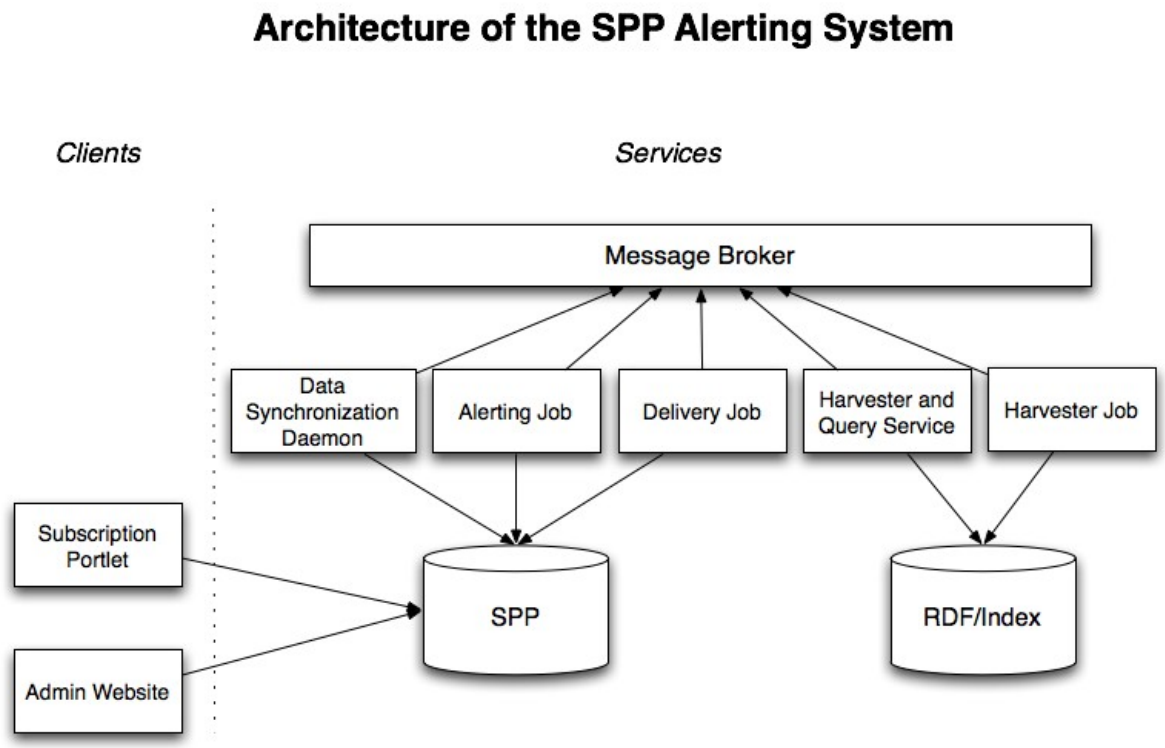

Figure 2: Alerting architecture

The Spring framework was adopted as the platform for several of the components. Spring has proved itself to be a valuable Java application framework and it proved adept in helping the developers make good design decisions and in removing some of the more repetitive tasks involved with developing Java applications. Hibernate was also used in places to provide a persistence layer.

The components were developed largely during the end of 2006 and the first half of 2007, leading to a point when user testing could be undertaken, in May 2007.

On-line Alerting Portlet documentation can be found on the project web site:

http://www.spp.opensource.ac.uk/spp/AlertingPortlet

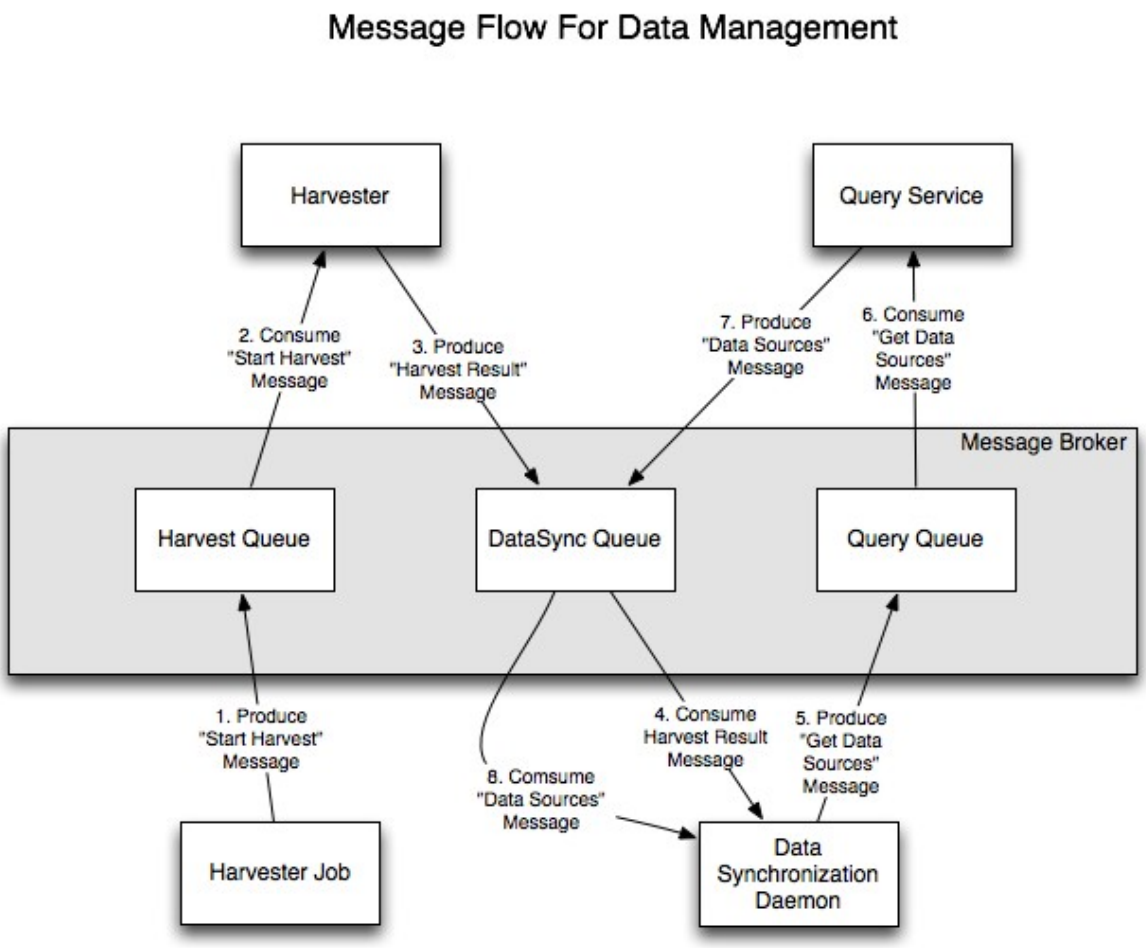

Figure 3: Example alerting message flow

Search Portlet

The direction of the developments within the Search Portlet effort changed considerably after the loss of PerX as a project partner. The project's replacement 'client' was effectively the CREE Extension project, and it was in discussions with them, and Matthew Dovey of the Jafer project, that the new priorities of development were established.

Rather than continue to develop the search portlet existing at the end of the original SPP work, the effort was to be spent enhancing the Jafer Search Portlet [JAFERPORTLET]. This portlet had been developed after the original SPP work as part of the first CREE project. It was based on the Jafer Z39.50 work – work that this project had intended to utilise during any further development of the SPP search portlet in any case. Taken with the fact that the Jafer portlet was JSR 168-compliant (the SPP search portlet was not natively compliant) and the fact that CREE Extension project was due to start

mid-2007 and wanted to continue the development of the Jafer search portlet, the decision to use our effort to fold SPP changes into the Jafer search portlet seemed a good one.

Although not all the areas of work discussed with CREE and Jafer were able to be addressed, the project did make some useful code contributions to the Jafer software, including the addition of a search history. Details of these are discussed in the Outputs section below.

On-line Search Portlet documentation can be found on the project web site:
http://www.spp.opensource.ac.uk/spp/SearchPortlet

Funding timescales

The project suffered in a couple of ways regarding the staffing of the team and the funding timescales.

The period from the conception of the original continuation project to its execution was unfortunately over-long: some 15 months. In addition the time from the final official confirmation of project funding to the project start date was short, some 2 months. These two facts had implications for the availability of staff, some were committed to existing projects until several months into the new SPP continuation project. As such little effort was available to the project during its first 4-5 months. Although the staff were then able to expend extra effort in the remaining months of the project, and the total effort was as originally planned, the effect of the late start and the effort imbalance was to make our timescales incompatible with our original project partners PerX, as discussed elsewhere.

Use cases, evaluation and dissemination

As the Alerting Portlet and its supporting infrastructure neared a point at which it could be usefully tested work commenced in earnest on the use cases and evaluation. Two specific types of use case were highlighted: that of the researcher end user, a particular user community with an interest in being kept up-to-date with current developments in their specialist fields, and that of the portal administrator, a technical user tasked with the responsibility of installing and maintaining the portlet and its infrastructure.

To address the first use case two researchers were located as volunteers via our relationship with the IBVRE project. A test installation comprising uPortal [UPORTAL] and the Alerting Portlet was set up and the users given accounts. The Alerting Portlet was configured to harvest journals from the PubMed OAI-PMH repository [PUBMED]. The two users were asked to try out the portlet and create and manage subscriptions. After a suitable time, our project researcher conducted an interview with them to discover their experiences.

As regards the portal administrator use case a similar approach was undertaken. A portal administrator was identified as a volunteer. He was asked to install, configure and run the software using solely the information contained within the distribution. Afterwards, the same approach was taken to eliciting thoughts and opinions on the software – an interview was conducted. In addition, the tester took the trouble to produce a written report on his findings.

Towards the end of the project the team's researcher undertook to write an article for Ariadne [ARIADNE] on the Alerting Portlet and her user evaluation findings.

Additionally the project web site [SPP] and SourceForge.net project site [SFSITE] were made available. The Clearinghouse [JASIGCH] contribution was imminent at the time of writing, just awaiting the migration of the Subversion repository to SourceForge.net. More details are provided in the following section.

7. Outputs and Results

7.1. Software

The primary outputs of this continuation project were software components in two areas, the Alerting Portlet and the Search Portlet. In addition, some work was done regarding user evaluation, both with end users and portal administrator users, and this was done in the context of use cases.

Alerting portlet

By the end of the project the Alerting Portlet had taken the place as the main project deliverable. This was not the original project intention, but was a decision taken in light of changes to our external collaborating partners, as described elsewhere in this report.

The overall aim of the Alerting Portlet was to allow end users a means of keeping abreast of new resources appearing across a wide range of sources. For example, a researcher may want to be informed about new journal articles in their subject area as they appear in any one of a number of online journals. In order to provide this functionality various infrastructural components are required:

- The user portlet. This portlet allows the user to create and manage their alert subscriptions. A subscription comprises a set of data sources (for example, e-journals), some keywords, and option settings including the frequency with which the alert should be checked, and the alert delivery mechanism (at present only email is supported).

- The administrative web interface. This interface, provided by a servlet, allows a portal administrator to define the data sources that should be made available to their end users. These data source definitions include details as to the means of machine access (at present only OAI-PMH repositories are supported).

- The harvester. This back-end component manages the actual harvesting of those data sources configured by the administrator. It also provides an index and a search interface to the harvested data. Harvested data is converted internally to RDF and stored locally in an RDF database. The execution of alert subscriptions result in SPARQL queries being generated and run against this RDF dataset.

- Timed tasks to trigger both the checking of the user subscription database, executing subscriptions that are due to be checked, and the refreshing of the data harvested from the external data sources.

- The delivery component. This back-end component manages the delivery of any alerts created as a result of subscription executions. At present only an email delivery component has been implemented.

This software has been released under an Open Source BSD licence [BSD] and is available from SourceForge.net [SFSITE].

Search Portlet

Originally it had been envisaged that the Search Portlet would be the focus of development effort under the continuation project. However, due to changes in the project partners, as described elsewhere, it was deemed appropriate to re-focus effort on the Alerting Portlet and use the remaining Search effort on folding changes into the already existing Jafer Z39.50 portlet. This portlet was the result of effort undertaken by the Jafer team within the CREE project, and the work areas for the SPP developments were decided in discussion with both Jafer and CREE. The modified search portlet could then become a useful input into the CREE Extension project.

Developments undertaken by the SPP team included:

- Search History. Searches undertaken by a user (within a single session) are recorded upon execution and can be browsed via a user interface icon. These previous searches can be re-run or edited.
- Single Search Portlet. An additional portlet that displays only the results of a single saved search. The search to be run is defined via the EDIT portlet mode. The search is run on submission from within EDIT mode, and can be re-run from the VIEW mode by use of a refresh icon.
- Contextual help. Depending on the user's location within the various VIEW portlet mode screens, switching to HELP mode will allow text contextualised to that VIEW mode screen to be displayed. This help text would be supplied by the portal administrator.
- OpenURL integration. JISC's OpenURL router was configured into the presentation of records.
- User interface edits. The user interface was improved, including the addition of Tango Desktop Project icons.

The Jafer search portlet, with SPP additions, is available from the SPP web site [SPP].

7.2. Reports

The project also produced various written outputs as part of its work. Unless otherwise mentioned, these reports are all available from the project web site [SPP].

- Ariadne article. This article was published at the end of the project and looked at the Alerting Portlet work and the evaluation done around it. It can be found on the Ariadne web site [ARIADNE].

- Portal Framework review. Towards the beginning of the project a review was undertaken to discover the current state of the available portal frameworks. This was a valuable exercise and aided the team in understanding how the portal landscape had changed since the ending of the previous SPP work in 2004.

- JSR 286 and WSRP 2.0 impact report.

- Report on portlet inter-communication use cases and implementation routes. Included in the JSR 286 and WSRP 2.0 report above.

- Use cases and evaluation reports. Evaluation of the Alerting Portlet was carried out during the latter stages of the project. The use cases of the end user researcher and of the portal administrator were considered. Volunteers were located and asked to test the software in various ways. After a period of time they were contacted and interviewed about their thoughts and experiences. Feedback from this exercise resulted in some useful modifications to the software prior to the end of the project.

8. Outcomes

The main overarching aim was to ensure that some of the original SPP work made its way into the community, and by meeting the objective of releasing Open Source portlet software the project fulfilled a key criterion of this aim. The focus of the software released was not entirely as originally envisaged, for reasons detailed above, but it was in line with the modified project plan.

The project's work with the IBVRE project around the development and evaluation of the Alerting Portlet showed that the functionality it offered would offer clear benefits to end users, in particular, researchers. A system that pushed alerts to users informing them of relevant updates to resources of interest to them was seen as a very useful addition to their research tool-set. In addition, portal sites (for example, institutional portals) that hosted the Alerting Portlet could also benefit from the improved service offering for their users.

The project's work around the Jafer Search Portlet is probably of less immediate use to portal administrators and their users, but it captures some of the search-related user interface work done

within the original SPP and places it in a context that makes it available to those conducting future work with the portlet, for example the CREE extension project.

For future projects it is worth recording this project's experiences with aspects of the adopted project methodology. Broadly speaking the use of Agile software development methods worked, and allowed the team to manage the software's requirements and effect change where and when required. The use of an issue ticketing system [TRAC] and frequent milestones helped with the capture and monitoring of all development tasks. Version control software is of course essential, as is the active involvement of the software's end users, or clients. In our experience it is easy to overlook this area of the development chain, or to leave it to the latter stages of the project and regard it as part of the testing work. Agile methods emphasize the centrality of the client as the source of key information about the functional domain. They are the people who ultimately can say whether the software is performing 'correctly', since they are the people who specify the functional requirements. However, it can be difficult in projects such as these to locate and involve a suitable 'client'. In our case, one of the early project tasks was to locate such a client (PerX, and then IBVRE). Their involvement was crucial both to the requirements capture and the evaluation of the functional fit and utility of the resulting portlet.

A recognised feature of successful and sustainable Open Source software projects is the active community that they enjoy, both in terms of developers and users, though of course the latter will always outnumber the former. In attempting to seed such software this project, with hindsight, should have spent more time addressing the question how do you build a community around your software. The project met some of the key criteria, in that it used recognised development infrastructure and then hosted the work on a public Open Source repository, but the strategy was to produce working code and then place that in the public domain. This did not leave enough project time to follow up community development. Instead the community work to the form of active engagement with other JISC projects within the relevant domain, certainly an improvement over no engagement at all, but given project timescales (for example, IBVRE finished during this project's lifetime) and funding issues perhaps not the best way to maximise the chances of software sustainability post-project.

This is a perennial problem for JISC Open Source software development problems and it is good to see it being addressed through, for example, Open Source Watch [OSSW]. Some of this project's developers attended a OSS Watch seminar entitled “Communities for software development” (unfortunately just post-project end). This very usefully discussed the importance of and techniques for the active building of communities around software development during the lifetime of JISC projects. These lessons (for example, see it as a key, effort-requiring, part of the workplan, develop in public from the outset, release early and often) chimed with those learnt from this particular project.

9. Conclusions

It was clear from the review work undertaken within the project that standards-based portlets should still be viewed as a useful platform for delivering functional software components. The continuing development of the Java portlet standard (JSR 286) and its related remote portlet standard (WSRP 2.0) serves to extend potential portlet functionality and overcome some of the limitations of the previous standards. This, the scope of commercial standards-based portal offerings, and the still increasing size of the installed portal base at educational institutions suggest that portlets are very much a live technology. However, of course, it is wise to be aware of the status of proprietary solutions too (e.g. iGoogle [GOOGLE]) and seek to maximise interoperability by, for example, providing alternatives to the portlet application view layer.

To our knowledge, there is not another portlet available that provides the alerting functionality that the SPP Alerting Portlet provides. This functionality was well received by our evaluators and project collaborators and could be a valuable area for future development.

The project's use of Agile methodologies was rewarding and successful. It proved, at least to the team members, the benefits of pursuing development along these flexible lines and provided an incentive for further adoption in future projects, especially of XP and SCRUM practices [AGILE].

The issue of successful community building around an Open Source project is another valid conclusion from this project, as discussed above. Within the relatively short lifetime of a typical JISC Open Source project this is an area that requires early attention and a high priority. Also key is the adoption of techniques designed to maximise the chances of a community being built. These include: development in public; open communications (e.g. mailing list/web forum with public archive); appropriate management tools (e.g. issue tracker, version control); and public dissemination (e.g. blog, wiki). Following these best practices should increase the chances of software being sustained post funding.

10. Implications

The Alerting Portlet is of course open to further development by dint of its public Open Source status, although at present the nominal development team consists only of those who were funded to work under this project. One potentially interesting future work direction would be to develop the Alerting Portlet in the context of a service pilot within a particular specific user community. Thus far the development of the alerting functionality was very much driven from a generic research use case, though provided from the IBVRE project community. A useful way to progress the work would be to formulate a project, attached to a particular subject service, that aimed to provide a pilot alerting service tailored to their user community. The software developments that ensued should of course remain subject-neutral but much would undoubtedly be learnt from the real user feedback generated by a live pilot service.

The Search Portlet work has been made available publicly and interest has been shown in it by the CREE extension project which will be looking into search presentation on portal platforms.

Future work should also consider the nearly-released portlet standards JSR 286 and WSRP 2.0 and the impact upon the functionality that portlets can offer. It should be noted though that this would be medium-term developments. The specifications are nearing final release at the time of writing, and it is to be expected that it will take some time before there is much of a deployed base of portal platforms that will support them. Being able to support JSR 168 and WSRP 1.0 will still be of importance to portlet developers for some time to come.

11. References

[AGILE]

For a discussion of the various agile software development methods the Wikipedia article is a good place to start: http://en.wikipedia.org/wiki/Agile_software_development

[ARIADNE]

Ariadne: a web magazine for information professionals in archives, libraries and museums
http://www.ariadne.ac.uk/

[ATHENS]

Eduserv Athens
http://www.athensams.net/

[BSD]

The BSD License template
http://www.opensource.org/licenses/bsd-license.php

[CREE]

Contextual Resource Evaluation Environment project
http://www.hull.ac.uk/esig/cree/index.html

[ECLIPSE]

Eclipse IDE
http://www.eclipse.org/

[GOOGLE]

iGoogle portal
http://www.google.com/ig
Though see http://blogs.sun.com/portal/entry/syndicating_portlets_to_igoogle for a discussion of bridging portlets to iGoogle.

[HIBERNATE]

Hibernate: a object/relational persistence and query service

http://www.hibernate.org/

[IBM]

IBM Websphere Portal

http://www.ibm.com/websphere/portal

[IBVRE]

Integrative Biology Virtual Research Environment project

http://www.vre.ox.ac.uk/ibvre/

[INTELLIJ]

IntelliJ IDEA Java IDE

http://www.jetbrains.com/idea/

[INTUTE]

http://www.intute.ac.uk/

[JAFER]

Java Access For Electronic Resources Project

http://www.jafer.org/

[JAFERPORTLET]

Available from the CREE deliverables page

http://www.hull.ac.uk/cree/deliverables/index.html

[JASIGCH]

JA-SIG Clearinghouse

http://www.ja-sig.org/wiki/display/JSG/Clearinghouse+Contributions

[JETSPEED1]

Apache Jetspeed 1

http://portals.apache.org/jetspeed-1/

[JSR168]

JSR 168: Portlet Specification

http://www.jcp.org/en/jsr/detail?id=168

[JSR286]

JSR 286: Portlet Specification 2.0

http://www.jcp.org/en/jsr/detail?id=286

[NETVIBES]

Netvibes Portal

http://www.netvibes.com/

[ORACLE]

Oracle Portal

http://portalcenter.oracle.com/

[OSSW]

OSS Watch. JISC's Open Source software advisory service

http://www.oss-watch.ac.uk/

[PERX]

Pilot Engineering Repository Xsearch

http://www.icbl.hw.ac.uk/perx/

[PORTLET]

A standards-based software component which provides user interaction via a markup fragment that can be integrated with other fragments within a portal interface.

http://en.wikipedia.org/wiki/Portlet

[PUBMED]

PubMed OAI Service

http://www.pubmedcentral.nih.gov/about/oai.html

[SAKAI]

Sakai: an online Collaboration and Learning Environment

http://sakaiproject.org/

[SFSITE]

Alerting Portlet at SourceForge.net

http://sourceforge.net/projects/alertsportlet/

[SPP]

Subject Portal Project Portlets site

http://www.spp.opensource.ac.uk/

[SPPARCHIVE]

For archives of previous Subject Portal Project work see:

http://www.spp.opensource.ac.uk/spp/Archive

[SPRING]

Spring Framework: a Java/JEE application framework

http://www.springframework.org/

[SUBV]

Subversion version control

http://subversion.tigris.org/

[TRAC]

Trac: a wiki and issue tracking system for software development projects

http://trac.edgewall.org/

[UPORTAL]

uPortal portal software

http://www.uportal.org/

[WSRP1]

Web Services for Remote Portlets 1.0

http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf

[WSRP2]

Web Services for Remote Portlets 2.0

http://docs.oasis-open.org/wsrp/v2/wsrp-2.0-spec-pr-01.html

[XTRA]

TechXtra Service

http://www.techxtra.ac.uk/