CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2007; **19**:1643–1652
Published online 7 June 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.1115

Using the Sakai collaborative toolkit in e-Research applications

Charles Severance[1], Joseph Hardin[1], Glenn Golden[1], Robert Crouchley[2], Adrian Fish[2], Tom Finholt[3], Beth Kirschner[3], Jim Eng[3] and Rob Allan[4,*,†]

[1] *Sakai Project, University of Michigan, Ann Arbor, MI, U.S.A.*
[2] *Centre for e-Science at Lancaster University, Lancaster, U.K.*
[3] *MGrid Center, University of Michigan, Ann Arbor, MI, U.S.A.*
[4] *CCLRC e-Science Centre, Daresbury Laboratory, Warrington, U.K.*

SUMMARY

The Sakai Project (http://www.sakaiproject.org) is developing a collaborative environment that provides capabilities that span teaching and learning as well as e-Research applications. By exploiting the significant requirements overlap in the collaboration space between these areas, the Sakai community can harness significant resources to develop an increasingly rich set of collaborative tools. While collaboration is a significant element of many e-Research projects, there are many other important elements including portals, data repositories, compute resources, special software, data sources, desktop applications, and content management/e-Publication. The successful e-Research projects will find ways to harness all of these elements to advance their science in the most effective manner. It is critical to realize that there is not a single software product that can meet the requirements for such a rich e-Research effort. Realizing that multiple elements must be integrated together for best effect leads us to focus on understanding the nature of integration and working together to improve the cross-application integration. This leads us not to drive towards a single toolkit (such as Sakai or Globus), but instead to a meta-toolkit containing well-integrated applications. When considering a technology for use, perhaps the most important aspect of that technology is how well it integrates with other technologies. Copyright © 2007 John Wiley & Sons, Ltd.

Received 9 January 2006; Revised 12 July 2006; Accepted 13 July 2006

KEY WORDS: e-Research; collaboration toolkit; Sakai; portal

*Correspondence to: Rob Allan, CCLRC e-Science Centre, Daresbury Laboratory, Daresbury, Warrington, WA4 4AD, U.K.
†E-mail: r.j.allan@dl.ac.uk

INTRODUCTION

Project teams trying to solve e-Research problems are often very similar to the blind men trying to describe the elephant. Depending on where the project's team first begins to attack their particular problem often leads the team to think that they understand the 'whole' e-Research problem space, based on their initial encounter or the type of technology first picked to solve the problem. Often, e-Research applications would find use of the following technologies.

- A Grid system such as the Globus Toolkit [1]. Globus provides mechanisms to harness distributed resources (mainly compute but also data movement) using global identity credentials that work across a multi-institution Grid.
- A data repository system such as the Storage Resource Broker (SRB) [2] or the Flexible Extensible Digital Object and Repository Architecture (Fedora) [3] allows the long-term storage and retrieval of data and metadata. This system can be used both for basic storage, retrieval, and archiving of data, but additionally is often used to support the publication or e-Publication activities of the field.
- A collaborative system such as Sakai [4] allows people to interact and work together as a distributed team. Groups can dynamically form on projects or sub-projects or for purposes requiring specific tools and authorization. All of the collaborative data in Sakai are maintained and can be archived to associate the collaborative activity along with the results of any compute or experimental data associated with a particular research effort.
- Portal systems such as GridSphere [5] or uPortal [6] are widely used. Portals which support standards such as JSR-168 [7] or WSRP [8] provide an excellent mechanism to bring together the user interfaces to tools from many disparate resources into a single 'portal', which makes them easy to find for the community.
- Knowledge building tools/software such as Data2Knowledge (D2K) [9] or Kepler [10] that allow scientific workflows to be produced which can be used to orchestrate scientific software.
- Large data sources such as the National Virtual Observatory (NVO) [11]. Often these data sources, such as astronomical telescopes, are very specialized pieces of equipment that need to be managed with appropriate security to gather data using advanced techniques. Once gathered, the data are made available to the e-Research efforts through some form of repository.

Many e-Research projects have relatively small staffs and short timelines that lead them to ignore the larger potential scope of their domain or even new multi-domain opportunities. For instance let us take the collaborative system above. e-Science projects usually adopt an *ad hoc* approach to provisioning themselves with core e-Collaboration tools/services, each component usually requires a separate logon, for example, for wiki, Intranet, and forum. Once a particular way of working or tool has been adopted, the co-workers become reluctant to replace these tools with something that would integrate with their domain specific tools/services, because, for example, a database would need converting to another format, an interface would need to be re-skinned or the team would have to learn about new tool. However, if these projects were to start by recognizing the existence of a bigger picture of open standards (which allows for integration/interoperability), then any new project would be able to pick up on earlier developments made on core tools/services. This would enable them to re-use the data and the staff to concentrate more of their effort on the actual scientific challenges to be solved. The collaborators should then be able to work more efficiently, for example, by cross searching the

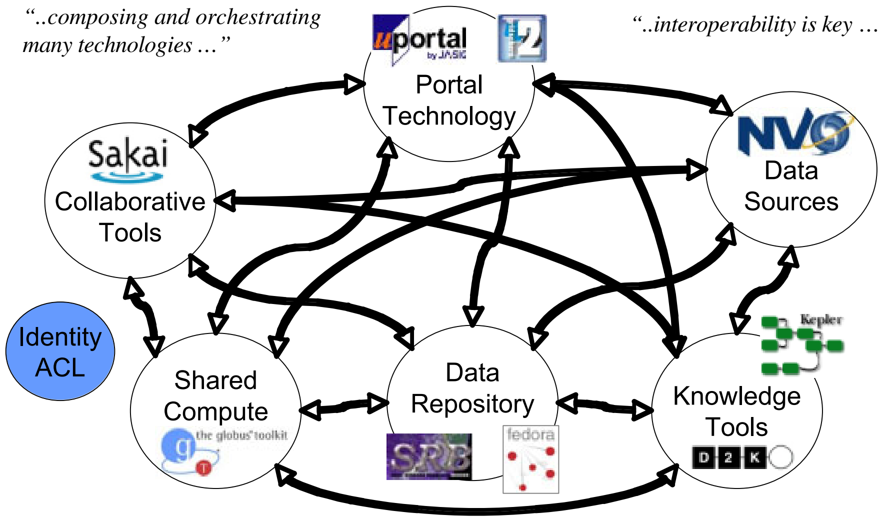

Figure 1. Cyber e-Infrastructure linking components.

various e-Collaboration tools and using them together. Although we concentrate on Sakai and Web portals in this paper, this argument can just as easily be applied to any of the technologies discussed above. Desktop office software suites provide some of this kind of capability, but need to be linked into scientific applications. By using the appropriate exchange technologies (JSR-168, WSRP, etc.) we can create a common cyber e-Infrastructure from portals that enables us to hide much of the underlying complexity (see Figure 1).

There are a number of important cross-cutting aspects such as global identity and global access control that are also an important part of any cross-application integration. The Globus Toolkit is commonly used to provide this cross-application identity and cross-institution security.

SAKAI OVERVIEW

The Sakai Project is a community source project developing a collaborative toolkit used both for teaching and learning and *ad hoc* collaboration. By focusing significant resources on building core collaborative capabilities, the Sakai project has provided a framework that is both useful ‘out of the box’, and also specializable to numerous domains by adding extra tool components (see Figure 2).

To solve an application in a particular domain, one takes the core elements provided by Sakai and then adds capabilities unique to the domain of the application.

Sakai core capabilities include (see Figure 3):

- Announcements;
- Chat Room;

DOI: 10.1002/cpe

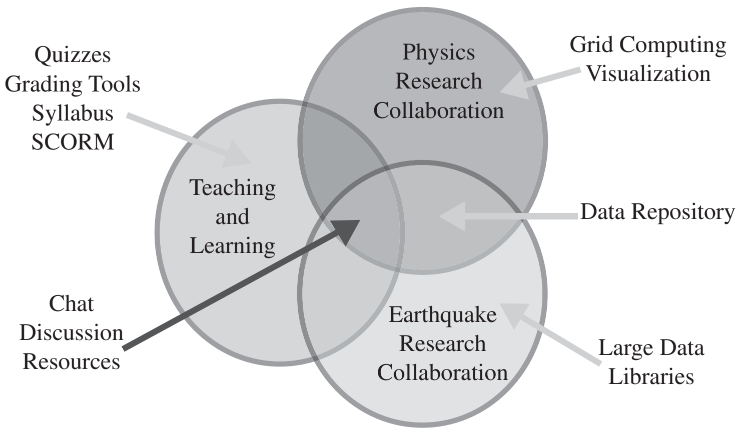

Figure 2. Commonality of e-Research domains.

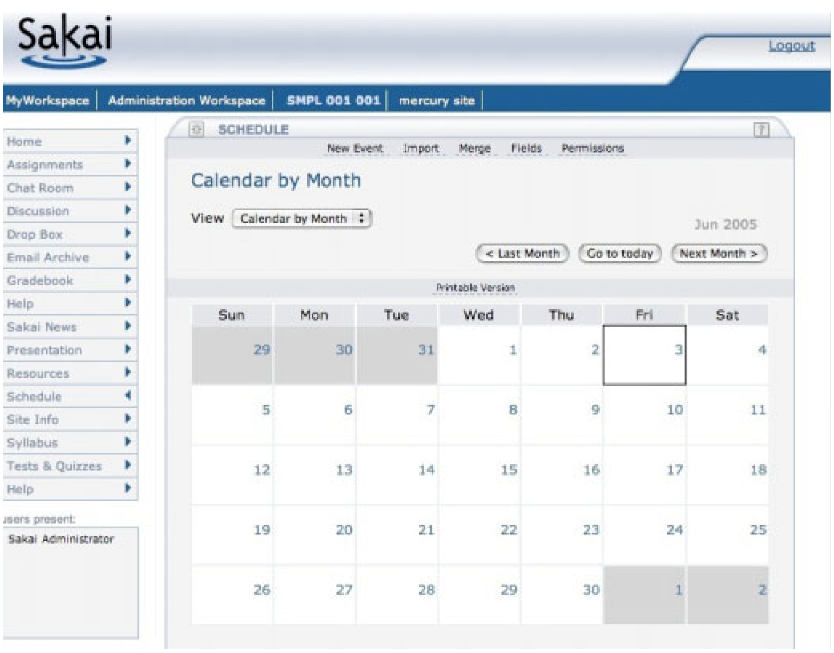

Figure 3. Screen view of the Sakai core schedule tool. Copyright © Sakai Foundation.

DOI: 10.1002/cpe

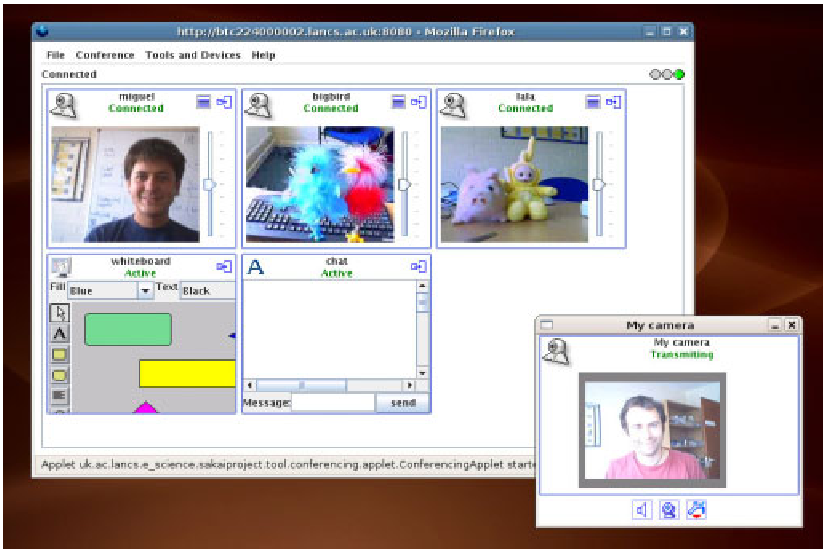

Figure 4. Screen view of the Agora conferencing tool. Copyright © Centre for e-Science, Lancaster University.

- Threaded Discussion;
- Drop Box;
- Email Archive;
- Message Of The Day;
- News/RSS;
- Preferences;
- Presentation Tool;
- Resources;
- Schedule;
- Web Content;
- Worksite Setup;
- WebDAV.

The Sakai community is actively developing new tools to extend the core collaborative toolset. These tools are not part of the Sakai 2.0 release but are under active development by researchers developing collaborative Sakai tools at Lancaster and Cambridge Universities in the U.K.

To build an e-Research collaborative environment, specialized tools are built based on the needs of the scientists and combined with the Sakai core tools to produce the collaborative environment. In the Sakai VRE Project [12], a tool was developed which could display a number of data and video channels simultaneously (see Figure 4) so that researchers could share data and discuss its significance. Components of this and related tools include:

- wiki based on Radeox;
- blog;
- shared display;
- shared whiteboard;
- multicast audio; and
- multicast video.

Sakai as a collaborative environment can be useful to an e-Research collaboration well before the first data repository or experimental data source is on-line. Some may argue that a Sakai site is the *first* part of an e-Research solution that should be deployed. Sakai is an excellent tool for planning the development and deployment of the remaining elements of the e-Research solution. As it is very simple to set up a rich collaborative environment using Sakai, installing a production instance of Sakai is a good 'first milestone' to be accomplished in the first few months of any e-Research effort.

SAKAI ARCHITECTURE

Sakai is designed to specifically support tools that need to make shared use of collaborative services. Sakai's architecture is more complex and more constraining than what can be achieved by developing a simple set of Java servlets. There are a number of reasons that Sakai needs its own framework.

- A Sakai installation must be dynamically configurable: with 20 or more tools running in Sakai, it is important that a problem in one tool is not allowed to affect the proper function of any other portion of the system. Any tool can be added or removed without harming the system.
- Even though Sakai is assembled from tools that are independently developed it must function smoothly with the natural feel of a single application. Sakai provides a style guide and a set of presentation widgets to help keep Sakai tools looking consistent.
- Each Sakai tool must be produced with markup that can be used in a number of presentation environments including both HTML display in a browser and display within a portal.
- Sakai demands that the code to support a capability (such as chat or discussion) be broken into a presentation component and a services component. The services component is responsible for the persistence of the data objects (chat messages, etc.). The presentation component and service component are to be cleanly separated across an API abstraction. Providing a clean API abstraction makes it possible to perform all Sakai functions using Web services in addition to the Sakai GUI.
- Sakai tools must be production ready and perform well at scale. There are production educational installations of Sakai that must support 3000 or more simultaneous users every day.

Whilst these requirements initially may seem onerous to application developers used to writing simple servlets, there are a number of services that Sakai provides to its tools through standard and published APIs.

- A rich set of administrative tools allowing the user to configure their environment.
- User identity and directory services with flexible plug-in mechanisms allowing easy integration of technologies such as Kerberos, LDAP, X.509, Globus, etc.

DOI: 10.1002/cpe

- A rich and flexible authorization system that supports roles and fine-grained access control that is easily used by Sakai tools.
- An event delivery mechanism that allows one tool to subscribe to an event channel and receive asynchronous notification when another user takes some action. There is support for the delivery of these events right out to the browser using XHtmlRequest (Ajax-style) technology.
- Support for operating in a clustered application server environment to support high-performance deployments.
- A set of standard APIs (50) to access framework and application services.

For a developer who truly wants to build a powerful collaborative application, Sakai is an ideal framework for the development and deployment of such tools. However, Sakai is not appropriate for the implementation of every tool. Owing to this, and because many e-Research projects are already using portlet technology but want the power of the collaborative tools that Sakai provides, it is very important that Sakai integrates closely with portals.

PORTAL ARCHITECTURE

Sakai is focused on people and groups collaborating. Portals are focused on assembling a number of relatively simple elements to create a single user interface or one stop shop for capabilities, in particular those involving data management and access to resources distributed across a Grid.

In the portal display shown in Figure 5, the tabs across the top are different major functions that are available to the portal. Sakai is incorporated into this portal as one of these tabs, possibly labeled ‘collaboration tools’. In a typical portal, all users see roughly the same information presented. This information is generally organized to best suit the needs of the science effort, organization providing the portal, or the resource that the portal represents.

Portals are constructed of software elements called portlets. The standards for portals are as follows.

- JSR-168 is a standard for developing Java portlets that run ‘within’ the portal and perform some function [7].
- WSRP is a standard allowing a portlet that is running on a system other than the portal system to be accessed and included in the portal by transferring the markup and user requests across Web services [8].

WSRP has two elements. The WSRP Consumer is the component that runs in the portal and requests HTML fragments from the remote portlet. The WSRP producer responds to the requests from the consumer and returns fragments to the consumer in compliance with the WSRP protocol. As long as the remote portlet and WSRP Producer meet the requirements of the WSRP protocol, they can be written in any language, for instance there is a Perl WSRP producer in the U.K. Go-Geo! Portal [13].

INTEGRATING SAKAI WITH PORTALS

Sakai is capable of functioning completely as a standalone application with its own internal rendering capability for HTML. In this section we describe the approach for integrating Sakai into an e-Research project-wide portal using WSRP and JSR-168.

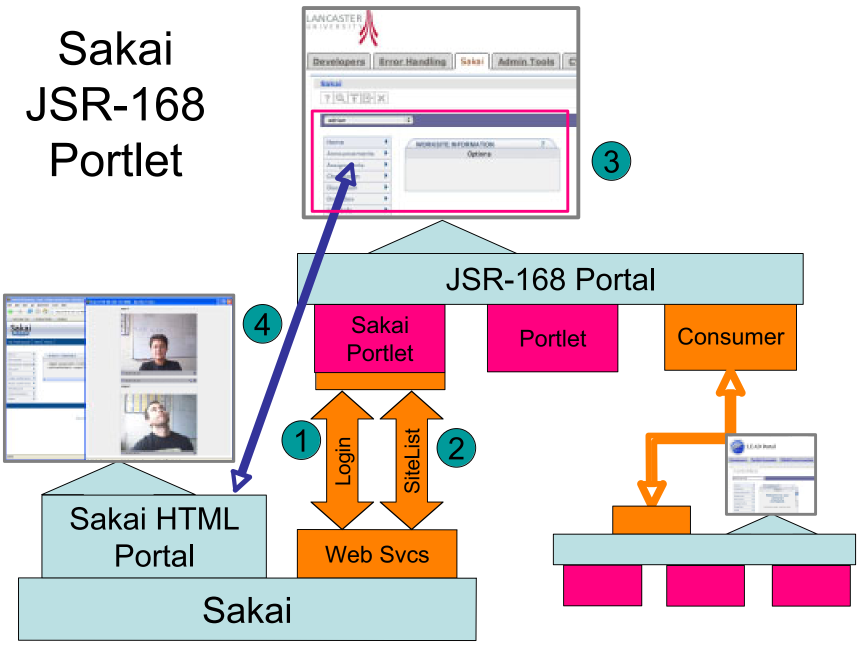

Figure 5. Portal architecture.

As Sakai is organized around people and the groups that each person belongs to, each user 'sees' a different view of the Sakai 'portal' based on their rights and permissions within Sakai. This is challenging for many portals. Portals often support 'personalization' where users choose or arrange from a pre-defined set of tabs and portlets, but do not support a situation where a single application working within the portal is allowed to dynamically reconfigure the overall portal interface content based on each user and their identity.

Currently the best way to integrate Sakai into a JSR-168 compliant portal is to use the Sakai JSR-168 portlet. This portlet uses the implementation approach of using Web services and providing a relatively thin presentation layer in the portal. All of Sakai appears in the portal at some location determined by the portal administrator, for instance the 'collaboration tools' tab mentioned above.

When the Sakai JSR-168 portlet starts, it puts up a login screen or auto-logs in depending on the portlet's configuration options. The login (1) is done using Sakai's Web services. Once the Sakai session is established, the Sakai site list for the current user is retrieved (2) and the portlet is displayed (3) giving the user a choice as to which Sakai site to select. As a site is selected, the portlet generates an appropriate iFrame with a URL to display the Sakai site (4). As the Sakai session was established using Web services, any Sakai login processing is completely bypassed, effectively allowing the portal login and authentication to act as the login and authentication for the Sakai instance (single sign-on capability).

DOI: 10.1002/cpe

If the Sakai instance and the portal instance agree, it is possible for the Sakai JSR-168 portlet to automatically create and populate user directory entries within Sakai.

The Sakai 2.1 release also includes a WSRP producer for Sakai applications. The problem with using Sakai's WSRP producer is that a separate producer–consumer pair must be established and configured for each 'tool placement' within Sakai. Since each user is presented with a completely different set of tool placements, it is difficult to replicate the entire Sakai structure in the portal using only WSRP unless some non-standard mechanism is used to communicate the structure for each user.

Like many of the other technology components in the e-Research solution, Sakai has a very intimate relationship with the project-wide e-Research portal. The other area that is very important to Sakai is the data repository.

SAKAI AND DATA REPOSITORIES

When an e-Research collaboration uses Sakai for its collaborative activity, there is a unique opportunity to capture that activity as part of the scientific record of the e-Research project. Unlike a set of *ad hoc* mailing lists or Web sites, all of Sakai collaborative activity is stored in a single place and associated with rich metadata. Every mail message, chat message, schedule entry, and uploaded file is tracked and tagged.

It is quite natural to export a Sakai site into an XML format for long-term storage in a repository. The ideal situation is when the e-Research activity has some type of unifying identifier so that various elements of an activity can be related in the repository. In life sciences (such as the Integrative Biology project [14]), this might be a life sciences ID (LSSID) that would be associated with a particular set of investigations of the research project. By marking all of the activities (experimental data, compute runs, collaborative activity, etc.) these can then be stored 'together' so that anyone looking back at the data will get the full picture of the data elements they are looking at.

It would be possible to not only see experimental data from sensors, such as in NEESGrid [15], but also to see the draft design documents, and the discussion around those design documents. Recorded multimedia or Access Grid sessions could also be stored and re-played later. The U.K. VRE projects Memetic [16] and IUGO [17] are developing software for recording and discovery/playback of Access Grid meetings and multi-media conference material, respectively. This rich information can be used to provide a much more complete picture of the data in its full context.

Sakai currently allows the manual export of sites and manual placement of the data in a long-term store such as SRB or Fedora. This manual process could be automated to automatically produce, export and store this information.

In the long term, Sakai is looking at using RDF [18] and OWL [19] to make the transfer of data between Sakai and data repositories more natural and dynamic.

CONCLUSION

To best serve the e-Research community an e-Research deployment team must look beyond a single technology and work to integrate a number of important technologies. It is more difficult to integrate technologies, but it allows the team to configure the best solution for each aspect of the e-Research project.

Increasingly, technology providers must understand that their tools are not standalone but must cooperate with other solutions as peers. Data exchange between large applications is an essential aspect of the use of an application. Strong support for Web services in each application helps significantly in allowing the project team to do the necessary integration.

Like many of the other elements of an e-Research solution, the Sakai collaborative toolkit must function as a standalone application. Sakai is also working to properly integrate with the other elements in the e-Research solution and, in particular, Sakai is working on integrating smoothly with JSR-168 portals and data repository systems as described in this paper.

REFERENCES

1. Globus. http://www.globus.org [11 December 2006].
2. SRB: Storage Resource Broker. http://www.npaci.edu/DICE/SRB [11 December 2006].
3. Fedora: Flexible Extensible Digital Object and Repository Architecture. http://www.fedora.info [11 December 2006].
4. Sakai Project. http://www.sakaiproject.org [11 December 2006].
5. GridSphere. http://www.gridsphere.org [11 December 2006].
6. uPortal. http://www.uportal.org [11 December 2006].
7. JSR-168. http://www.jcp.org/en/jsr/detail?id=168 [11 December 2006].
8. WSRP. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp [11 December 2006].
9. D2K: Data2Knowledge. http://www.d2k.org [11 December 2006].
10. Kepler. http://kepler-project.org [11 December 2006].
11. NVO: National Virtual Observatory. http://www.us-vo.org [11 December 2006].
12. Sakai VRE Project. http://www.grids.ac.uk/Sakai [11 December 2006].
13. Go-Geo! Portal. http://hds.essex.ac.uk/Go-Geo/ [11 December 2006].
14. Integrative Biology VRE Project. http://www.integrativebiology.ac.uk [11 December 2006].
15. NEESGrid Project. http://www.neesgrid.org [11 December 2006].
16. MEMETIC VRE Project. http://www.memetic-vre.net [11 December 2006].
17. IUGO VRE Project. http://iugo.ilrt.bris.ac.uk [11 December 2006].
18. RDF: Resource Description Framework. http://www.w3.org/RDF [11 December 2006].
19. OWL: Web Ontology Language. http://www.w3.org/2004/OWL [11 December 2006].
20. NGS Portal. http://portal.ngs.ac.uk [11 December 2006].

DOI: 10.1002/cpe