The Sakai Project Final Report to the Mellon Foundation

July 2006

A Final Project Report to the Andrew W. Mellon Foundation

By

University of Michigan

Primary Contact:

Joseph Hardin
Principal Investigator

2161 Duderstadt Center
2281 Bonisteel Blvd
Ann Arbor, MI 48109
734-763-3266
hardin@umich.edu

Executive Summary

The Sakai Project is an excellent example of leveraged resources, having grown over the last two years far beyond the original core institutions funded by Mellon. The project and the number of participants has grown dramatically, the initial code base has been extensively evolved, improved and expanded, the Sakai product is in production at a number of schools and is under pilot as a step to adoption at many more, and the Sakai Community has emerged as a significant contributor on the international stage of open source software for the global educational teaching, learning and research community.

The work of the original team funded by Mellon Foundation (and by the Hewlett Foundation for the Sakai Partners Program) has succeeded in producing production level software within the projected timeframe, is on a trajectory for adoption at educational institutions large and small around the world, and is in the process of providing a rapidly growing number of features for users and methods of interoperability for users and developers alike. In addition, this project has inspired and excited the larger academic community with a growing, developing vision of open source software in higher education, the means to create it, and a way to sustain it. This is the emergent model of Community Source. The Sakai Project is developing a roadmap for future academic software infrastructure development that others can follow, and some already have. In doing this, Sakai is pioneering a sustainable model for higher education's software future.

The original developers have not done this alone. They have been joined by a growing community of contributors to the Sakai Project. Nearly one hundred institutions of higher education (see Appendix B) and a dozen commercial organizations have become part of the Sakai Community and made healthy contributions. Even more have made contributions on the lists and participated in conferences. The organization of the project has evolved from an informal group of grantees to an active web-based community with active lists, numerous distributed projects, twice-yearly conferences, and a not for profit Sakai Foundation with an international board membership.

The first two years of this effort, the period covered by the Mellon Foundation support, have brought significant, impressive, successes and raised considerable challenges for the future. The Mellon Foundation can be justifiably proud of the contribution that it has made through support for the Sakai Project to the revolution underway in higher education around open source software, the collaboration of institutions of higher education in constructing tools to fit their needs, and the support of innovative tools for teaching, learning and research

The successes include the collaborative design, development and evolution of the forward-looking Sakai modular software framework, and the production implementation of Sakai software at the University of Michigan, Indiana University, Foothill College (and by extension 28 colleges in California), the University of South Africa, Yale,

Appalachian College (see sakaiproject.org/index.php?option=com_content&task=view&id=397&Itemid=312) and many more (see Appendix B) with tens, indeed hundreds of thousands of users relying on Sakai for their teaching, learning and research work. Sakai has established itself as the enterprise open source solution for collaboration and learning environments in a timeframe where most projects are still struggling with their early releases. This has been done while simultaneously continuing a high level of software development and delivery as the core group of developers expanded considerably, and channeling effectively the community energy that the partner's program has unleashed.

This software development effort has delivered working, production quality software on schedule, and will continue to do so. It is simultaneously building the community that will enhance and support that software over the long haul. As the report below shows, Sakai has moved in this short time from the delivery of tools to the creation of an academic framework. Sakai has become much more than just a course management system, and is becoming an increasingly inclusive and integrated environment for support of the full range of scholarly activities.

Sakai Contributions

Sakai has set the terms of the debate in the higher educational community concerning future institutional infrastructure software choices.

- The response at Educause and other conference experiences, articles, publications, and presentations universally attest to this. Sakai has been the focus of discussions of Higher Education's software future, and is increasingly seen as an example to be emulated. A recent example is where the CEO of Red Hat held up Sakai as an example of how to do things right in higher education: "A better alternative [for universities to follow], he said, might be to embrace efforts such as the Sakai Project." (http://www.crn.com/sections/breakingnews/breakingnews.jhtml?articleId=189601910) see also http://sakaiproject.org.

Sakai has helped to make possible the formation of other open and community source efforts in Higher Education

- The Kuali effort has tracked closely the Sakai experience: "Many of the decisions and insights from Sakai's approach to Community Source have transferred quite well to Kuali's work for admin systems. The Sakai Community's rapid coming together substantially bolstered confidence that this model could help higher education coordinate its resources for collective benefit." - Brad Wheeler
- The emerging Student Services System also sees Sakai has preparing the way for their efforts.

Sakai Project Board Development – Providing Community Source Project Leadership

- The Sakai Project started by building a working Board, bringing together the talents of the participating schools and organizations and providing leadership for this large-scale distributed software development and community building project
- The Board added new members to the Sakai Board from a successful Partners Program, in the process charting the evolution of the project beyond the first 2 years and through the transition to a community-governed organization
- The Sakai Foundation became the Sakai formal structure, the existing Board members Founders and then its Board, and the first round of annual elections successfully held within the Sakai Community.
- The development of the Sakai Board has paralleled the development of the Sakai Community, moving from the grant-centric early phases of the project to the community-centric state we desire, in the process steering a large collaborative effort. This can be a model for other open source projects that are grant initiated.

Sakai Development of IP Management within Higher Education

- The Board has spearheaded the education of higher education leaders and staff around the issues and value of open source, and helped pave the way for subsequent adoption of open source software in higher education.
- The Sakai Community continues to be at the forefront of discussions concerning how the academy can best handle software related IP.

SPP Development – Building a Sustainable Future

- Established the Sakai Educational Partners Program (now called the Sakai Partners Program) in a matter of months ex nihilo.
- Successfully held SPP conferences every 6 months as attendance grew from 160 to 580.
- The conferences have been extremely successful, impressing many who come with the interest and commitment of the community members and are well on their way to developing the commitment, sense of community and organizationally sustainable structures and processes that will provide the future development and support needed to keep the Sakai Project successful well beyond the term of the current grant.
- The Discussion Groups and Working Groups in the SPP are taking on increasing responsibility for mapping the future and contributing to its development, especially in the areas of Libraries and Repositories, Content Management and Authoring, Migration, and Strategy and Advocacy
- Open forums - sakai-dev and sakai-user – have been set up and are open to the world, and very active

SCA and IMS Group establishment – Encouraging Commercial Options

- Established the Sakai Commercial Associates program and has commercial partners actively developing, and offering, support for the Sakai platform for schools that wish such support
- Established an IMS Working Group on tool portability to tackle head-on the issues of tool portability across Collaboration and Learning Environments, with participation from university, open source and commercial sectors

- Established another IMS WG on CC – Common Cartridge specs for sharing content from publishers

Cooperation with Exploding Open Source Community in HE – Building this community

- The Sakai Project has developed strong cooperation with related open source software efforts, like OSPI, and national and international open source efforts like the ETUDES Project at Foothill Community College, and JISC in the United Kingdom
- Sakai is a central part of the emerging consensus in the advanced collaborative research software community – including work with OGCE and discussions with European eScience groups like GridSphere

Developing and delivering production software

- The Sakai Project has developed the Sakai software that is now in production at the University of Michigan, Indiana University, University of California – Berkeley, in the 28 ETUDES Project schools, and in colleges and universities around the world (see Appendix B for listing).
- The evolving technical teams in the project have developed in-depth, forward-looking architecture and tool design processes and patterns that are supporting the distributed development necessary to meet the goals of portability, interoperability, support for innovation and user configurability that informed the creation of the project.

Rapidly Evolving the Sakai SOA

Later sections describe the evolution of the Sakai architecture over the lifetime of the project. At the highest level, Sakai started out with the CHEF software from University of Michigan which was a collaboration system with a modular architecure based on Java interfaces and a relatively simple framework based on the Jetspeed portal. This system was used to jump-start the project. This was the ground functionality that allowed us to get started quickly and meet our early production requirements. This system also had technical limitations in terms of the technologies used within CHEF and their performance characteristics.

The Sakai project has rapidly re-thought, re-architected, re-factored, and re-written nearly every single aspect of the original CHEF software to produce the Sakai 2.0 architecture and framework. The current code base of the Sakai project (800K lines of code) is far larger than the original CHEF code base (200K lines of code). The entire framework, for instance, has been re-written from scratch to produce the 2.0 framework. The 2.0 framework added ubiquitous web services to Sakai which allows Sakai to be integrated into an Enterprise using a Service Oriented Architecture approach. This rapid evolution has reached a point in the Sakai 2.2. release of July 2006, where the Sakai Architecture, framework and code base are all at a high level of sophistication.

Sakai 2.2 is an excellent example of a well-factored code base that allows modularity and flexibility in assembling a Sakai system, using either a very few or a full compliment of Sakai services.

As one school recently posted:
“One challenge we have faced is how to re-develop some of our surprisingly successful ideas that we had developed as "research" tools while continuing to do more research. Developing on

Sakai's stable framework is our approach to institutionalize successful ideas while building prototypes of new concepts.

In our case, we hope to realize IT value by using Sakai as a means to promote innovation and develop new ideas that advance teaching practice and change attitudes and behaviors toward technology use in the classroom."
Sean Keesler, Project Manager, The Living SchoolBook, Syracuse University

This example points up the dual value of a system that is capable of scaling to the enterprise level, supporting thousands of simultaneous users, and yet is also capable of incorporating the innovations necessary for it to be considered a useful tool at leading edge institutions. These have both been fundamental principles of the Sakai Project: to support enterprise-level services, and to be able to rapidly incorporate innovation.

Going forward Sakai has published a Roadmap to Data Interoperability which will allow every aspect of Sakai to participate in an enterprise which uses a Service Oriented Architecture.

The Sakai Phenomenon

The past two years have been really been phenomenal for the Sakai Project. From a notion in the fall of 2003 to a Mellon-funded collaboration of the initial partners in the beginning of 2004, to production quality software in 2005, Sakai has quickly become a vision and program that have caught the imagination of the higher education community, in the U.S. and worldwide. The project quickly brought on over 50 committed partners in its first year, and has grown at a steady rate to nearly 100. Early on, in a Syllabus Campus Technology article slightly over 50% of those schools asked said they saw Sakai in their own institution's future, and over 90% said Sakai would affect the market in higher education software positively (http://www.campus-technology.com/article.asp?id=10308). Sakai has been fortunate to be one of those rare events – a market changing force that has grown from the bottom up.

Internationally, universities in England, Australia, Japan, Holland, Sweden, Germany, and Denmark, South Africa have joined the SEPP (now SPP) program, and regional conferences are growing up like the one for Europe in Luebeck this fall (see http://www.oncampus.de/index.php?id=730 or Appendix E). Regional organizations to support Sakai use and development are also emerging, like Sakai Quebec, and the South African Sakai Association (SASA).

At recent Educause Conferences and others in the higher education community, such as the League for Innovation in the Community College conferences, Sakai has seemed to be everywhere. Sessions devoted to Sakai, the software and the project, have often filled to over-flowing. Sakai has had meeting rooms and suites at EduCause and given continuing presentations on its software plans and processes, and answered questions of potential partners, adopters, and development collaborators. Interest in the development progress and adoption experience of Sakai has been consistent and deep. Sakai has appeared at numerous booths in these conferences, including those of Sun, Apple, IBM,

rSmart, Unicon, and others on exhibition floors, and has been invited back again and again. Sessions discussing open source in higher education regularly reference Sakai and its ideas, as do papers concerned with the future of educational software in general. Various meetings of university CIOs have been organized and are heavily attended, spending much of their time discussing the import of Sakai's successful community source project. It seems that the open source conversation in higher education now necessarily includes the Sakai Project as a formative example

Indeed, one of the major contributions of the Sakai Project over the past two years has been to raise the level of the conversation about open source solutions in higher education, to the point where any discussion about software acquisition or development is now incomplete without consideration of existing open source alternatives and the prospect of community source development efforts. Articles in the Chronicle of Higher Education have focused on the Sakai Project and its importance as a path finding experiment in the higher education community (http://chronicle.com/prm/daily/2004/01/2004012204n.htm). Presentations at Educause about the advantages of "Buy, Build or Borrow" (sakaiproject.org/presentations/Educause2004/ChoosingOpenSource.ppt), updates of the Sakai Community in publications such as Campus Technology about the emerging methods of open and community source development (http://www.campus-technology.com/article.asp?id=9940) have accelerated this conversation. See the Sakai Project's, incomplete, list of articles at http://sakaiproject.org under News.

In related meetings, the Educause Summit on Open Source Software hosted by the Mellon Foundation brought together 29 higher education IT leaders to discuss open source and raise its profile within Educause, and Sakai was both well represented and a central example in the discussions.

The "Open Source - Opens Learning: Exploring the Impact of Open Source in Education" conference, organized in part by a Sakai Commercial Affiliate, brought together leaders in the higher education community to discuss just these issues, with Sakai Board members prominent participants (http://www.opensourcesummit.org/presenters.html). The value of the Sakai label is becoming apparent in these forums as it is repeatedly referred to and held up as an exemplar of the emerging model of community source. Sakai has rapidly become a brand.

An obvious benefit of the Sakai Project's recognition is that it is acting as an attractor for contributions and participation by others

Important examples of this include the contribution of tools like the rWiki from Cambridge University, the integration of the OSP tools into Sakai (see Appendix D), the SakaiBrary work, and the recent contribution by IBM of SCORM tools:

SCORM-Compliant Code to Accelerate Open-Source Movement in Higher Education

ARMONK, NY -- (MARKET WIRE) -- July 11, 2006 -- IBM today announced it has donated e-learning software code to the Sakai Project, a group of

learning institutions creating and deploying open-source course management, collaboration and online research support tools for higher education. The donation is part of IBM's overall support of open computing in education. Last year, the company announced that it will allow royalty-free access to its patent portfolio for the development and implementation of selected open software standards for Web services, electronic forms and open document formats in education.

To see the full article:
http://www.marketwire.com/mw/release_html_b1?release_id=143063

For many projects within the higher education community who wish to find wider distribution and collaborators in development for their own innovative educational software, Sakai has provided a route to sharing with the wider community. This is true for the SPP and SCA partners and for anyone who has a need for the functionality of the Collaboration and Learning Environment tools it offers as well as the forward-looking architecture that the project is developing, and who see value in leveraging this for their user communities. The Open Source Portfolio Initiative is leveraging their development with Sakai through their commercial partner developer, utilizing features of the Sakai framework and tool set that they would otherwise have had to develop themselves. And the complete functionality of the ePortfolio software that OSPI is developing will, of course, appear in the Sakai system as another set of tools, increasingly integrated with all the other tools in Sakai (see Appendix D). This is an example of the coordination of software development efforts, and realizing the synergies that can result, that was part of the original Sakai vision.

At the SPP conference tech demonstrations, we are seeing literally dozens of similar software projects, eg, from partners like the University of Wisconsin, and the Carnegie Foundation for the Advancement of Teaching, University of Missouri and Syracuse, that are reorienting their development around the Sakai software core, making sure they will become a part of the Sakai extended family of software. The NSF supported Open Grid Computing Environment is also on a design and development track that is leading them to increasing adoption of the Sakai framework and method of portlet construction, with early grid portal elements already demonstrated in the Sakai environment. Collaboration with JISC in England is leading to similar results with JISC grant recipients working on adoption of Sakai in pilot projects at their institutions, and with Sakai partners on grid, Samigo and the Twin Peaks projects. Commercial partners have packaged and released supported versions of the Sakai software. Sakai's gravitational field is expanding rapidly.

California Community Colleges Move to Sakai

Sakai has also been fortunate in seeing its software adopted by statewide initiatives.

The California Community Colleges Chancellor's Office (CCCCO) ended support for central hosting for WebCT and BlackBoard on June 30, 2007, and has officially endorsed the California Community Colleges Consortium (CCCC) ETUDES Project and

open source development and implementation effort, managed by Vivie Sinou, Dean of Learning Technology & Innovations at Foothill College. The System Office has sent out a formal communication to all colleges regarding this change, but we are also going to be doing some outreach to make sure colleges have enough lead time to make a decision as to their course management system. With the merger between Bb and WebCT, the System Office for CCC's was simply unable to generate enough of a consortium pricing reduction to justify continuing with centralized hosting.

Last month, the CCCCO and the ETUDES Project signed a multi-year development contract ($1.25M over the next 3 years) to fund new and continued development for ETUDES-NG which is based on the Sakai Collaboration and Learning Environment (CLE). ETUDES-NG is in use by over 30 California Community Colleges to deliver and support instruction, and is well supported through Vivie Sinou's team at Foothill. The technology services contract and agreement with the System Office will fund salaries for project management, coordination, and new development, primarily for the Melete Lesson Builder, as identified and prioritized by the participating California Community College members, and conversion utilities for BB and WebCT. Effective July 1, 2006, the CCCC ETUDES Project becomes a state project that is funded by a combination of sources, rather than mainly Foothill: Consortium members, CCCCO, and grants (Hewlett). The project continues to be housed at the founding institution, Foothill. Vivie Sinou will be reporting to Catherine McKenzie at the CCCCO.

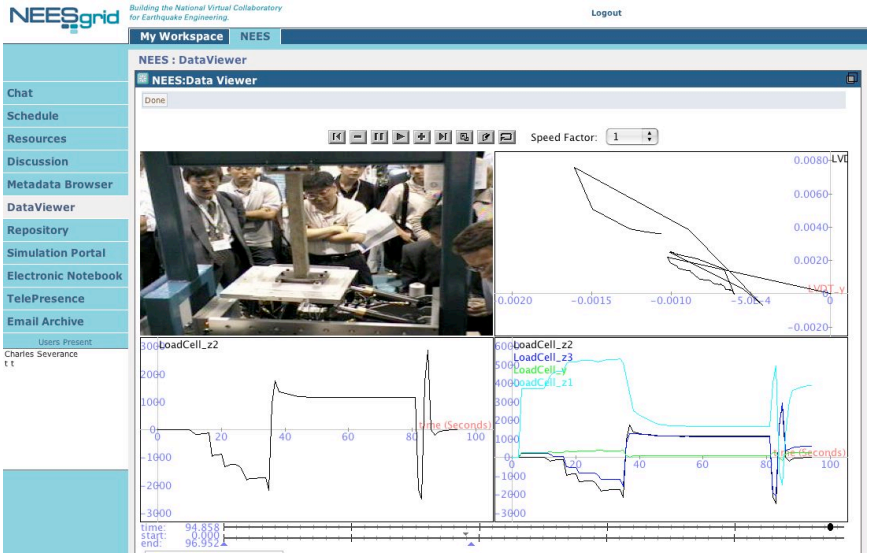
Twenty-four colleges will be offering classes in the Sakai-based ETUDES-NG system this fall, and all 30 colleges will be transitioned over by June 2007, at which time, the legacy ETUDES Classic system will be retired.

Sakai as more than a CMS – an Emerging Academic Framework

Sakai has also been able to effectively extend the conversation beyond the traditional model of course 'management' systems, CMSs, and into the arena of support for online research and group work at institutions of higher education, the arena of an academic framework. It has always been a fundamental precept of the Sakai Project that faculty should be supported in all their scholarly activities, including research with distributed colleagues, and that this research should be able to be easily brought into the teaching and learning environment. The development of "eScience" software in the United States, Europe and Japan has been progressing rapidly, with increasingly active and well-funded grid software projects (see Appendix G).

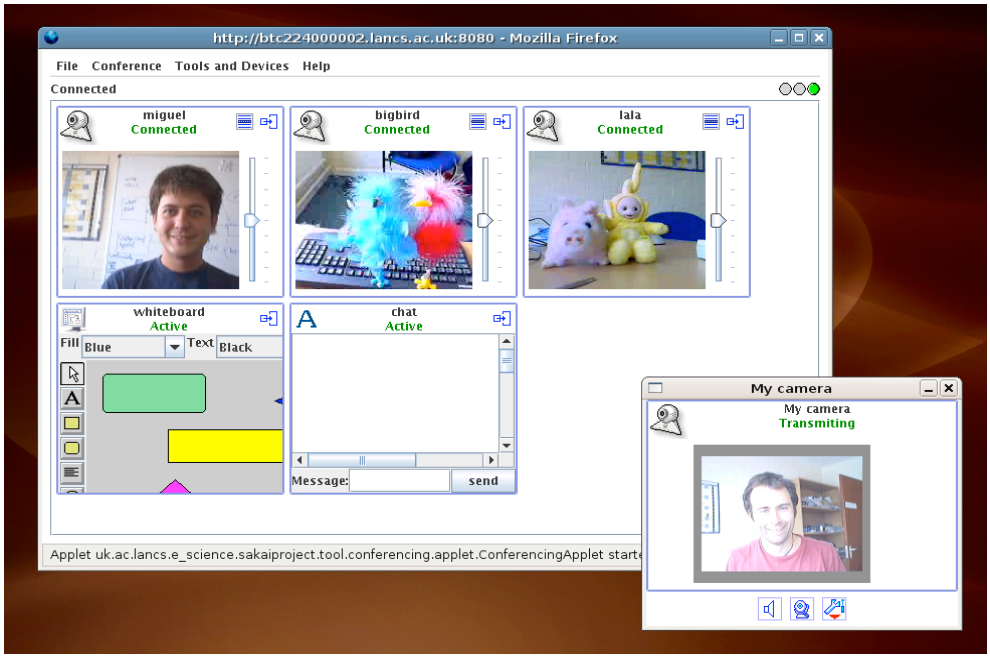
The Sakai Project is participating in such efforts in the US directly through the Open Grid Computing Environment (www.ogce.org) work and also the National Earthquake Engineering Simulation grid (NEESgrid) effort funded by the NSF. The NEESgrid project used early versions of Sakai to support its extended community of researchers and research facilities, and is collaborating with Sakai developers in the next phase of its development.

The NEESGrid project made use of the general purpose collaborative capabilities available in Sakai, augmented by adding domain specific research oriented tools to Sakai specific to data capture and analysis.



Follow-on projects after NEES in medical informatics and AIDS research are likewise incorporating Sakai into their research communities.

Lancaster University in the UK is leading a project to use Sakai to enhance collaboration between researchers in the Humanities and Social Sciences field. Their requirements analysis has led them to begin work on a set of synchronous tools that will augment the Sakai capabilities for both the teaching and learning as well as the collaborative applications of Sakai.



These tools are expected to be in limited production in Fall of 2006 at Lancaster University and then released to the entire Sakai community some time in 2007. A number of JISC-funded projects in the UK have explored the use of Sakai in virtual research environments. A list of those projects is in Appendix G.

The Open Grid Computing Environment (www.ogce.org) is an NSF-funded effort which has released a toolkit for setting up scientific collaboration portals, and includes Sakai, uPortal, and GridSphere support in the OGCE toolkit.

The release of Sakai 2.2 continues to progressively empower these efforts, allowing any user to set up an online worksite that she can then invite others to join, and that can be enhanced in increasingly easy and rich ways by users and developers. For faculty who are working with local or remote colleagues, this makes it easy for them to mobilize a full, preconfigured suite of collaborative tools to support their distributed research, and then to use the same tool to bring portions of that work into their teaching. Likewise, for students, or staff for that matter, it becomes easy to set up fully supported collaboration sites with class members or work team members to pursue their work. Some sites here and abroad are implementing Sakai as an online group work tool first, and only later bringing it into the classroom. Taking advantage of both of these routes to adoption, Sakai is rapidly finding its way into higher education infrastructure.

Organizational Development

Throughout all this activity, the Sakai Project has had to experiment with a number of different methods for management of the various tasks and groups involved in the project. The Sakai Project is best described as an experiment in constructing a community to support distributed software development, starting with the relatively tightly coupled core group, extending out to the full SEPP community, and then beyond as designers and developers in the wider open source community participate. As we move across these groups, and even within them, different styles of management and development are encountered, and one task of the project has been to balance these different styles to achieve its goals. In every case, the Board has found that a bottom up approach, relying on the skills of the professional staff involved, has yielded the best results.

For example, the more tightly managed architecture design activity has been matched by a more distributed but still tightly coordinated requirements gathering and user interface design effort, the structure of which was designed by the staff involved. This has resulted in the early production of a Sakai Style Guide as well as the first versions of the code and the Tool Portability Profile.

Evolving Methods to meet Project and Community Demands

The process of releasing code is a fundamental one in projects that are developing software, and especially in open source projects. Traditionally, there has been a release date largely set by the management team in consultation with the development staff, and

code has been built, tested and released. The experience of the first year of the Sakai Project brought with it a new release methodology. In the past, a fully packaged code release was thought to naturally precede its production implementation. Experience showed, however, that this sequence would only slow down both the implementations in production environments, and the, not unrelated, release of well-tested, stable code. Missing our production milestones was not acceptable at the individual participating schools, since this would have meant slipping transition plans and incurring significant costs in maintaining legacy systems through an extended transition period. So Sakai moved to what it called a "production driven" release schedule, where the needs for meeting local production deadlines were paramount, and the full release of the code, which had been hardened in its production use, followed the installation of the most recently developed code in production. This was done with the release of the Sakai 1.0 Final after it had been running under heavy load at Michigan for a full month, and was the case with the Sakai 1.5 Final, after it had been running in production at Indiana for a similar period.

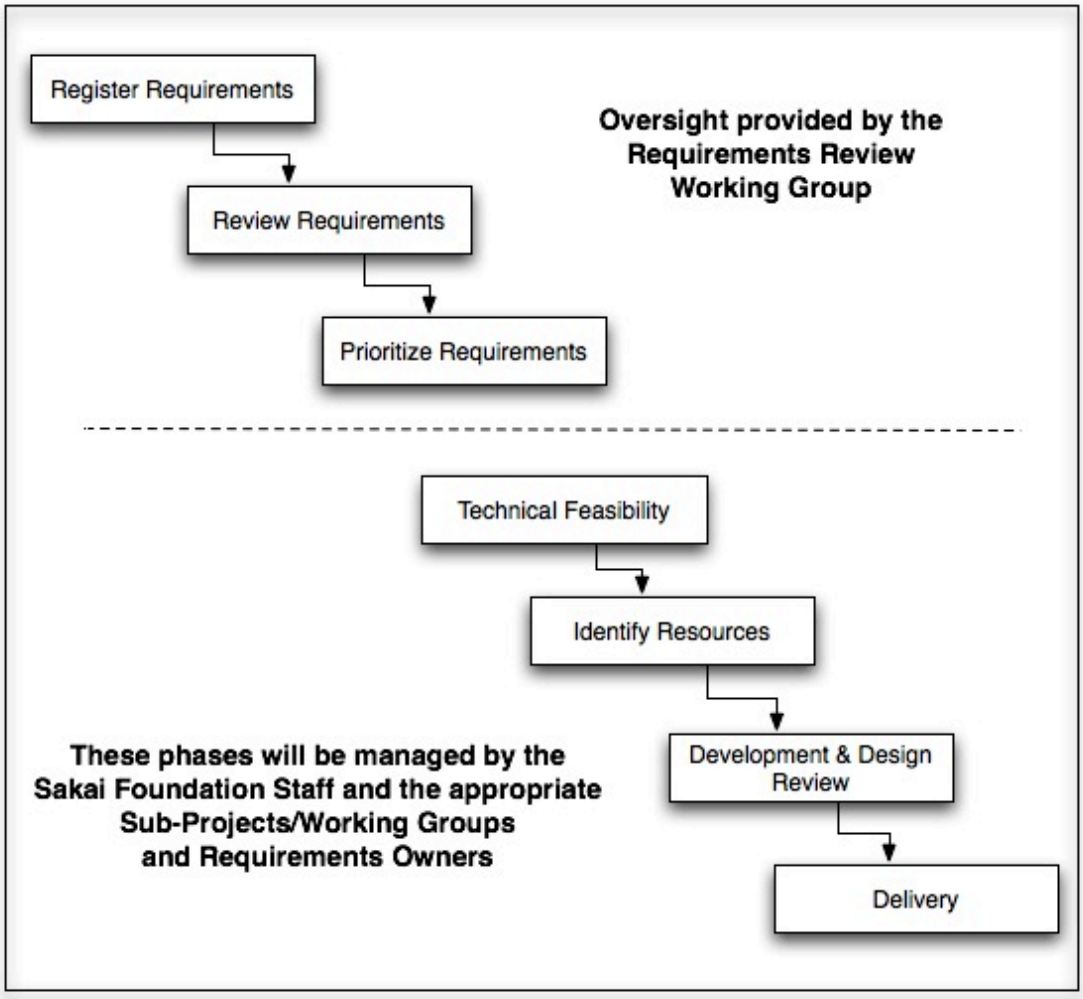
This signaled a significant change, and the pressure placed on the whole project by the demands of early implementation. It temporarily helped the project solve one of the persistent problems in collaborative software development, that of local imperatives swamping collective needs, splintering of the software efforts and leading to uncoordinated development among the project members. With this form of local production driven implementation with a following release, those formerly conflicting demands were brought more into alignment. The local imperatives served to force the rapid development of the next generation of the software, as each campus tried to meet its own production deadlines, and the time and resource consuming effort of putting out a new, fully packaged release was done after those local imperatives had been met as best they could. In addition it resulted in a tested and hardened set of code that others can install with increased confidence.

This has evolved into a more mature model, where the local imperatives are still drivers of much of the development that will be in the next release, but now, as the number of local systems relying on the Sakai release has increased, a more community-wide process of requirement requests, code contribution inclusion, staged framework enhancements, and distributed testing has been incorporated into the Sakai code release process. Here are some examples.

The Requirements Process – Scaling Up

Mara Hancock of UCal-Berkeley has helped lead the development of a community process of requirements gathering over the last year, helping to make the transition from a project to a more broadly based community a reality. When the Sakai project was only a few core schools, running to get a first version of the system up, the coordination mechanisms were more focused on gathering requirements and doing gap analyses of the core institutions' needs. Now, with scores of contributors to the requirements of the system, and with widely distributed development by teams around the world to meet those requirements, a method had to be developed to gather, filter, rank, and

communicate the requirements to the community as a whole. The Requirements Review WG has arisen to do just this, developing the tools and practices that are currently in use by the Sakai community. This is a good example of the development of new practices by the community as it evolves and takes advantage of the power of open source methods. Traditional models of command and control would be counter-productive in the open source environment for such tasks, and community practices that emphasis communication and coordination of distributed self-delegation of work have to be developed and made part of the community's culture. For detail on this evolving work, see: http://bugs.sakaiproject.org/confluence/display/REQ/Home



The Distributed QA Regime in Sakai – Example of leveraging community resources through partial position funding

As with the Requirements example above, methods for doing quality assurance and testing the Sakai software in the development and release cycle had to be developed. This feature and load testing is a key component of delivering a solid release every time the software revs, and requires tight oversight and the construction and encouragement of large set of community volunteers to do the work. Sakai can not afford here, as in many

other places, to hire full-time staff to perform these functions. The long term viability of the Sakai effort requires a way of harnessing community resources and managing them. One solution has been to do what has been done with QA, to partially fund a staff member at one of the participating schools and have them assigned to developing the community practices necessary, coordinating the activities with other centers of effort, such as the development and release teams, and communicating with the wider community the results. Megan May at Indiana University is the staff the Sakai Foundation is fractionally supporting here. She recently sent out a note thanking and congratulating those who participated in the 2.2 testing. They included 82 people from 28 institutions, in 7 countries.

Sakai Fellows – Recognizing Excellence in Community Contributions

An example of the community building methods that have evolved over the first two years of the grant, and that have been been developed and adopted by the Sakai Community, is the Sakai Fellows Program. Overseen by a community work group that includes board members, a call for nominations is put out for those who have made strong contributions to Sakai. The work group then chooses those deemed best from the nominations.

As this newsletter release points out, these people are well known within the community, and this recognition helps them back home, as well as adding to their earned referent authority within the community.

Sakai Newsletter, June, 2006

The Sakai Foundation is pleased to announce the 2006 Sakai Fellows. The Sakai Fellows program seeks to foster community leadership and contribution through recognizing and supporting active contributors. The Sakai Fellows were selected from an outstanding set of nominations of contributors to the Sakai community.

The Sakai Fellows Selected for 2006:

Johan van der Berg, UNISA – University of South Africa
Ian Boston, Cambridge University
Steven Githens, Northwestern University
Aaron Zeckoski, Virginia Tech
John Ellis, rSmart Group
Zach Thomas, Texas State
Seth Theriault, Columbia University
Clay Fenlason, Boston University

None of these names come as a surprise to anyone who is involved in the Sakai project – these folks are the ones who we all count on when something needs to be done quickly and well.

The way Fellows work is that 5% of the membership budget is dedicated to the Fellows program – so the more members -- the more fellows we have. Each fellow serves a 2-year term, and is awarded $5000 per year to be spent on things like travel, laptops, etc.

The Sakai Conferences – Community Building

The Sakai Conferences are becoming central events in the development of the community, coordinating its activities, and building its membership by infecting newcomers with the excitement of the Sakai effort. Participants consistently report that the conferences have become key community building events, solidifying relationships across projects and organizations and introducing recent adoptees to the contents and culture of the Sakai Community.

The most recent Sakai Conference with the Open Source Portfolio met in Vancouver, May 30 to June 2. The conference was attended by 487 participants from 192 colleges universities, government agencies and businesses. Twenty-five countries were represented. The conference included 119 sessions and group meetings covering diverse topics including how to use Sakai, development, sustaining the Sakai community, pedagogy and learning technology, and information technology and architecture. The conference included 24 tech demos and 21 officially scheduled birds-of-a-feather (BOF) meetings, in addition to many spontaneous BOFs. Mozilla's Mitchell Baker gave the keynote address.

The December, 2006 conference will be held in Atlanta, Georgia, and the June, 2007 conference in Amsterdam, the Netherlands.

Comment from Conference Participants

It was wonderful to put faces to the names I interact with on a daily basis. I found the exchange of ideas was overwhelming/exciting -- there is so much everyone would like to do with Sakai! I look forward to reflecting on the progress that has been made in Atlanta.

- Megan May, Indiana University

It was really encouraging to see the number and breadth of the presentations. I was mainly interested in library and repository issues surrounding Sakai, and there were a large number of relevant sessions. In retrospect, maybe there were too many -- I was able to attend sessions almost exclusively on these topics, and so may have missed opportunities to branch out during "off" times. Another striking aspect of this conference was the broadening of the number of institutions contributing really meaningful work. It indicates to me that the community as a whole is moving in the right direction.

- Jim Martino, Johns Hopkins University

I liked the longer breaks between sessions, which gave me plenty of time to talk to the presenter if I needed to, get a drink, answer email, or have a decent conversation. I thought the bootcamp was a good learning experience and we got a lot of good feedback from the attendees. I feel like the conference is a natural place for developers to get together and learn from each other.

- Aaron Zeckoski, Virginia Tech

At each conference since the first one in Denver, I've seen the Sakai community grow more mature. In Vancouver I think we transitioned from a majority interested in understanding what "they" were doing, to a majority that know we *are* they. I saw lots of people with great ideas holding themselves accountable. Sakai's greatest opportunity (and challenge) is to maximize the potential of so many eager contributors, and I think we're headed in the right direction..

- Chris Coppola, The rSmart Group

The atmosphere at the Vancouver conference was different from previous conferences. More attendees were well-informed about Sakai at this conference, so in addition to answering questions about Sakai, we were able to ask some of our own questions and receive answers from non-core institutions. This is exciting and one of the main reasons why we embarked on the Sakai journey in the first place. It goes without saying that Karaoke night was the best community bonding experience ever! I look forward to future Sakai conferences with a growing community.

- Kristol Hancock, Indiana University

The conference in Vancouver was the fifth Sakai/SEPP conference I've attended, and it was deeply gratifying to see how the Sakai community has matured over the past three years. It is a testimony to both our leaders and participants that we are moving smoothly from a grant-based project to a true collaborative community. Everywhere were signs of vitality, new ideas, and commitment to the vision of Sakai that we share. Recognition of the new Sakai Fellows is a very positive way to recognize new and established voices in the community. The commercial affiliates are also coming into their own and demonstrated clearly their support for the community. Like all of the previous conferences, Vancouver was a great place to meet old and new friends, to network with people, and explore projects. Overall, the quality of the presentations and sessions was good and I look forward to continuous improvement in future conferences.

- Mark Norton, MIT

Vancouver was the 2nd Sakai Conference that I attended. It was great to meet some fellow Europeans again and to meet some new colleagues from the worldwide community that Sakai is becoming more and more. I was very impressed by the presentations Unisa gave at the conference. Furthermore, I was pleased to see Chuck Severance being appointed Executive Director. I hope that we can make the first European Sakai Days in Luebeck (Sept 6 & 7) as much a success as the “original” Sakai

conference has become. So… see you next time in Luebeck or in Atlanta!

- Wytze Koopal, University of Twente

Despite the confused look I wore on my face at the conference, I was having a great time....I always look like that. It was very inspirational. Before the event, I was not sure why I was attending really, since I knew that it was taking me away from a checklist. It was not any particular event that inspired me, although Mitchell's talk was so very pertinent; rather it was the meta-environment.... it was just-be-nice time peppered with a lot of smart talk.... I love it.

- Jon Gorrono, UC Davis

For me, the conference at Vancouver revealed two critical characteristics about the Sakai Project in 2006:

Enthusiasm continues to build for Sakai not only in the developer community, but in the broader communities of faculty and their support staff. Many if not most of those looking at Sakai today are thinking of it as much more than a course management system, but as an enterprise framework based on solid, open standards with a loosely coupled design that allows for integration and promotes innovation. The technology demos on Thursday evening confirmed the dynamic nature of the project and justify this enthusiasm.

The Sakai community is also maturing and becoming techno-realistic about Sakai, which is a good and inevitable step in the evolution of any new technology. Creating robust and feature-rich software is challenging. I am very much encouraged by the realistic outlook shared among conference attendees that Sakai not only has value today but justifies our enthusiasm for the future. I overheard a new member of the community say that "Sakai will be the platform of choice over the long haul."

- Scott Siddall, The Longsight Group

Post-Grant Phase Sustainability Plans – SPP – The Sakai Foundation

As we end this Mellon Grant Phase and go beyond the second year it's good to reflect on the models we presented initially for community building and sustainability and see how well they've served us, and how far we've come. The period of the Mellon grant for the Sakai Project was two years. During this period the Sakai Project Board was responsible for the development and release of the Sakai 1.0 and 2.0 software. The focus of the Sakai Project Core team members necessarily remained on the development and delivery of these software components. For the longer term, however, the Sakai Board has, from the beginning, recognized two conditions that are essential to the success of the Sakai open source effort:

1) Creating a community of adopters. The board must engage in efforts to build a community of adopters that will install the software at their institutions and will pass the knowledge and practices gained in doing so back into the community knowledge base. A community needs trained developers who can contribute to the open source base with fixes and extensions to existing tools or contribute new tools that integrate with Sakai's architecture. Support for this effort comes through the Sakai Partners Program and the constant participation of community members on the open lists and in the development teams.

2) Sustaining a community of adopters. The board must plan for long term sustainability in the Sakai community and for the Sakai software itself. Without an effective method of sustained support the promise of an alternative to current software choices will be short lived, and institutions will not choose to participate in the bootstrapping necessary for the development of a long term success.

To achieve this the board implemented plans that included an evolution of the board itself as the core products matured and as the Partners Program and community as a whole gained strength. Over the two years of the Sakai Project, the participation of the partners has steadily and progressively increased, as the initial Sakai software was released and the conditions for increased community participation (such as the Requirements Process, above) put in place. As this happened, the board evolved into an institution more representative of the community, first by bringing selected partners onto the board, and then by initiating elections for the Sakai Foundation Board, the first being held in the last months of the grant period, late 2005. Since the SEPP (now SPP) was initiated as a 3 year program, the second year of the overall project was the period targeted for transition. This allows the third year to be one in which the initial SPP funding stream can be used to complete the transition to a subscription-based, community supported effort. The kind of developments outlined above are examples of this, and such experimentation and evolution will continue throughout the rest of the project.

Community Building - Governance Plans and Sustainability

The organizational model, with some enhancements as we have gone along, proposed by the Sakai Project members in the first year has proven to be a successful model. It currently has three components to engage different elements of the software world in higher education.

Sakai Partners Program

First, by building a member and subscription based organization from the ground up composed solely of organizations and institutions of higher education, we have guaranteed that the needs and requirements of this community will drive this and future development efforts. By involving the SPP members actively in planning and development through the Discussion Groups and Work Groups, we have begun the process of building an effective extended organization. By fulfilling our promise to bring SPP partners onto the Board at regular intervals, by the formation of the Sakai

Foundation and the elections for the Sakai Foundation Board held in the Fall of 2005, we have emphasized our commitment to a community run organization. As we experiment with different mixes of closely coupled development and the more distributed open source models of development, we are learning what can work in this community, a community that at times looks like a consortium of varying size, and at times can begin to look like an unbounded open source effort.

The Sakai Foundation

The evolution during the Mellon grant period to a formal organization that would be independent and inclusive, and that could manage the Sakai Community's IP, led to the formation of the Sakai Foundation. As the following press release describes, there was considerable effort involved, and high expectations accompanied it.

Sakai Partners Launch Sakai Foundation for Open Source Software

2005-Oct-17

ANN ARBOR, MI -- 17 October 2005 The Sakai Project today announces the creation of the Sakai Foundation to provide a permanent home for the growing Sakai Community. As a non-profit, membership corporation the Sakai Foundation will provide Sakai developers, adopters, and users a place to coordinate their efforts.

"We've been planning for the foundation all year," notes Joseph Hardin, who chairs the Sakai Project Board. "The 80+ Sakai Educational Partners and 12 Commercial Affiliates have participated in developing the governance plans for our software community via meetings, videoconferences, and online voting." The Foundation will oversee the Sakai Partners program and relationships with companies. All the founding institutions, the University of Michigan, Indiana University, MIT, and Stanford along with board members from Foothill College, UC-Berkeley, University of Delaware, U. of Hull (UK), and U. of Toronto strongly support the creation of the Sakai Foundation. "The community response has been universally positive," Hardin said.

The foundation marks an important step in the rapid rise of the Sakai Project (see sakaiproject.org). In less than two years, the project has attracted worldwide interest, managed enterprise-scale deployments, and developed a sustainability path with financial support into the future.

"Colleges and universities around the world are watching carefully as Sakai has transitioned from a grant project to a community-based foundation. With it go our collective hopes for a creative future. This is company that's good to be among," Phil Long, Senior Strategist for the Academic Computing Enterprise at MIT, said.

The launch of the Sakai Foundation builds on the highly successful Apache Software Foundation model. The Sakai software remains open for anyone to use, modify, and distribute without any obligation to join as a Sakai Partner. Those who wish to can become Sakai Partners and participate in the governance of the foundation and its operations by contributing $10,000 per year ($5,000 for smaller colleges). "This modest

cash flow into the foundation from members provides a real sustainability path for coordination and leverage of the Sakai Community work," Hardin noted.

The Sakai Project, announced in January 2004, promised to develop an open source Collaboration and Learning Environment for the needs of higher education. The Sakai Software is already deployed as the primary teaching and learning system at the University of Michigan with over 35,000 users. It is in a full parallel year at Indiana with deployment to 90,000. Major pilot projects and rollouts are underway at Stanford, UC-Berkeley, MIT, Rutgers, Yale, UC-Merced, UNISA (University of South Africa), Universitat de Lleida (Spain), Roskilde Universitetscenter (Denmark), Universidade Fernando Pessoa (Portugal), and others (see Sakai Pilot Projects at SakaiProject.org).

The Foundation will manage a small staff to coordinate evolution of the Sakai software, provide advanced developer support for members, conduct quality assurance work on Sakai releases, track contributor agreements and manage the Sakai IP, and manage conferences and meetings for the Sakai Community. Much of the innovation and tool development will continue to be done where it is best understood -- among the distributed community of Sakai users and developers.

"The Sakai Foundation creates a shared place that is wholly of, by, and for the educational and research community to coordinate a healthy ecosystem among all who wish to participate," Brad Wheeler, Indiana University and Sakai Board member, said. "The foundation provides all participants open licensing, leveraged work, walk away rights, no inhibiting NDA's or discriminatory pricing, and a place to shape our software destiny together."

The establishment of the Sakai Foundation is a key part of opening up the Sakai Project from the original work supported by the Mellon and Hewlett grants and founding institutions to a global open source Collaboration and Learning Environment community, including work in the eSciences and digital libraries.

"Here at Cambridge we warmly welcome the incorporation of the Sakai Foundation as a safe harbour for the Sakai code and a new focus for the growing community that is deploying and developing the Sakai Collaborative Learning Environment," said John Norman of Cambridge University.

"Already Sakai is allowing several inter-institution distributed research groups to innovate in the way they carry out their research and we are planning to promote Sakai as offering valuable tools to support every individual on our campus in addition to its groups and course management support," Norman added.

"The recent news in the learning management enterprise business underscores the wisdom higher education has exhibited in placing value and trust in the 'commons', and in particular in the commons tended by the new Sakai Foundation," Phil Long said.

"The formation of the Sakai Foundation is a momentous event that reaffirms the

collective commitment of higher education to provide organizational, legal, and financial support for the Sakai project and the community that has developed around it. I am filled with great pride for what we have accomplished in the last 18 months. It marks the beginning of greater innovation and collaboration - the real value of open source," said Vivian Sinou, Dean, Distance & Mediated Learning, Foothill College, and Sakai Board Member.

Open source software thus has a new foundation - the Sakai Foundation - leveraging the talent of institutions and individuals worldwide joining to build a common, open future.

The formation was preceded by extended discussions among the Sakai partner institutions, and agreements to contribute code, trademarks and other IP to the Sakai Foundation.

Sakai Commercial Affiliates

Second, by establishing the Sakai Commercial Affiliates as an organization composed of commercial partners interested in providing support for the Sakai software, as a group of "Sakai Redhats", we have helped to insure the existence of a wide range of commercial offerings, from consulting to complete ASP services, for those wishing them. This is a critical need to get Sakai software available to smaller schools and those not wishing to run the software themselves. An example of the success of this is shown in the adoption of Sakai by Appalachian College through the efforts of the Longsight consulting firm (see: sakaiproject.org/index.php?option=com_content&task=view&id=397&Itemid=312). It also gets some larger institutions over the hump of adoption of open source software, giving them someone to call late at night if they encounter problems. In short, the SCA lowers the adoption threshold for a number of schools and organizations and increases the likelihood of wide scale adoption. SCA members have also emerged as prime contributors to the development of the Sakai code base and community, with contributions from many SCA partners and sites like the Unicon "TestDrive Sakai" site providing valuable exposure for the Sakai software (https://www.academusopencampus.com/registration/register/index.php).

Working with Industry Organizations

Third, by helping to initiate the IMS Working Groups on Tool Portability and Common Cartridge, and working on a charter that is targeted at collaboration between providers of course management software, proprietary and open source, we have engaged this community effectively in the areas of mutual interest that will be of most benefit to the larger educational community, without getting bogged down in unproductive debates surrounding definitions of open standards implementations and standards compliance. By working with, among others, like publishers, WebCT/Blackboard, in developing actual examples of tool portability between all our systems, we are advancing the practice, as well as the theory, of tool portability, to the benefit of adopters of any of our systems.

All this of course points to a whole dimension of effort in the Sakai project, and especially the SPP development. We are not only developing software and methods as we go along. We are also developing, experimenting with, discarding some and choosing between, practices that will result in a sustainable model for Community Source.

Sustainability – Building Community Resources
The first years of the Sakai Partners Program have seen the steady and measured growth of an increasingly dedicated community of software designers, software developers, IT administrators, pedagogical technologists, User Interface and Usability experts, and support staff. Active and effective Discussion and Work Groups have arisen, and more will as we move along. Workgroups dedicated to adding functionality and reviewing existing software and solutions have been formed, and the strengths of the community are increasingly tapped and coordinated through the various Sakai practices.

The Sakai Foundation Board strongly believes that this is the way to build an enduring community of support within the higher education community: from the ground up, including staff and leadership from all levels in every stage of development. This rich mixture of commitment, experience and expertise from all levels of the individual academic enterprise and the academic community as a whole is necessary to move beyond traditional models of isolated software development, outsourced development, or mandated institutional collaborations, that are generally not picked up by a community of support, and toward a set of methods that increase the probability of sustainable success of open source efforts. This is not achieved by the application of resources alone, but by mobilizing the interest and commitment of staff and leadership throughout the academic organization. The open source model we are pursuing is founded on a mix of dedicated resources and volunteerism based on both enlightened self interest and a knowledge that contributed work can never be taken away, that the source remains open.

The success of these first years in developing a resource base for the Sakai Foundation and in involving and exciting a diverse, international community of developers and users, from large research institutions to small liberal arts colleges to community colleges, leads us to feel confident that the future will see even more progress. Add to this the continuing commitment of core schools and the deepening commitments of new institutions and it is clear that we are well on our way to meeting our goals of having a vibrant and self-sustaining community-based organization to see this community into the future.

Sakai Project Report: The Evolution of the Sakai Architecture

Charles Severance (csev@sakaifoundation.org)

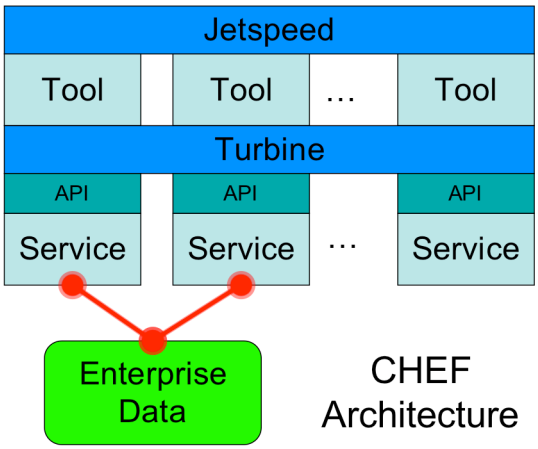
This section is a historical perspective on the architecture activity throughout the Sakai Project. The Sakai Project was a very fast paced project and it was necessary to rapidly evolve the Sakai Architecture during the project to meet the emerging requirements of the Sakai project and ultimately Sakai community.

As of the Sakai 2.2 release, the Sakai Architecture is relatively stable and while the architecture will continue to evolve, as Sakai is in production at hundreds of places, architecture changes will naturally be slowed, becoming more considerate and deliberate so as to provide the greatest benefit to the community.

Since the Sakai code base started from the CHEF code base a number of aspects of the CHEF architecture were inherited by Sakai from Sakai 1.0 and forward. However, the architecture of Sakai evolved significantly during the project initially under the guidance of the Sakai architecture team and later under the guidance of the Sakai community.

The CHEF Architecture

The CHEF architecture was based on building Jetspeed portlets and the Apache Turbine service framework. Jetspeed was an Apache open source portal which pre-dated the current JSR-168 Portlet standard (see http://portals.apache.org/jetspeed-1/). Turbine was an aging Apache project that had already been replaced by the Apache Avalon project.



The CHEF Architecture had two strong aspects that remain in the Sakai Architecture:

- CHEF's architecture was carefully layered in that all presentation (Tool) aspects were kept separate from the Services which contained the business and persistence logic. Services were programmed as implementations of abstract well-defined Java interfaces that allowed for the substitution of the implementations of each Service. Chef maintained two versions of each service - a version that stored all data in XML files for developers and a version that stored data in the database for production. It was a simple reconfiguration step for system administrators to pick and choose amongst the different service implementations. CHEF used Turbine to make the connections between the Tools and the appropriate services.

- CHEF had a basic integration for Users and Roles in the form of a set of provider interfaces that allowed for each site to integrate their own directory and role services. This allowed each site to integrate their enterprise data into Sakai without modifying the Sakai code base. These generalized plug-ins provided a direct way for an organization to integrate Sakai into an existing Service Oriented Architecture approach to their user and directory information.

At the beginning of the project, there were a number of weaknesses and shortcomings in the CHEF architecture that needed to improve including:

- As the portal world was evolving beyond Jetspeed, we did not want Sakai to be based on the Jetspeed portal. We wanted Sakai to comply with standards and support more modern presentation technologies.

- We did not want the Sakai architecture to depend on the Turbine technology for service discovery. Turbine was an aging technology and we did not like a code-dependency on Turbine scattered throughout the Sakai code base.

- We did not like the Velocity templating language that was in use by Jetspeed. We wanted a more abstract presentation layer with the potential to have our tools used as Desktop applications.

- We wanted a better data abstraction - CHEF used SQL with a home-grown abstraction layer.

- We wanted to improve the integration of Enterprise data into Sakai. The User and Role providers needed to be improved and new providers were needed for Course information, Digital Repository integration, and others.

Given the need to evolve the Sakai architecture and address these challenges, the Sakai Project was formed with an Architecture team as part of its governance structure.

The Sakai Architecture Team

The Architecture team was charged with evolving the architecture to address the limitations of the CHEF architecture to insure that Sakai would be more broadly usable across all of the Sakai adopting sites. During the first 18 months of the Sakai Project, the architecture team was made up of lead technical representatives from each of Sakai project participants (Michigan, Indiana, Stanford, MIT, Berkeley, OKI, and uPortal).

- Charles Severance, Lead Architect
- Glenn Golden, University of Michigan
- Lance Speelmon, Indiana University
- Craig Counterman, MIT
- Rachel Golub, Stanford University
- Mark Norton, Sakai SEPP Representative
- Ken Weiner, Unicon, uPortal Project Manager
- Daisy Flemming, Stanford University
- Lydia Li, Stanford University
- Peter Wilkins, OKI Project
- Jeff Kahn, OKI Project
- Ray Davis, University of California Berkeley
- Josh Holzman, University of California Berkeley

This team met regularly throughout the first 15 months of the project to evolve the architecture through the Sakai 2.0 release. During the preparation for the Sakai 2.0 release, Sakai was transitioning from project-based governance to community-based governance so the activities of the Sakai Architecture Team was transitioned to community processes with the Sakai Framework Discussion Group.

By the end of the project with the Sakai 2.1 release the Sakai architecture is relatively mature - with over 500 pages of architecture documentation and thousands of pages of developer documentations. The Sakai Architecture continues to evolve during the community phase, albeit more slowly than during the project phase.

The Sakai 1.0 Architecture

The primary goal of the Sakai 1.0 release was to quickly address the major shortcomings in the CHEF architecture and get a release in the hands of developers so that new services and tools could be developed to address the immediate tool needs of Stanford, Michigan, Indiana, and MIT.

The Sakai 1.0 release made a number of significant architectural changes that resulted in a dramatic departure from the CHEF architecture.

- Replacing the dependency on the Apache Turbine framework with the newly emerged Spring framework. The Spring Framework had many advantages over Turbine. The most significant advantage was the concept of "Inversion of Control" (IOC). IOC allows Spring to assemble and "wire up" Tools, Services, and Enterprise Service Integrations without requiring any Spring-specific code within any of the Tools, Services, or Enterprise Integrations.

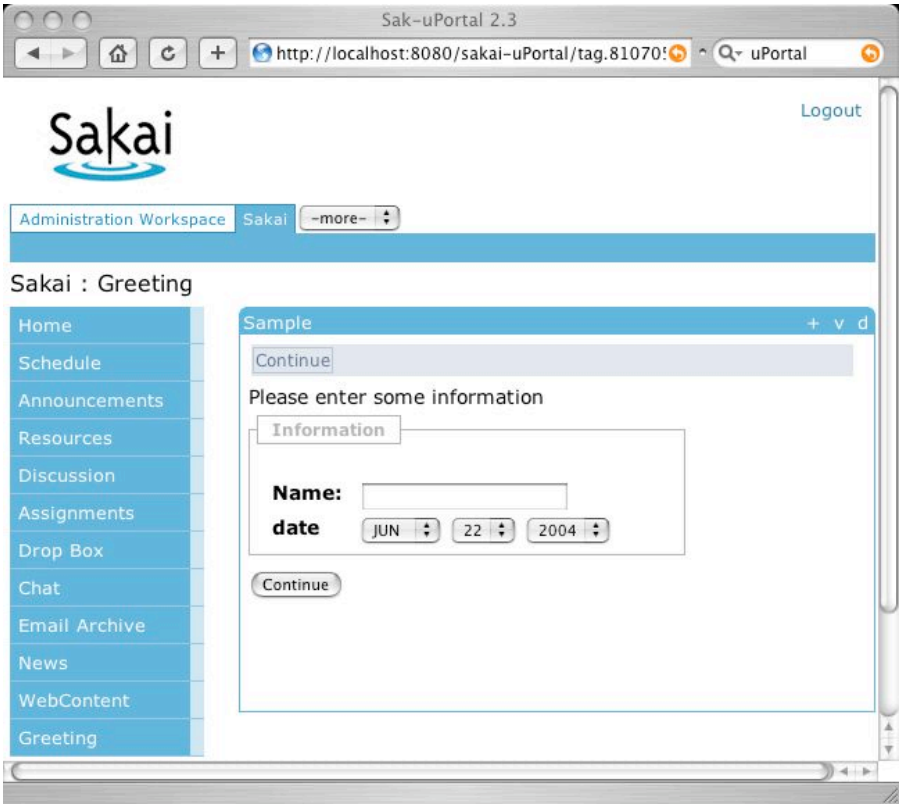
- Add support for Sun's Java Server Faces (JSF). JSF appeared to be a major step forward in the evolution of web frameworks that implemented the Model/View/Controller pattern. JSF used a symmetric Java Bean pattern as its interaction between the template language (JSF Components in JSP) and the presentation login (backing bean). In addition the JSF components allowed for the development of Java Swing implementation of the components as well as a JSR-168 (portlet specification) of the components. We hoped that by adopting JSF we could enable a single Sakai tool to be usable across several presentation technologies without modification.

- Removing Jetspeed from the Sakai architecture.

- Improving support for the providers used to integrate Enterprise Services into Sakai. A new Course Management provider was added to generalize the way that Sakai retrieves course lists from the University Student Information System.

These architecture changes were done very quickly and prerelease versions of Sakai 1.0 with these changes were delivered during the period February 2004 through September 2004.

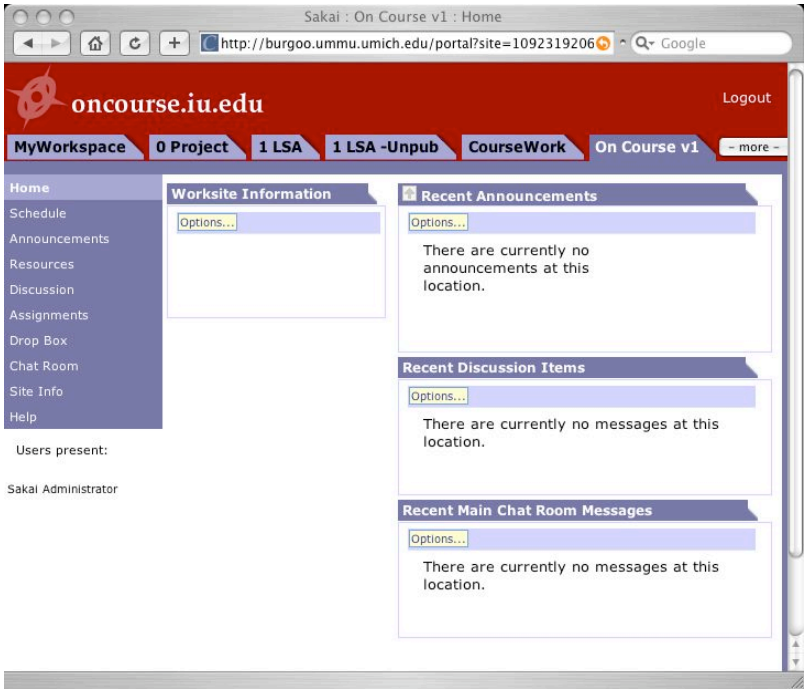
During the Spring of 2004 we performed an initial integration between Sakai and uPortal where uPortal functioned as the aggregator for Sakai content. The following shows a JSF based Sakai tool running in uPortal in June of 2004 at the time of the Sakai 1.0 Beta.



This integration was technically successful and proved that our new presentation approach was indeed a successful abstraction as it transparently supported either uPortal or the Sedna aggregator. However once we looked at what had been changed in uPortal to work with Sakai, it was decided that too many drastic changes had been made support the layout needs of Sakai. We felt that if Sakai were to distribute this heavily modified version of uPortal, it would create a dangerous fork of the uPortal code base in a way that would be harmful to the uPortal community.

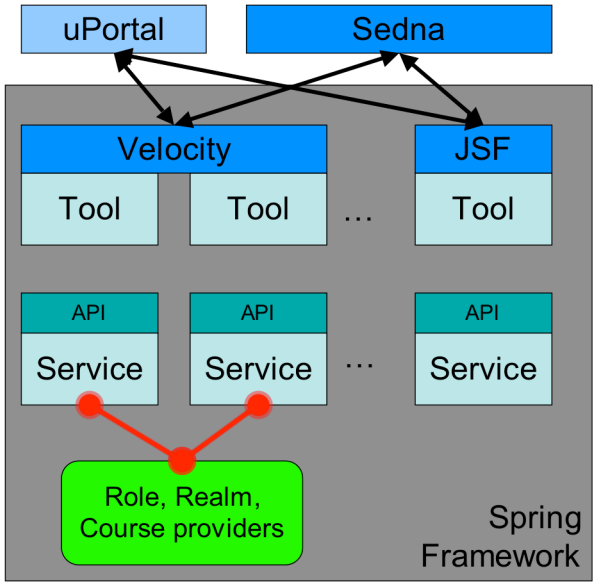
In conversations with the uPortal leadership, the decision was made to avoid producing a Sakai-unique version of uPortal and instead to focus efforts on integrating Sakai into an unmodified uPortal using the JSR-168 standard and the WSRP (Web Services for Remote Portals) protocol. With funding and direction from Sakai, uPortal had developed support bother for JSR-168 and WSRP in the 2.4 release of uPortal.

As a result, Sakai 1.0 shipped with the Sedna internal iFrame based aggregator.



Sakai 1.0 was put into production at the University of Michigan and Indiana University with user bases starting at 10,000 and moving upwards.

The Sakai 1.0 Architecture was as follows:



Sakai 1.0 Architecture

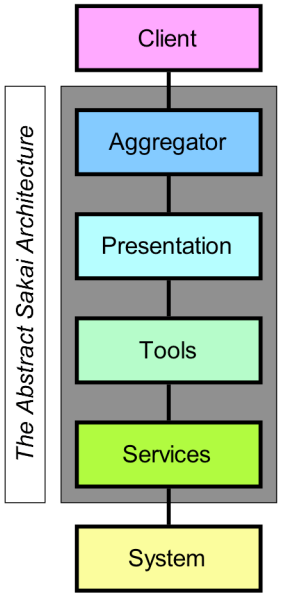
The integration to uPortal was completed but not shipped as described above.

The Sakai Abstract Architecture

During the development of the Sakai 1.0 release, the Architecture team was meeting to plan the strategy for the evolution of the Sakai Architecture - one of the first efforts was to define an abstract architecture.

The purpose of the abstract architecture was to identify critical areas where the Sakai needed interfaces and abstraction. We needed a modular architecture and knew as the community grew, clean abstractions would allow development to happen in parallel across many institutions associated with Sakai across the world.

The following diagram was initially presented by Mark Norton as a summary of several months of discussion in the Architecture group. This was adopted as the Sakai Abstract Architecture and has remained consistent as pattern throughout the remaining time of the project and as well as during the community phase of Sakai development.



Sakai applies modularity and interfaces to all levels of this diagram - not just the interaction between the tools and the services. We hope to be able to produce clean boundaries so that Sakai could allow undetectable substitution of implementations at any level of the Abstract Architecture.

Each layer within Sakai has the following purpose:

- The **Aggregator** is fully responsible for the ultimate user interface - it determines how sites and tools are laid out, how navigation between tools happens. There have been a number of different aggregators produced in the project including Sedna, Varuna, Charon, SakaiDesktop, etc.

- The **Presentation** layer is responsible for providing the framework capabilities in order for tools to effective reuse presentation elements. And example component would be a

widget that makes a calendar picker or WSYWIG editor available in a uniform fashion across all tools. One goal was to centralize these reusable presentation elements so as to encourage consistency across tools. Sakai's primary presentation layer is Java Server Faces with a umber of presentation technologies such as Velocity or Spring MVC and more recently experimented with Real Server Faces (RSF).

- The **Tool** layer is where the business logic of the tool resides. One of the key aspects of the tool layer is that it represents data and information associated with each user's session. A tool handles session state, and handles the navigation between the different views of the Tool. The tool is not responsible for the long term persistence of data.

- The **Service** layer is where data is stored and all of the non-presentation business logic is kept. For a properly written Sakai tool, interface between a Service and a Tool is the entire contract between a tool and persistence.

Above Sakai is the end user's client environment and below Sakai are the basic system resources like the Java Virtual machine, CPU, memory, file system, database, etc.

The Sakai 1.5 Architecture

Sakai version 1.5 was developed and released in a relatively short time period from October 2004 through December 2004 so there were very few changes to the overall architecture from the 1.0 architecture.

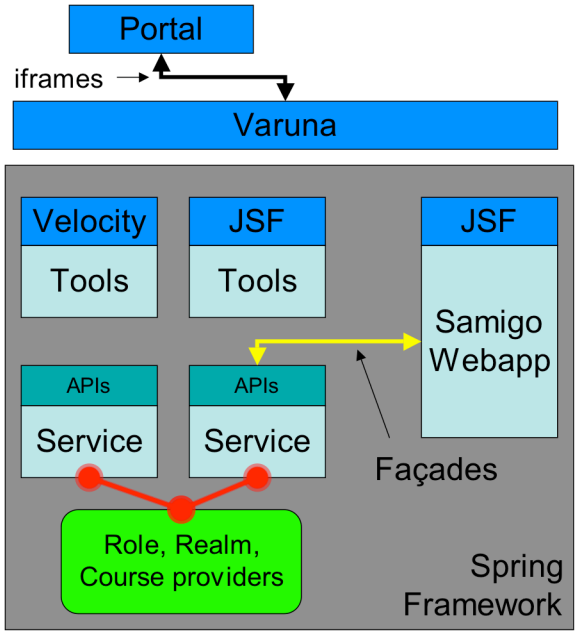
Once the Sakai 1.0 release was put into production at a user level above 10,000, it turned out that some of the implementation patterns of the Service implementations in Sakai 1.0 over utilized memory and underutilized the database's capabilities and resulted in performance bottlenecks.

Much of the effort in 1.5 was effectively performance tuning Sakai 1.0. This was made much easier because Sakai had already well-defined services and the tool code was relatively unchanged for 1.5 even though significant work had been done on the service implementations.

Also during the Sakai 1.5 development, we dropped support for the XML implementations of the Sakai Services in favor of using Hypersonic SQL as the out-of-the-box database used for developers and the Sakai Quickstart distribution.

Toward then end of the 1.5 development cycle, we began to integrate the Samigo testing engine from Stanford into the Sakai release. Because of late arrival of some of the components in the Sakai 1.5 framework, Samigo development moved forward as a stand-alone web application that interacted with the rest of the Sakai framework through a set of Façade classes. Integrating a stand-alone web application proved challenging for the 1.5 release and showed limitations in the Sakai 1.5 architecture.

Another architecture effect was the increasing need to closely integrate the Open Source Portfolio (OSP) into Sakai. Up to the Sakai 1.5 release, OSP was forking the Sakai code at each release and patching in capabilities needed in the Sakai framework to support OSP's development approach. This was becoming painful and time consuming as both OSP and Sakai were rapidly moving projects. For Sakai 2.0 we wanted to merge the needed OSP capabilities into the base Sakai release.



Sakai 1.5 Architecture

Sakai 1.5 replaced the Sedna aggregation layer with Varuna. The primary benefit of the Varuna aggregator is that all of the iFrames in the aggregator had well-documented URLs which allowed the inclusion of Sakai within any portal using an iframe and Single Signon shared between the portal and Sakai.

The above shows an iframe integration between Sakai and uPortal in production at the University of California Merced.

Planning the Sakai 2.0 Architecture

During the summer of 2004, there was a joint meeting of the Sakai Board and Architecture Team at Indianapolis. This was a important meeting with two valid but very divergent points of view emerging during the retreat as to what was the highest priority for resources in the Sakai project.

- One viewpoint suggested the highest priority was to throw out the Sakai 1.0 framework code (keep the tools and Abstract Architecture) and build a whole new framework to implement the Abstract Architecture.

- The other viewpoint felt that while a "new framework" was highly desirable the highest priority was to add needed features (we called them "the gaps") so that the schools could put Sakai into production and meet user needs and requirements.

Neither group was right or wrong - ultimately a compromise was worked out. We would make features the highest priority through the end of the 2004 and then early in 2005 begin work on Sakai 2.0. During the Fall of 2004, Craig Counterman of MIT was to evaluate design patterns for the new framework. The Architecture was code named "Murimoto" or "Framework II".

Our experiences during Fall 2004 effort shaped the shaped Sakai 2.0 framework. Craig's research produced some simple and elegant technical approaches to problems that plagued the Sakai 1.0 and 1.5 frameworks - also the experience with OSP and Samigo integration led us to feeling that we needed to be much more flexible about the definition of "what was a Sakai tool".

Originally the Sakai 1.0 framework was intended to be very restrictive to enforce and only allow two types of tools in Sakai - Velocity (we called this legacy) and JSF. The ultimate plan was to rewrite all tools in JSF when time permitted. But by the end of 2004, it was very clear that the Sakai 2.0 framework needed to be friendly to web applications in general rather than trying to limit what developers could do within Sakai in the name of look and feel consistency.

Developing Sakai 2.0

The Sakai 1.5 release was relatively mature in terms of system reliability and as of 1.5.1, with significant attention being given to general tuning and database related interactions within Sakai, its performance was very strong, so we truly had time to start the Sakai 2.0 effort "from scratch" as suggested during the Summer 2004 retreat meeting. The Sakai 2.0 effort could be likened to moving from a "monolithic" UNIX kernel to a "modular" UNIX kernel and all that entails.

The Sakai 2.0 development started with a completely empty code space in January 2005 and started building the framework from the ground up making use of the solutions to the problems identified in the results of Craig Counterman's research. Each new module was added carefully with intent to keep the code base as clean as possible. Design documents were produced and reviewed for each component by the Architecture team. These documents are available at:

https://source.sakaiproject.org/svn/reference/trunk/docs/architecture/

These documents are regularly updated as the framework is changed and consist of over two hundred pages of Sakai's documenting how Sakai works internally and how to develop within Sakai. These documents are part of each Sakai release and are tagged with the release so it is possible to look at older versions of these documents at the same time as you are looking at the code for the older versions of Sakai.

For more detail on the Sakai 2.0 architecture, you should look at those documents.

There were many innovations in Sakai 2.0 - the most significant were as follows:

- The introduction of Components that created services that had their own class loader and a deterministic loading order and far cleaner integration into the Spring framework. In Sakai 1.5 services were kept in servlets and there were situations where Sakai 1.5 would have trouble starting without deadlocking. We found a workaround for the problem in 1.5 but built 2.0 to completely eliminate any potential for webapp startup order to cause any problems.

- Supporting the aggregator protocol in a Servlet Filter rather than the servlet itself. This meant that simple servlets could be integrated into Sakai by only adding elements to the web.xml servlet configuration file, rather than modifying the Java code of the servlet.

- Support for *any* presentation technology to be used in a webapp with the addition of a bit of framework glue. The community quickly extended Sakai to support Struts, Spring MVC, and Cold Fusion within Sakai in addition to the existing support for Velocity and JSF.

- A rapid expansion of the reference implementations of the Sakai User Directory providers. The community quickly produced reference implementations for: Kerberos from Seth Theriault of Columbia University, JLDAP from Ravid Ross of Albany Medical College and Rishi Pande of Virginia Tech, and OpenLDAP from Adrian Fish at Lancaster University, Jose Garcia of Universitat de Lledia, and Alex Balleste of Universitat de Lledia.

- Sakai 2.0 added support for web services using the open source Jakarta Axis web services engine. Since Sakai already a very modular architecture with the main functionality described in clear API contracts, adding web service support across all of the Sakai APIs was straight-forward. The actual development of the detailed web services was done completely by the Sakai community led by Steven Githens of Northwestern University, Seth Theriault of Columbia University, and many others. As Sakai 2.0 was deployed, sites increasingly used Sakai's SOAP web services, allowing Sakai to be orchestrated by the other parts of an organization's Service Oriented Architecture.

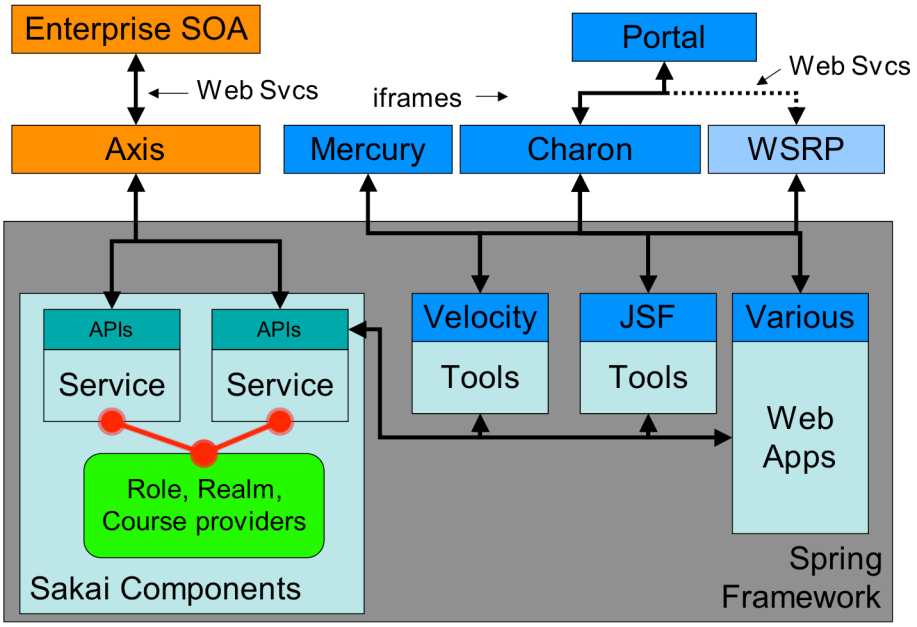
- The protocol between the aggregation layer and the tools was cleaned up and fully documented. The new default aggregator was called Charon and maintained the Varuna ability to use iFrames for portal integration. Sakai also shipped with a second aggregator called Mercury that was much simpler than Charon and intended for use by developers during tool development.

- The development of an implementation of Web Services for Remote Portlets (WSRP). This was developed but not released. WSRP also used and improved the Sakai 2.0 aggregation / tool protocol.

- A pass through all of the Sakai tools to normalize the user interface according to the Style guide produced by the Sakai Tools Team. Daphne Ogle (now of UC Berkely) led this work.

The overriding goal was to simplify everything down to its most minimal level. The idea was that a Sakai tool was a Servlet/webapp with only the minimum added to make it so the Sakai APIs would function in that webapp.

The ground up framework rewrite progressed smoothly for the first months of 2005 and the Sakai 2.0 framework was substantially complete by April. This left very little time to refactor any of the tool code from Sakai 1.5 and we were running out of time for the June 2005 release. So in the interest of time, we placed all of the cleaned up Sakai 1.5 tools into a single large module called "legacy".



Sakai 2.0 Architecture

Sakai 2.0 shipped on time June 15, 2005 and marked a very important tuning point in the project. As of the Sakai 2.0 release we had a very solid code base and a very nice implementation of the Sakai Abstract Architecture.

During the first 18 months of the project we had rapidly evolved the architecture to improve the abstractions and implementations of the layers of the architecture: Aggregation, Presentation, Tool, and Service. After the 2.0 release, we turned our focus from rewriting the low level Sakai framework "glue" and focused on evolving Sakai's capabilities within the context of the forming Sakai development community beyond the scope of the original grant/project.

Sakai 2.1 Architecture

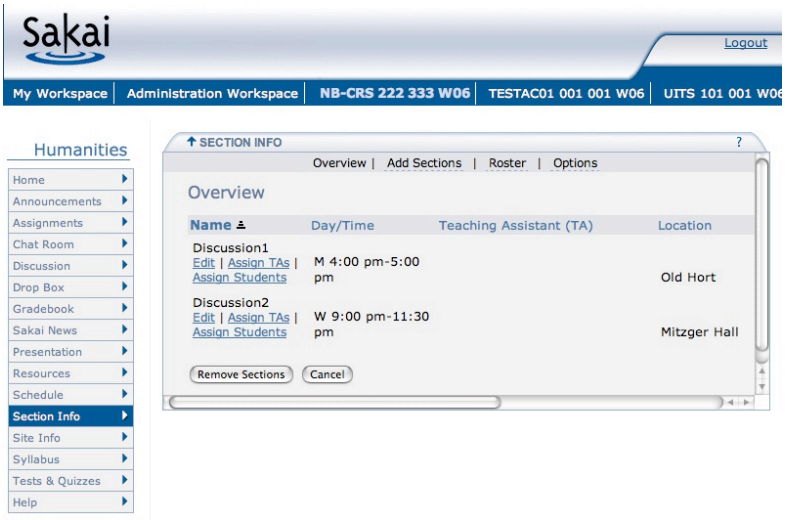
The Sakai 2.1 release was evolutionary and filled in gaps in the Sakai 2.0 architecture.

Sections and Groups in Sakai 2.1

From the beginning of the Sakai project, one of the most-requested features was to support groups or sections within a course for courses with large enrollments and teaching assistants. This capability was added to the Sakai Services for the 2.1 release.

This work was led by Oliver Heyer, Josh Holzman, and Ray Davis of the University of California, Berkeley. The design allowed for supporting both simple groups as well as complex

sections and included a Section Tool that allowed instructors to control the creation of the sections using a number of different approaches including dynamic signup for sections.

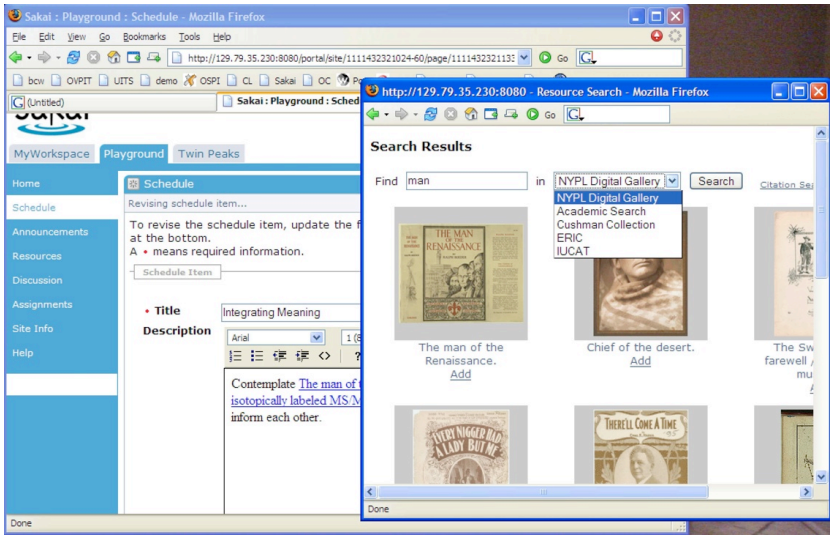


Sections and groups were supported in the Announcements Tool, the Samigo assessment system, and the Gradebook tool developed by University of California Berkeley.

OKI OSIDs in Sakai 2.1

Digital repository integration is an important aspect of the Sakai Project. As early as the 1.5 release there were a set of patches and demonstrations which were developed to explore this important area. This effort was called "Twin Peaks" to symbolize the equal importance of the collaborative learning environment and the campus library system to our user community.

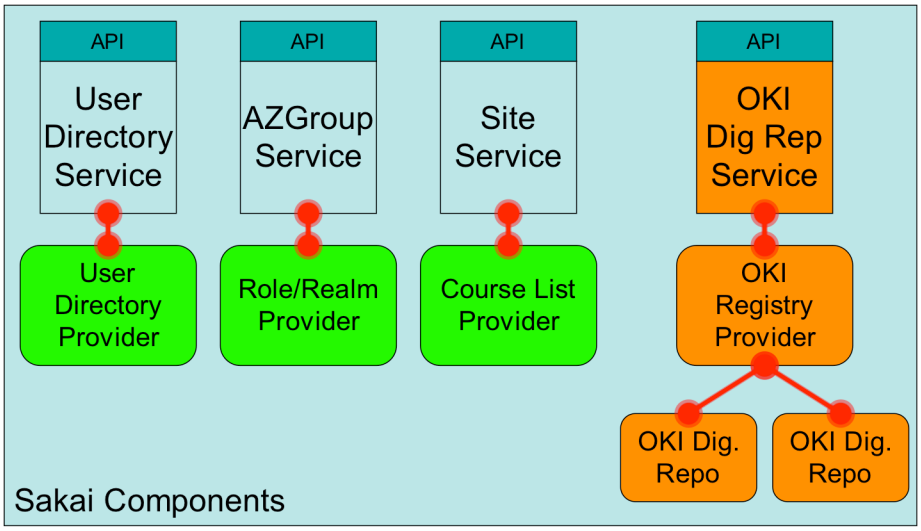
Steve Smail of Indiana University and Jeff Kahn of MIT and the OKI project led this work. TwinPeaks was available as a patch to Sakai 1.5 integrated directly into the text editor used by many tools:



For Sakai 2.1 we integrated the TwinPeaks capability directly into the Sakai release by adding general support of OKI OSIDs into Sakai and then developing the following OSID implementations:

- ID OSID
- Logging OSID
- Registry OSID
- Digital Repository OSID

The OKI support gives a new way to configure Sakai behavior by implementing services to provide information to Sakai.



OKI Digital Repository Service Integration

The TwinPeaks capability was replicated in Sakai 2.1 using this new Digital Repository OSID support.

This work continues under a separately funded project called SakaiBrary. The goal of SakaiBrary is to make it so that resource that can be searched and used through the OKI Digital Repository OSID can be used throughout Sakai in the same places resources stored within Sakai can be used.

Unit Testing in Sakai 2.1

An important addition to the Sakai framework was improving support for unit testing. Unit testing is essential to produce high quality code and insure that the code remains high quality over time as the code changes and evolves. Prior to Sakai 2.1 there was very limited unit testing because it was very challenging.

Starting with Sakai 2.1, Josh Holzman began to organize the unit testing efforts so that we could approach unit testing in a more consistent and coherent manner. Josh began to develop reusable unit testing support that replicated the run-time environment of the Sakai framework. Josh also

has acted as a mentor to Sakai developer community who need to develop unit tests for their application.

Integrating Sakai 2.1 into Stellar at MIT

Early in 2006, MIT undertook an effort to take advantage of the new Sakai framework to integrate Sakai into their Stellar environment. The goal was to integrate some of the Sakai tools into Stellar maintaining the Stellar user interface. This effort was led by Mark Norton and Craig Counterman and portions of this work are expected to be in production at MIT Fall 2006.

There were several elements to this approach:

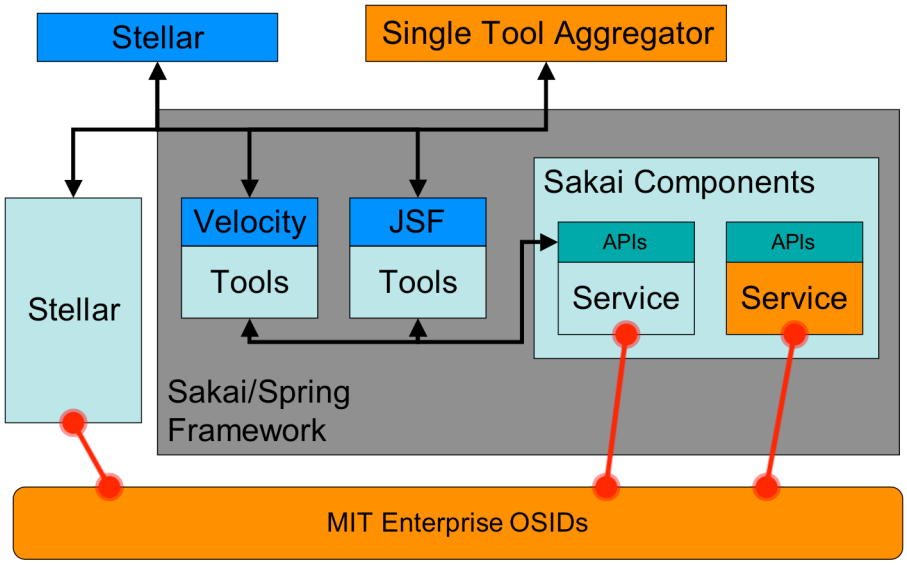
- Building a "Single Tool Aggregator" which would simply display a single tool with none of the navigation that other aggregators such as Sakai's Charon would add to the output of the tool.

- All navigation between sites and tools was delegated to the Stellar user interface which already had site and tool navigation and Stellar already supported a rich hierarchy between Stellar Nexuses (Stellar term for Site is "Nexus").

- Using a set of OKI OSID implementations, integrate the MIT Enterprise integration into both Sakai and Stallar. This demonstrates a basic value proposition of OKI OSIDs in that from the MIT Enterprise data perspective Sakai and Stellar were two applications, but from the end user perspective at the Aggregation layer, they are seen as a single integrated application. OSIDs allowed the reuse of identical integration code in Stellar and Sakai.

- Make modifications to Sakai services as necessary to create and/or improve the Sakai Provider patterns so as to be able to inject the needed data into the Sakai services to support the new requirements identified in the effort.

This is the architecture of the Stellar-Sakai integration. This work was presented at the June 2006 Sakai conference and generated great interest around the notion of embedding Sakai capabilities in other applications.



Integrating Sakai into Stellar

This effort was made more challenging because it was based on Sakai 2.1 and not Sakai 2.2. If the MIT team could have taken advantage of the code refactoring in Sakai 2.2 it would have made the Stellar-Sakai integration a much simpler task.

Because the Sakai 2.2 effort and Stellar-Sakai integration was being done at the same time, there was a good deal of idea sharing among the teams. Many of the problems that were encountered in the Stellar-Sakai integration were simply solved in the Sakai 2.2 release by having Mark Norton deeply involved in the review of the interim steps of the 2.2 refactoring as it was being produced.

This project produced a point solution for MIT and informed the Sakai 2.2 refactoring and it points the way forward toward needed innovations in the Sakai framework to support this deployment pattern/use case in a general fashion.

Because of the timing of the two efforts, not all of the MIT modifications made it back into the Sakai code base for the 2.2 release. It should not be challenging to bring the rest of the concepts used in the MIT project back to the Sakai code base as part of the normal development cycle for Sakai 2.3.

The Sakai 2.2 Architecture

While the Sakai 2.2 release was delivered in July 2006, six months after the Sakai project was completed, it completed several of the architecture tasks that were left unfinished at the end of the grant period of Sakai.

The primary changes to Sakai 2.2 were a complete refactoring of the code base, breaking the legacy module into a number of separate components and reflecting the Sakai Abstract Architecture in the structure of the code base.

The results of the framework delivered in Sakai 2.0 and the code refactoring in Sakai 2.2 leave the Sakai Foundation with an excellent code base that is well suited for extensive parallel development by a large community-based development team.

Remaining Challenges for the Sakai Architecture

We have accomplished many things in a relatively short time frame with respect to the Sakai Architecture during the Sakai project. Even with this success there are many important areas of Sakai and the Sakai architecture that need to be improved over the next two years and beyond.

Sakai is in a strong position to be more patient with the architecture changes going forward, as Sakai 2.2 is a very solid code base with a very solid framework implementation. Hopefully by making architecture changes more gradually, we will be able to improve and extend the architecture over the next two years with less impact to the Sakai tools and the Sakai developer/adopter community.

The Sakai Foundation has three mechanisms for evolving Sakai and the Sakai Architecture:

- The first high level Roadmap for Data Interoperability was published in May 2006. This outlined at a high level a multi-year plan to identify and address data portability and interoperability issues that will allow Sakai to be increasingly used in a Service Oriented Architecture. The roadmap is not a precise project plan - its intent is to help organize the volunteer resources within the Sakai community. Also this document can serve as supporting materials for members of the Sakai community to request funds to address these important needs.

- A Sakai Requirements process to identify and prioritize needed changes to Sakai. Peter Knoop (Sakai Project Coordinator) is responsible for working with the volunteers in the Sakai community to try to find resources that can address the most important requirements as identified by the requirements process.

- The Sakai Architecture and the Sakai Framework are now evolved as a part of the community process around Sakai. Glenn Golden is the lead committer for much of the Sakai Framework code base and he works with a team from around the world to evolve the Sakai Framework (the implementation of the Architecture) together.

Several important areas where improvement is needed are highlighted in the rest of this section.

Hierarchy

One of the weaknesses of the Sakai is that each Sakai site is an "island". Which this is a powerful approach and has served Sakai well, there are a number of important use cases where a hierarchy between sites would be very useful. Three important use cases are:

- Assigning departmental, college, or campus level permissions to enable a distributed technical support staff. This can be simulated in Sakai by adding staff explicitly to each of the sites, or even by building this into the Sakai Providers, but a far better approach would be to simply put sites into a hierarchy and granting administration to users at various levels of the Hierarchy.

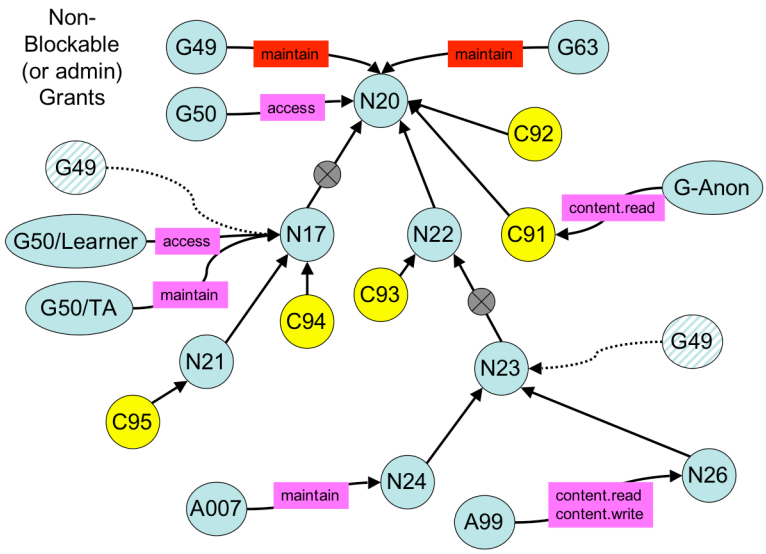
- Building a course site out of "sub-courses" granting teaching assistants or different instructors different access to these sub-sites as to the main site.

- Assigning fine-grain permissions to the Resources area to implement structures such as Drop Boxes where a student can write data, and the teaching staff can read and write data. This feature is currently supported within Sakai, but without a clean hierarchy, the code is brittle and increasingly difficult to maintain.

Hierarchy also has a number of valuable use cases for research and ad hoc collaboration applications of Sakai.

We have been able to introduce features (primarily groups) to partially meet some of the use cases for Hierarchy but we still need to produce the truly general solution.

We have had a design "on the books" for hierarchy support within Sakai for over a year but have never had time to focus on implementing the design.



This is an area of the Sakai architecture that needs to be modified with care. Like many software packages that use fine-grained Authorization, it is very important to carefully implement and tune Authorization code to maintain our current scalability to over 100,000 users.

Going forward beyond Sakai 2.2 we hope to have time to properly implement and test hierarchy without disturbing the Sakai developers and while maintaining the high performance of Sakai.

Improving Portal Integration

One of the recurring themes of the Sakai Architecture is the repeated attempts to integrate Sakai into portals in a way that will meet the use cases of the end users and portal administrators. Limitations in JSR-168, WSRP, and the WSRP4J code base have kept us from delivering a solution that is satisfying to Sakai and to the portal community.

The current best solution is either using iFrames directly through Charon or the Sakai JSR-168 proxy portlets that also use iFrames, but move navigation into the JSR-168 portlet.

Our efforts going forward fall into the following categories:

- Improve the JSR-168 portlets including eliminating iframes using web-clipping techniques.

- Continue to explore possible WSRP 1.0 uses. Unfortunately the very poor quality of the Apache WSRP4J code base makes it unlikely that WSRP 1.0 will ever reach production quality in open-source applications.

- Build a set of RSS feeds for Sakai and use RSS portlets to allow portal staff to easily create custom Sakai "dashboards" within their portals.

- Add the iCal protocol to Sakai so that Sakai events can be viewed in a portal-based calendar system that pulls in feeds via iCal.

- Work with the JSR-286 (Portlet 2.0 API) and WSRP 2.0 standards efforts so that we are more aware of these standards *before* they are released and where necessary influence their directions going forward so as to better support Sakai's use cases when these standards are released.

- Add support to Sakai for JSR-168 portlets using the Pluto 1.1 container. This will add one more option to developers who want to develop a Sakai tool. This way a portable tool could be developed which would both run within Sakai and within a portal.

We have much to do in the area of connecting Sakai properly to portals - the good news is that a strong community is evolving around this problem of sites which both have an enterprise portal and Sakai. These sites will provide the detailed use cases and the ultimate high quality implementations of the code to deliver an integration which is needed by the end users.

Import/Export and Data Interoperability

With the pace of innovation in the Sakai project consistent support for Import/Export across tools has fallen behind. Sakai 1.0 was the last release that could 100% import and export an entire site using a single file.

Sakai 2.2 has resolved some of the Technical issues which were limiting a complete cross-tool import and export for Sakai and so we need to go back and connect all the Sakai tools to the import and export process.

In addition we need to switch our import and export format to use the emerging IMS Common Cartridge profile for IMS Content Packages. Sakai already has Beta-level support for a limited subset of IMS Common Cartridge, but we need to invest the effort to provide 100% coverage for IMS Common Cartridge as well as make the default export format for Sakai be IMS Common Cartridge.

Work is currently underway on profiling individual tools and developing a set of associated metadata, and this will feed back into the Common Cartridge work, as well as other IMS standards. This work also simultaneously looks forward to capabilities based on rich metadata descriptions and the management of arbitrary content. This is part of the effort to make it easy to look inside tools in Sakai and move materials from the “fair use” environment of the classroom to the “open web site” arena of the OpenCourseWare site.

Evolving the Sakai Presentation Layer

When the Sakai Architecture was first developed and presented in February 2004, the centerpiece of the presentation layer was Java Server Faces. At that time JSF was in very early days and the only available implementation of JSF was a Beta version.

JSF was very promising as it gave a clean abstraction between the Tool and Presentation layers using Java Bean patterns. It provided the hope of building an HTML-free Tool layer, outstanding component reusability in the Presentation Layer, and multiple implementations of the presentation layer including Servlet, Portlet (JSR-168), and Swing support.

While Sakai continues to use JSF as its preferred presentation technologies, there are a number of disappointing aspects:

- Building components is very unwieldly and as such Sakai's component library has not evolved to meet the developers' needs. Thankfully the MyFaces project and Oracle's recent donation of its ADF components has provided a better set of reusable components. Even with these component sets, each tool ends up writing special components for their own use that end up fragmenting our presentation layer.

- JSF ends up with a very heavy memory footprint for each user's session with the storage of the entire component tree from request to request.

- A weak component library leads to developers mixing JSF control structures (JSTL) with JSF layout directives in the same JSP files. While this is expedient for individual developers, it blurs the boundary between Tool and Presentation layers in the Sakai Architecture.

There are many problems with JSF that the Sakai community has discussed at length so the community is well aware of the issues. Going forward there are two possible directions and both are being pursued.

One direction is to "fix" JSF. This entails going back and cleaning up our JSF support and our use of JSF in tools. Ray Davis of the University of California Berkeley is leading this effort.

Another direction is to simply discard JSF in favor of a better presentation framework. No framework has been identified nor chosen at this point. There are a number of criteria that we are looking for in an ideal Presentation technology.

- True abstraction between the Presentation and Tool layers
- The Tool layer should not be aware that they are in a web browser or even HTML environment
- GUI Widget reusability in the Presentation layer hiding widget implementations from the Tool layer
- Able to produce markup for multiple types of aggregators using only the Presentation layer (Sakai, JSR-168, WSRP)
- Support multiple types of ultimate display devices using only the Presentation layer (Browser, PDA, etc)
- Support internationalization and localization
- Be as flexible as possible - support CSS and allow transformability of the user interface under control of the end user to the extent possible exclusively in the Presentation Layer

One promising potential new presentation approach is called "Reasonable Server Faces" (RSF) developed by Antranig Basman of Cambridge University. RSF was designed after a careful evaluation of JSF's strengths and weakenesses on many levels. RSF suggests the promise of retaining all of the benefits of JSF and eliminating all of the weaknesses. Antranig has already demonstrated the ability of an RSF tool to generate HTML, JSR-168 markup, or output in Spring MVC without requiring any changes to the tool or the RSF templates.

At this point the Sakai community led by Steven Githens of Northwestern University, Aaron Zekowksi of Virginia Tech, and others are evaluating early releases of RSF for its suitability for use within Sakai. RSF will only be adopted if the community collectively feels that it is a better framework and simply begins to use it instead of JSF.

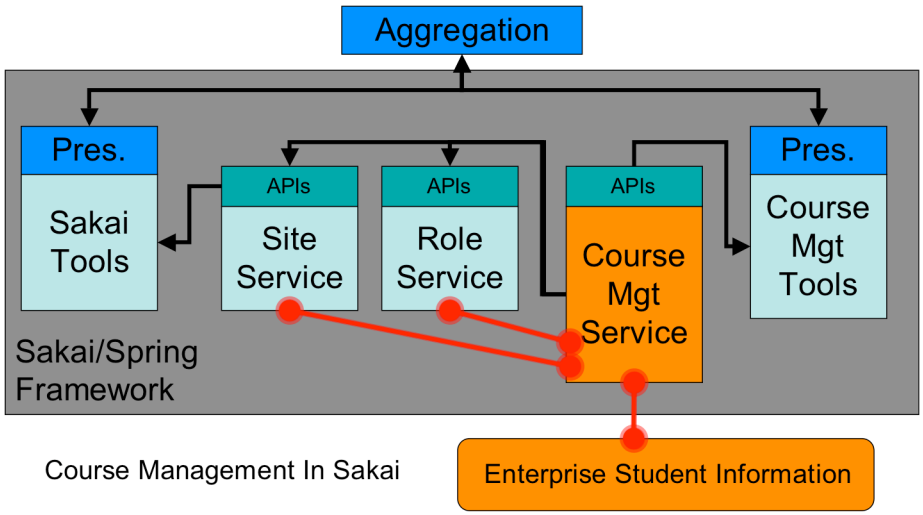
Because the Sakai Framework and Aggregation are agnostic with respect to a Tool's presentation layer, RSF tools can be mixed into Sakai along with the rest of the presentation technologies.

Course Management

Another of the areas where Sakai is still lacking is in support for data structures to represent course structure within Sakai. The CourseManagement Provider currently in Sakai only supports a small set of the real requirements in this area and a more comprehensive solution is needed to this problem.

There has been a long-term effort to define a CourseManagement Service and a set of tools around this service. The basic use case is to provide a service that can represent a very rich set of information that precisely describes course information. Another key use case of the CourseManagement Service is to provide a single entry point to integrate course information than then automatically populate all of the other important Sakai structure such as Sites, Groups, and Roles automatically from that information.

The architecture of CourseManagent is still being evolved but this diagram shows an architectural direction this effort may take:



Course Management In Sakai

The Course Management Service can also enable the development of a new set of tools can be to examine, and update this information as well as a new set of integration interfaces to allow institutions to populate very rich Course data into Sakai. In time, this could lead to increasingly supporting the basic use cases of a Student Information System through the development of specific CourseManagement Tools.

The leadership team for the CourseManagement effort is Marc Brierley of Stanford University, Ray Davis of UC Berkeley, Daisy Flemming of Stanford, Duffy Gilman of University of Arizona, Josh Holzman of UC Berkeley, Lydia Li of Stanford University, and Daphne Ogle of UC Berkeley. Mark Norton led a team to develop an initial Sakai CourseManagement API specification based on the OKI CourseManagement OSID.

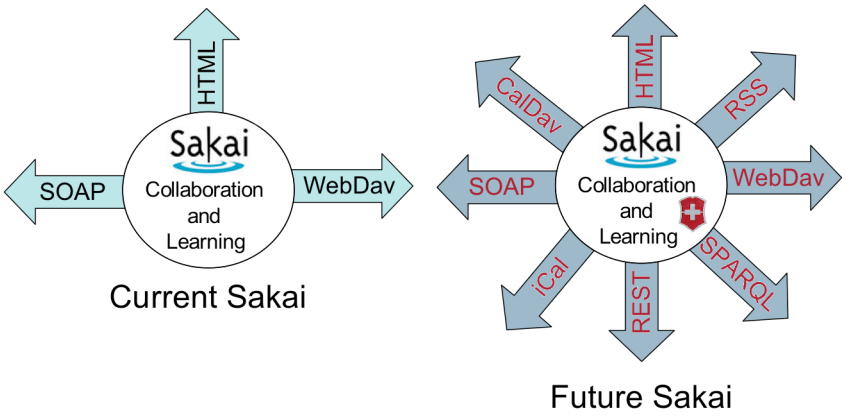
Web 2.0 and Service Oriented Architectures

While Sakai currently has excellent support for web services and can be part of an Enterprise's Service Oriented Architecture, Sakai must move toward being a fully service oriented architecture so as to allow elements of data and capability within Sakai to be reused and orchestrated by many of the other applications within an Enterprise to produce whole new applications for the users.

This requires additional web services support as well as a careful data modeling of all the elements within Sakai and the addition of ways of accessing the elements of Sakai directly through web services.

The ultimate goal is to allow the end-users to make use of a far wider range of software elements. Without rich data interoperability and data portability, users can only access the data and information stored in Sakai using the Sakai software. If the data can be easily moved between applications, then users can make use of their Sakai data with all of these applications.

Ultimately we need to evolve Sakai to the point where it is a veritable "Swiss-Army Knife" of data exchange protocols.

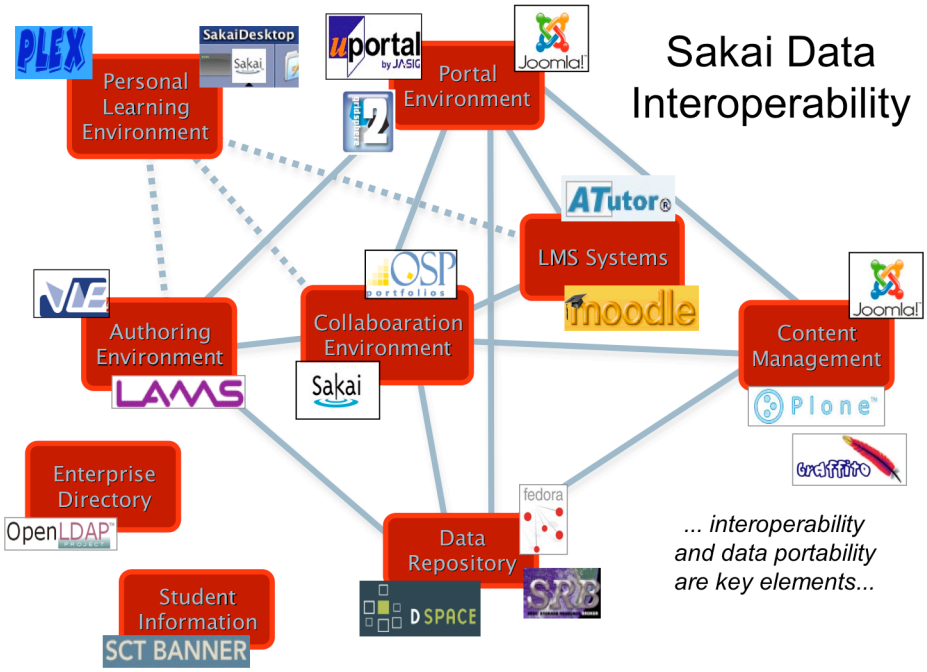


This section relates Sakai to the other applications that may be encountered in a typical Enterprise and explores the use cases enabled by allowing these other applications to access the data stored in Sakai.

In some situations, there will exist a rich set of enterprise applications already using a Service Oriented Architecture for integration with which Sakai will need to integrate. In other cases,

Sakai will be the core of a smaller enterprise and Sakai many pave the way for new enterprise applications and a service-oriented approach for these smaller organizations.

Sakai already is set up to integrate with Enterprise Directory and Student Information Systems using a service-oriented approach. Typically all of the Enterprise applications integrate with these directory and/or student information systems and so the relationships are not explicitly drawn for these two elements in the figure.



Ultimately as an organization assembles these applications they are creating what many call an "Enterprise Ecology" - this terminology is alluding to the idea that these applications operate in an environment with a rich set of data.

Architecture Summary

The Sakai Architecture is one of the keys to the current and future success of Sakai. To empower hundreds of highly talented developers and deployers around the word evolving a single product with a million lines of code together, it is very important to invest in a highly modular framework. A modular framework provides the necessary modularity and isolation to insure that the work of one developer does not inadvertently harm some other aspect of the system. This modularity leads to clean and universal support for web services across all of Sakai which allows Sakai to fully participate in an Enterprise's Service Oriented Architecture.

Prior to Sakai, the only framework that provided this type of modularity for the development of large applications is Enterprise Java Beans that are part of J2EE. EJB technology was evaluated at the beginning of the Sakai project and deemed to not be flexible enough to support the level of modularity and high levels of complex interactions between components that Sakai required to perform at the highest levels of scalability.

The decision was to go with the then emerging new technologies such as Spring and Hibernate which are loosely grouped together and called "J3EE". Sakai adopted, built on, and improved on these technologies to produce an architecture that is truly capable of supporting the needs of the Sakai community's development model.

By adopting and using an abstract architecture early in the project lifecycle and then evolving our framework against this architecture over time, we have produced extremely strong abstractions between, Aggregation, Presentation, Tool, and Service layers of the Sakai model. Sakai has used standards whenever possible at the layer boundaries and used industry standard capabilities whenever possible. But far too often the needs of Sakai outpace the capabilities provided by the current versions of standards such as JSR-168, WSRP, Spring, and others. When presented with these challenges, Sakai cannot wait for the standards process - world-class solutions must be produced and production ready in a four-month development cycle.

Even when the Sakai architecture must work beyond a standard, Sakai has a commitment to working to evolve those standards and then evolve the Sakai framework to make use of the improved standards that result from that work. Sakai is now heavily involved in IMS standards, the JSR-286 standard, and is working closely with portal developers to improve implementations for WSRP 1.0.

The decision making process for the presentation aspect and evaluation of RSF reflects a new approach that is being taken going forward with architecture decisions for Sakai. Instead of delegating technology choices to a "Chief Architect" or even a "committee of Architects", we involve the entire community in the evaluation of a new technology and we learn about a technology by actually using the technology in real applications over time rather than simply discussing technologies in meetings and then making a selection.

As of Sakai 2.0, the Sakai framework does not constrain the community in terms of choices - we can explore multiple alternatives at the same time, and then gravitate over time to the best technologies.

We see the Sakai framework as far more general that just a learning management system or even a collaborative application. We hope to continue to evolve the architecture, framework, and standards around tools and tool interoperability to ultimately produce an industry-wide consensus on how to build a general purpose standards-based eFramework for distributed application development. Sakai will not produce this eFramework alone. Sakai is simply one point solution exploring the problem space of the ultimate standards-based eFramework over the next 3-5 years.

Summary

Sakai has emerged as a critical component of the emerging model of infrastructure for higher education. This infrastructure is community source, based on a merging of institutional imperatives and the power of community distributed development. It is enterprise ready, capable of integrating with and running at scale in enterprise environments. It is simultaneously agile enough to incorporate the innovations that are necessary to the life of a vibrant academic community, innovations that are conceived and built by members of a globally distributed community. It is also focused on the extended set of scholarly activities, from teaching and learning, through research and collaboration with colleagues. This new infrastructure thus extends beyond the bounds of individual institutions and organizations and brings them into new relations with other members of the academy, with commercial organizations supporting open source, and with the global community of users and developers. It introduces those institutions to the emerging, open practices that the internet and the web have made possible as they drive communication and collaboration costs through the floor and enable new methods of constructing online educational tools and carrying on education itself.

The Sakai Community is pioneering this infrastructure and these new practices. The Mellon Foundation has helped bring this about, and in the process is helping higher education to meet the challenges of innovation that it faces as the world goes online.

Appendix A
Reports from Sakai Core Institutions

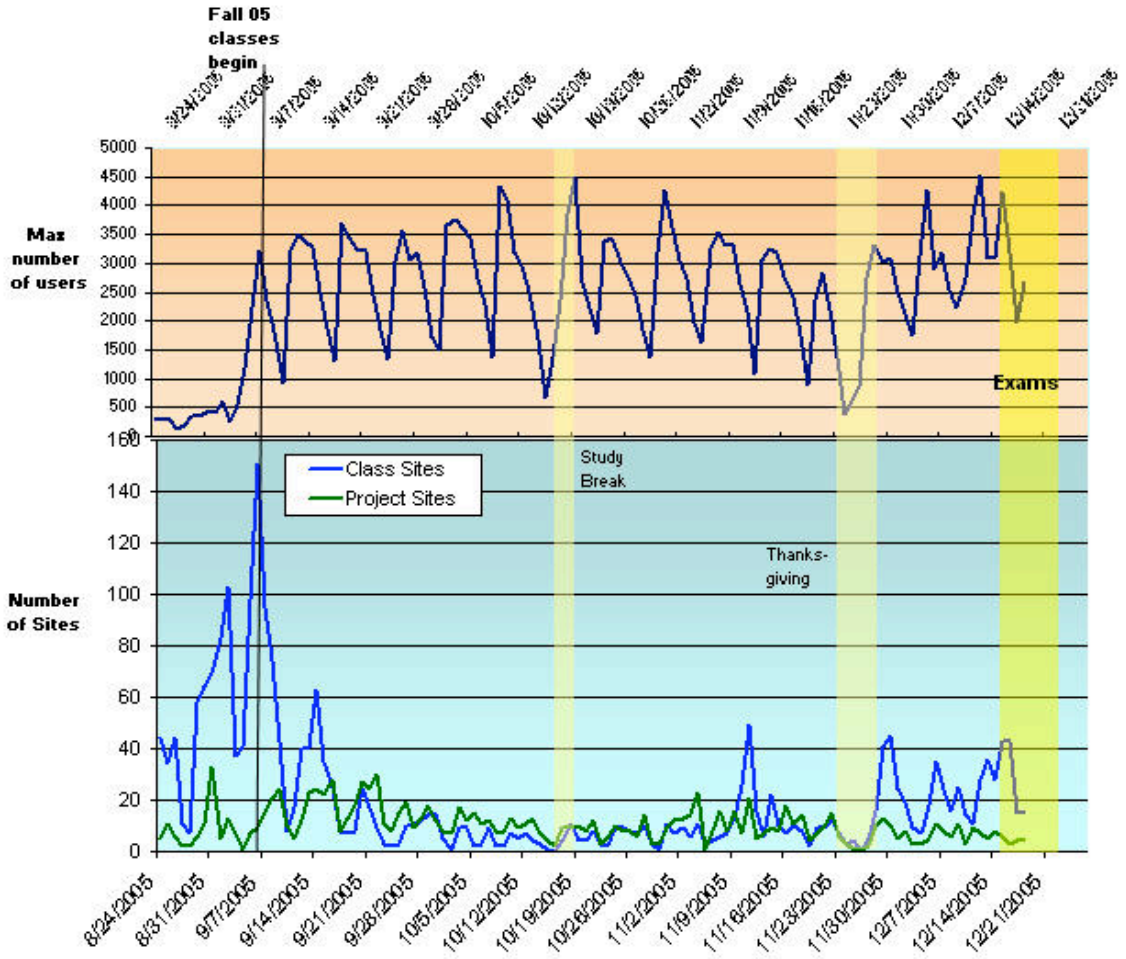
University of Michigan
The University of Michigan has completed a transition from its legacy CMS, UM Coursetools, a Lotus Domino-based system that was put into initial production in 1998 and since then had been supporting around 40,000 students and, and its legacy research support system, UM Worktools, which had been supporting over 10,000 University of Michigan and collaborating researchers for the previous 5 years. Both systems have been replaced by the Sakai-based UM Ctools system. The second year of the Sakai Project saw the turning off of the legacy systems. The University of Michigan is now running the Sakai software as its only production system for the support of teaching, learning, research and support of collaboration in all areas, including administration.

11,182 - Total class sites to date, all terms
7,667 - Total project sites
1,059 - GradTools student sites
71,746 - Total My Workspaces to date – total unique users of system

The University of Michigan has continued to upgrade during each academic year, and plans to move to the Sakai 2.2 codebase for the fall of 2006. UM Worktools support ceased in the winter semester of 2005, after all of its users were migrated to UM Ctools. Over the summer of 2005, the last of the 40,000 users of the legacy Coursetools system were moved over to Ctools, and we ceased support for UM Coursetools. In the fall of 2005, the University of Michigan started running only one Collaboration and Learning Environment for the support of online teaching and learning, online research and group work: the Sakai 2.0-based UM Ctools.

CTools Sakai Usage at UM – Fall '05

The graph below shows, at the top, the number of concurrent (simultaneous) users on the system. The regular rises and falls reflect weekly usage patterns.
The second part of the graph shows site creation statistics, showing the surge of new site creation for classes at the beginning, and ends, of the semester, and the relatively steady creation of project sites throughout the semester. UM CTools supports around 3000 class sites per semester now, in addition to the thousands of project sites.



UM Grad Tools

The past two years have also see the expansion of CTools on the University of Michigan campus with the development of Grad Tools, a tool now a part of the Sakai Community Development Process. Grad Tools development has been funded by the University of Michigan Rackham School of Graduate Studies, which oversees all graduate degree development at UM. It is a good example of how local development, placed in the shared framework of the Sakai architecture, can be shared and leveraged by other schools.

From the Rackham site (http://gradtools.umich.edu/):

The Rackham School of Graduate Studies at University of Michigan has assembled a special set of tools in the CTools environment to help Rackham doctoral students as they work toward their degrees. These tools are known collectively as Grad Tools. The unique feature of Grad Tools is the Dissertation Checklist, which presents the process for completing the doctoral degree in one personalized view.

The Dissertation Checklist in a student's Grad Tools site automatically inherits the required steps from the Graduate School and from the student's department. It may also include steps added by the student and by his or her committee members.

In addition to the Dissertation Checklist, Grad Tools provides several tools common to CTools, including a schedule, discussion feature, a place to store forms and documents, and more.

SakaiBrary
The University of Michigan is also working with Indiana University and others on the development of expanded access to library holdings through the Mellon-funded SakaiBrary Project. See (http://www.dlib.indiana.edu/projects/sakai/)

Michigan Contribution
Throughout the life of the Mellon grant the University of Michigan committed resources that exceeded those called for in the grant agreement. As we move on, this level of commitment to the Sakai effort has continued, and is even increasing. As shown by the fact that this has become a key production infrastructure component, and indeed because of that fact, the importance the University gives this work is extremely high.

Indiana University Sakai Deployment

Indiana University Participation in Sakai
Indiana University readily joined with the University of Michigan, MIT, Stanford, the Open Knowledge Initiative (OKI), and uPortal to form the community dedicated to developing an open source Collaboration and Learning environment. A generous $2.3M gift from the Andrew Mellon Foundation in December 2003 launched the Sakai Project.

Indiana University contributed eight developers or roughly $600K in resources to the development efforts. The Sakai Project produced three software releases in 2005—1.0, 1.5 and 2.0. Release 2.1 was available in 2006 and marked a significant milestone in the product and provided the basis for IU to begin total migration to Sakai for its Course Management and Collaboration tools. The subsequent 2.1.1 and 2.1.2 releases were made available in the first and second quarters of 2006 constituting hundreds of enhancements and fixes.

Indiana University Implementation
IU used the name of its original course management system, Oncourse, to brand its Sakai implementation. IU introduced Sakai Version 2.0 as Oncourse CL (Collaboration and Learning) in June 2005. Oncourse CL became the default course management system in fall of 2005 as IU created sites for all of its 62,000 course sections. Through spring 2007, faculty can opt out of the Sakai based system and continue to use original Oncourse allowing more time for transition, but usage statistics demonstrate steady adoption of the new system. In fall 2005, roughly 29% of users were regularly utilizing Oncourse CL for courses, project collaboration, and/or gaining familiarity with the new system during this transition year. The use of Oncourse CL has grown to over 39% in the Spring 2006 semester as it gains traction with faculty and students. Over 109,000 unique users have used Oncourse CL in the current semester to gain access to one of its 64,000 sites at least

once. During the month of February 2006, Oncourse averaged 3.8M hits and 115,000 page views per day.

The 2.1 release of Sakai was a large step in implementing Oncourse CL. It included enabling features required by IU. As IU integrated this release it also developed and deployed both the IU Post'Em tool for complex grade book problems and general feedback created in a spreadsheet format and the new Message Center tool for discussion forums and private messaging. Message Center supports tracking of read and unread messages, posting discussion grades and showing user statistics of posts and read messages. The Collaboration components of Sakai are essential to IU's strategy for classroom and research collaboration and have been widely adopted as Oncourse CL now houses over 1,000 project sites in addition to the traditional class space.

Sakai software is also at the center of IU's strategy for storage. The Sakai Resource tool is a component of the software to store and manage documents such as spreadsheets, word-processing, video and audio files. IU implemented the new Resource Tool file system for Oncourse CL that is tied ubiquitously to Open Source Portfolios (OSP), other Sakai tools, and all sites within Oncourse CL. The consolidated file space in Oncourse CL replaces two previous file storage systems. We now have 350GB of files created from the resource tool. IU is also strongly considering leveraging this investment in Sakai's Resources infrastructure as the foundation of its podcasting initiatives.

IU has made a significant investment in personnel and capital to support the Sakai product. The University uses state-of-the art software and hardware to provide a robust infrastructure. The combination of solid process and infrastructure allows Oncourse CL users to access the system nearly 24 hours per day and 365 days per year.

IU Implementation Plan for 2006-2007

IU is committed to making Sakai its only centrally supported Course Management environment by Summer 2007. The primary focus of 2006 is to assist the faculty and students in making the transition. Extending a model used for the original Oncourse development, the Office of the Vice President for Information Technology created a faculty Oncourse Priorities Committee to set current and future directions for Oncourse CL. This faculty committee was charged in May 2005 to provide guidance for developing new Oncourse CL features and enhancements. A related committee of faculty and staff from the teaching and learning centers and libraries across the IU system, the Functional Requirements Committee, assists in interpreting user needs and translating them into specific functional requirements.

IU's primary IT goal for 2006 is supporting the transition of its 109,000 users to Oncourse CL through classroom and online training, visits to regional campuses, online materials and documentation, and one-on-one faculty consultation, in supporting Oncourse CL, ePortfolio, and new research and collaboration tools. IU has created nearly 300 Knowledge Base documents and developed more than 45 feature demonstration animations. The teaching and learning centers provided over 130 workshops,

presentations, training sessions, and conducted more than 1,000 consultations with over 3,300 contacts related to Oncourse CL.

Future Improvements

Oncourse CL will become the nerve center for pedagogy and research collaboration at Indiana University. Extensions to the system and new tools will improve its usefulness and increases its richness as a learning tool. One of the most important aspects of future enhancements is the integration with Library services.

In 2005, staff from the Digital Library Program and IUB Libraries participated in a series of meetings sponsored by the Andrew W. Mellon Foundation to discuss issues of integration of library services and content within the Sakai collaboration and learning environment. These discussions are anticipated to result in a collaborative project beginning in 2006 to develop tools for faculty and student use of library resources within the Sakai environment and within IU's implementation of Sakai - Oncourse CL. These software tools will be based in part on the Twin Peaks tool developed by staff in IUB Library Information Technology to enable faculty to search for and link to library content from within the Sakai authoring environment; see https://twinpeaks.dev.java.net/.

Summary

Indiana University's commitment to the Sakai project is unwavering. Sakai 2.1.1, named Oncourse CL, is in full deployment at IU and it will become the only centrally managed course management solution for University in 2007. In fall 2005, IU began automatically loading courses into Oncourse CL; however, through spring semester of the 2006-2007 academic year, faculty may choose to opt out and use the original Oncourse. After spring 2006, the Oncourse CL Resourses tool will become the primary storage solution. After spring 2007, original Oncourse will be fully retired. In addition, Oncourse CL is a strategic collaboration tool for research and administrative uses.

Oncourse CL statistics as of April 6, 2006:

Total sites since full production release	64,024
Total project and practice course sites created	2,034
Distinct users with access to sites	122,522
Distinct users who have accessed the system (89%)	109,573
Total Gigabytes used in Oncourse CL file space	350

In response to local needs, Indiana University has designed and developed additional tools provided provisionally to the Sakai community:

- A Roster tool provides instructors and site owner access to official ID photos of students and university employees.
- The Message Center tool includes both discussion forums and private messaging.
- Post'Em provides the capability to share the information in Excel-based gradebook and feedback files to site participants.

IU continues its active participation in the community development effort with a strong emphasis on Library integration.

Stanford University
(Lois Brooks)

Summary
Stanford entered the Sakai project to fulfill several goals:

- ♣ Support the demand by faculty for innovative, new tools for collaboration and teaching
- ♣ Develop a sustainable model for cooperative software development in higher education that would allow for shared tools and best practices

These goals have been met. The Sakai code is in active use at Stanford with full campus deployment imminent, and the community of developers is active, productive and sustained.

The Sakai model is extremely attractive in that the combined resources of the participating schools can produce a far more complete and robust system than any single campus could develop, while development costs can be distributed among several campuses. Stanford's engagement assured that our particular needs would be addressed, and it would also assure that our acknowledged strengths will be brought to bear on the shared product. In reflection at the end of the project, we believe that we have contributed substantively to the success of Sakai. We also believe that Stanford is better off for its participation because of the opportunity to reap the tangible benefits of a large and robust code base, and intangible benefits of new technical and collaboration skills gained by our staff through work on this project.

Background
Since its rollout in Winter 2001/2002, CourseWork (Stanford's legacy homegrown course system) had experienced runaway success. During the last academic year, over 2500 classes used CourseWork, representing over 100,000 students and faculty in class (bodies in seats). (This is a ratio of 7 courses per Stanford student, per academic year.) This overwhelming adoption has clearly demonstrated the usefulness and timeliness of course software development. The availability of technology has sparked keen interest by faculty to extend CourseWork's functions to better meet their teaching needs, and to extend the ways they use technology in teaching. Stanford had developed faculty-driven features such as special support for languages and large lecture classes that spurred adoption and became critical to the teaching programs. Professor Elizabeth Bernhardt, Director of the Stanford Language Center, summed the dependence of teaching programs on CourseWork, "Let me be brief: the language programs would grind to a technical halt without Coursework. …a significant undergraduate requirement is linked to the effective and efficient functioning of the Coursework system." Similar enthusiasm has been voiced

by the Human Biology Program, which uses CourseWork to support large lecture classes. Feedback from faculty clearly affirms that:

1) A CMS is important to Stanford faculty and operations.
2) Meeting very specific (and diverging) faculty needs is critical to the success of the system.
3) It fosters faculty productivity; and
4) By inference, active engagement by Stanford in future development of the campus CMS is valuable and, possibly, mission critical.

The trends of widespread adoption and faculty demand for new capabilities dictated that we continue to develop course software, but the economics were unsustainable. The costs of maintaining a large code base, innovating to meet faculty needs, and migrating to new technology standards as they emerged were outside the scope of what was feasible.

Outcomes

The Sakai project has successfully met Stanford's goal of developing course software, albeit with some localization to be accomplished (described below). The Sakai code set offers most of Stanford's required capabilities, and many features that provide added value and were not available in the legacy CourseWork deployment. For example, WebDav support, personalized workspaces, WYSIWYG editing, and the ability to add customized navigation buttons to pages are new to Stanford. These were not required, but are warmly welcomed by Stanford faculty. Many of the features in Samigo Quiz and Test, funded separately by Mellon under the name Assignment and Assessment Manager, were designed to meet Stanford's growing needs for rich formative testing and assessment tools. These have been augmented by the summative assessment tools desired by Indiana University, again new to Stanford, that provide additional and useful capability.

We are currently in pilot, with a staged migration planned for Academic Year 06-07. Our stance on system deployment is risk averse; we carefully test and pilot software before making it widely available. Hence, deployment has been slower than at Michigan and Indiana, both of whom are in full deployment. However, pilots in the previous year have been successfully completed, and Sakai (under the local brand CourseWork v5) will deploy during the coming academic year.

There are two areas development required before Stanford can fully deploy; we are actively working on both. First, support for voice-based questions and responses in Samigo Quiz and Test must be added. This functionality is available in our current system and is used extensively by language classes. This code is almost complete. Once complete, it will be put into use at Stanford and committed to Sakai for others to use. Second, Course Management functionality is needed to support Stanford's complex system of classes, sections and labs. A team from Stanford, UC Berkeley, UC Davis and the University of Arizona are working together to fulfill the following goals:

- ♣ Create a single, decoupled, and non-legacy Course Management API within the Sakai Application Framework (SAF) for improved enterprise integration with Student Information Systems (SISs).
- ♣ Provide a reliable Course Management Service upon which Sakai tools can keep data in synch with underlying course information.
- ♣ Build an enhanced Course Management user interface for the administration of sites based on user-centered design. This tool(s) will have functionality designed to meet verified requirements of the Sakai community.

This functionality is a requirement for Stanford deployment to large lecture classes. Once completed, our deployment will quickly follow.

We are particularly interested in Sakai's ability to be extended beyond courses. While our local deployment will focus on courses to start, we intend to leverage Sakai's portfolio, research collaboration, and grad tools capabilities in the near future. Sakai is intended to be an academic framework at Stanford, and we are currently investigating its interoperability with library tools and uPortal, as well as extended toolsets, for wider deployment.

The Sakai project has also met Stanford's goal to develop a sustainable model for software development in higher education. Even as the Sakai project winds down and the Foundation takes over, we are actively engaged with partner schools on additional projects related to Sakai (like Course Management, mentioned above) and planning more local development. We remain committed to the continued success of the Sakai community.

Contributions

As committed in the original grant proposal, Stanford contributed 7 FTE dedicated to Sakai (5 FTE developers, 2 FTE functional design and interoperability). These resources were dedicated from January 2004 through January 2006. The work products of these seven staff include:

- ♣ development of the Samigo Quiz and Test engine (with Indiana University)
- ♣ interoperability demonstrations for QTI functionality
- ♣ development of Course Management specifications, API and related application coding
- ♣ development of the Sakai JSF module (components and utilities that can be shared by JSF based Sakai tools)
- ♣ user guide and demonstration application for JSF
- ♣ Other programming as requested by Severance/Golden, including Message of the Day tool
- ♣ Extensive best-of-breed analysis, functional design and specifications for core Sakai code and tools

In addition to the dedicated resources, Stanford has done extensive QA on the project, and has participated on the board and in Sakai community development efforts that are common to all the core contributors.

Dedicated development, design and QA resources contributed by Stanford total over $1.5 million, exclusive of local deployment, management, and board participation. Local investments and board participation bring the Stanford Sakai total to well over $2 million.

MIT
Under development – see Babi Mitra, MIT

Appendix B

Sakai Worldwide Deployments List

Below is a listing of known Sakai deployments worldwide. If available, additional deployment information can be reviewed by clicking on the institution's or organization's name. For a completely current list with additional information, including the operating system and database in use at the sites, and contacts for known sites, see http://bugs.sakaiproject.org/confluence/display/PROD/Implementations
This site was created and is maintained by Anthony Whyte (http://bugs.sakaiproject.org/confluence/display/~arwhyte), Sakai Foundation staff.

Region	Country	Organization	Type	Status	Production Start (YY-MM)	Sakai ver.	Site
Africa	EG	Cairo University, Faculty of Engineering	U	Production	2005	2.0.1	HEEPF
Africa	ZA	University of Cape Town	U	Production	2006-02	2.1.0	Vula
Africa	ZA	University of South Africa	U	Production	2006-01	2.0.1	MyUnisa
Asia-Pacific	AU	Australian National University	U	Demo	TBD	2.1.0	http://sakai.anu.edu.au/
Asia-Pacific	AU	Charles Sturt University	U	Production	2006-04	2.1.1	http://interact.csu.edu.au
Asia-Pacific	HK	Hong Kong University of Science and Technology	U	Production	2005	2.0.1	LMES
Asia-Pacific	JP	Hosei University, ITRC	U	TBD	TBD		
Asia-Pacific	JP	Nagoya University	U	TBD	TB		

Asia-Pacific	JP	ULAN Project	P	Production	2005		http://sakai.ulan.jp/portal/
Asia-Pacific	NZ	University of Auckland	U	TBD	TBD		
Europe	BE	Hogeschool Gent (Associatie Universiteit Ghent)	U	Production	2005	1.5.1	http://portfolio.hogent.be/pc
Europe	DE	Lubeck University of Applied Sciences	U	Pilot	2006-09	2.1.1	ONCAMPUS
Europe	DE	Universität Bremen, TZI	U	TBD	TBD		
Europe	DK	Roskilde Universitetscenter	U	Production	2005	2.0.1	eCampus
Europe	EI	University of Limerick	U	TBD	TBD		
Europe	EI	University College, Dublin	C	TBD	TBD		
Europe	ES	Universidad de Vigo	U	TBD	TBD		
Europe	ES	Universidad Politécnica de Valencia	U	Pilot	2006-09	2.1.2	https://poliformat.upv.es/po
Europe	ES	Universitat de Lleida	U	Production	2005-08	2.1.2	http://cvirtual.udl.es/portal
Europe	IT	Università degli Studi di Salerno	U	Production	2004	2.1.1	http://rp.csedu.unisa.it/porta
Europe	NL	Universiteit Twente	U	Pilot (Fall '06)	TBD		CBUS

Europe	NL	Universiteit van Amsterdam, SURF	U	Pilot	TBD		http://sakai.ic.uva.nl/tempor unavailable
Europe	NO	Universitetet I Oslo	U	TBD	TBD		
Europe	PT	Universidade Fernando Pessoa	U	Production	2004-10	2.1.2	https://elearning.ufp.pt/porta
Europe	SE	Stockholms universitet	U	Pilot	TBD	2.1.0	SU Groups
Europe	SE	Umeå universitet	U	TBD	TBD	2.1.0	Informatik
Europe	UK	Daresbury Laboratory, CCLRC	L	TBD	TBD	2.1.1	http://rhine.dl.ac.uk:8080/pc
Europe	UK	Glasgow Caledonian University	U	Demo	TBD	2.1.0	http://blueridge.gcal.ac.uk/p
Europe	UK	Lancaster University	U	Pilot	TBD		
Europe	UK	University of Cambridge	U	Pilot	2006-10	2.1.2	CamTools
Europe	UK	University of Cambridge, TLRP	P	Production	2005	2.0.1	TLRP VRE
Europe	UK	University of East Anglia	U	Pilot	TBD	2.1.0	http://vre.his.uea.ac.uk/porta
Europe	UK	University of Hull	U	TBD	TBD	2.1.0	

Europe	UK	University of Hull	U	TBD	TBD	2.1.0	
							http://vre.his.uea.ac.uk/porta
Europe	UK	University of Portsmouth	U	TBD	TBD	1.5.1	OSP Portfolios
Europe	UK	University of Strathclyde, AERS	P	Production	2005	2.1.2	AERS VRE
C. & S. America	CO	La Institución Universitaria CEIPA	U	Production	2005	2.1.1	La Universidad de la Empresa
C. & S. America	GT	Universidad del Valle de Guatemala	U	TBD	TBD	2.1.0	http://sakai.uvg.edu.gt/porta
N. America	CA	University of Alberta	U	Pilot	TBD	2.1.0	https://sakai.educ.ualberta.c http://temp28-2.cs.ualberta.ca:8080/portal
N. America	CA	University of British Columbia	U	TBD	TBD		
N. America	CA	University of Windsor	U	Pilot (F '06)	2007-09		
N. America	CA	University of Winnipeg	U	Demo	TBD	2.1.1	LEARN
N.							

N. America	US	Albany Medical College	C	Production	2005		
N. America	US	Alliance for Equity in Higher Education	NP	Production	2005	2.0.1	WebCenter
N. America	US	Appalachian College Association	NP	Production	2006	2.1.2	LAMP
N. America	US	Apple, Inc.	B	Production	2005	2.1.2	MacLearningEnvironments
N. America	US	Arizona State University	U	Pilot	TBD	2.1.2	https://sakai.asu.edu
N. America	US	Boston University, School of Management	U	Pilot	2006-09	2.2.0	http://smgtools.bu.edu
N. America	US	California State University, Fresno	U	Demo	TBD		Sakai Demo, OSP Demo
N. America	US	Cerritos College	CC	Pilot	TBD	2.1.1	TalonNet SE
N. America	US	Chariho Regional School District (Charlestown, Richmond, Hopkinton, RI)	K12	Production	2005	1.5.1	Chariho Portfolios
N. America	US	Claremont Colleges Consortium	C	TBD	TBD	2.1.1	http://sakai.claremont.edu:8

N. America	US	Coastline Community College	CC	TBD	TBD		
N. America	US	Community College of Southern Nevada	CC	TBD	TBD	2.0.1	http://sakai.ccsn.edu/portal
N. America	US	Columbia University	U	Pilot	TBD	2.0.1	http://sakaipilot.cc.columbia
N. America	US	Dana College	C	TBD	TBD	2.1.0	Sakai.Dana
N. America	US	ETUDES Alliance	CC	Pilot	2006-07	2.1.2	ETUDES-NG
N. America	US	Foothill College	CC	Production	2005	2.1.2	ETUDES-NG
N. America	US	Franklin & Marshall University	U	Pilot	TBD		http://sakai.fandm.edu/porta
N. America	US	Hebrew Union College	C	Production	TBD		
N. America	US	Hiram College	C	TBD	TBD	2.0.1	https://courses.hiram.edu/pc
N. America	US	IDEAL (Arizona Dept. of Education)	K12	Production	2005		https://sapp.ideal.azed.gov/p
N. America	US	Indiana University	U	Production	2005-01	2.2.0	OnCourse
N. America	US	Johns Hopkins University	U	TBD	TBD		
N. America	US	Kapi'oloni Community College (Univ. of	CC	Production	2005	1.5.1	

N. America	US	Kapi'oloni Community College (Univ. of Hawaii)	CC	Production	2005	1.5.1	OSP Portfolios
N. America	US	Longsight Group	B	Production Demo	2005	2.1.0	http://sakai.longsight.com/p
N. America	US	Lake Erie College	C	Demo	TBD	2.1.0	iLearn
N. America	US	Moody Bible Institute	C	Pilot	TBD		
N. America	US	Mount Holyoke College	C	Demo	TBD	2.1.0	http://luna.mtholyoke.edu:8
N. America	US	New York University	U	Pilot	TBD	2.0.0	http://bacon.cs.nyu.edu/port
N. America	US	Northwestern University	U	Pilot	TBD	2.1.2	https://arch.northwestern.ed
N. America	US	Ohio State University	U	Production	2005	1.5.1	OSP Portfolios
N. America	US	Ohio University	U	Pilot	TBD		
N. America	US	Open Source Portfolio Initiative (OSPI)	NP	Production Demo	2005	1.5.1	http://demo.osportfolio.org/
N. America	US	Oxnard College	CC	TBD	TBD	2.0.1	http://sakai.oxnardcollege.e
N.		Pacific Lutheran					

N. America	US	Portland State University	U	Pilot	2006-09		Samla
N. America	US	Rice University	U	Pilot	TBD	2.1.1	Owl-Space CCM
N. America	US	Ringling School of Art and Design	C	Pilot	TBD	2.0.0	https://ontario.rsad.edu/port
N. America	US	The rSmart Group	B	Production Demo	2004	2.1.2	Sakai Sandbox
N. America	US	Rutgers University	U	TBD	2007-09	2.1.1	https://sakai.rutgers.edu/por
N. America	US	Saginaw Valley State University	U	Pilot	TBD	2.1.2	http://www.svsu.edu/ols/, http://sakai.svsu.edu/portal/
N. America	US	Sakai Foundation	NP	Production	2004	2.1.2	Collab
N. America	US	San Mateo County Community College District	CC	Pilot	TBD	2.0.1	http://smcweb1.smccd.net:8
N. America	US	Stanford University	U	Pilot	2006-09	2.1.0	Coursework
N. America	US	Texas State University, San Marcos	U	Pilot	2006-08	2.1.1	TRACS
N. America	US	Trinity University	U	Demo	TBD	2.1.0	https://sakai.trinity.edu/

N. America	US	UNC Teaching and Learning with Technology Collaborative	U	Pilot	TBD	1.5.1	OSP Portfolios
N. America	US	Unicon	B	Production Demo	2005	2.1.0	Sakai Test Drive Site
N. America	US	University of California, Berkeley	U	Pilot	2006-09	2.1.0	bspace
N. America	US	University of California, Davis	U	Pilot	TBD		
N. America	US	University of California, Los Angeles	U	Pilot	TBD	2.1.1	http://sakai.ucla.edu/portal/
N. America	US	University of California, Merced	U	Production	2005-08	2.0.1	UCMCROPS
N. America	US	University of California, Santa Barbara	U	Pilot (Fall 06)	TBD		
N. America	US	University of Illinois at Urbana-Champaign	U	Pilot	TBD	2.1.0	http://sakai.lis.uiuc.edu/port
N. America	US	University of Michigan	U	Production	2004-08	2.1.2	CTOOLS
N. America	US	University of Missouri	U	Pilot	TBD	2.1.0	https://sakai.missouri.edu/p
N. America	US	University of Nebraska-Lincoln	U	Pilot	2006-05	2.1.2	http://sakai.unl.edu:8080
N. America	US	University of North Texas	U	TBD	TBD		

N. America	US	University of the Pacific	U	TBD	TBD	2.0.0	http://sakai.pacific.edu/porta
N. America	US	University of Virginia	U	Pilot	TBD		
N. America	US	Virginia Tech	U	TBD	TBD	2.1.0	Scholar
N. America	US	Walsh University	U	Production	2005-08	2.1.2	http://sakai.walsh.edu/portal
N. America	US	Weber State University	U	Production	2005-12	2.1.2	http://sakai.weber.edu
N. America	US	Western Oregon University	U	Demo	TBD	2.0.0	http://merlin.wou.edu:8080/
N. America	US	Whitman College	C	Pilot	2006-09	2.0.0	CLEo
N. America	US	Wofford College	C	TBD	TBD	2.0.1	http://sakai.wofford.edu/por
N. America	US	Yale University	U	Production	TBD	2.1.2	Classes*v2

Appendix C

Sakai Partners

Official Partner List - 06/05/06

Count	Institutional Partners
1	Albany Medical College
2	Arizona State University
3	Australian National University
4	Boston University School of Management
5	Brown University
6	California State University, Office of the Chancellor
7	Carleton College
8	Carnegie Foundation
9	Carnegie Mellon University
10	Cerritos Community College
11	Charles Sturt University
12	Coast Community College District
13	Columbia University
14	Cornell University
15	Dartmouth College
16	Edgenics
17	Florida Community College at Jacksonville
18	Foothill College
19	Franklin University
20	Georgetown University
21	Georgia Tech
22	Harvard University
23	Hosei University
24	Indiana University
25	Johns Hopkins University
26	Lancaster University
27	Louisiana State University
28	Loyola University, Chicago
29	Luebeck University of Applied Sciences
30	Maricopa County Community College District
31	Marist College

32	Massachusetts Inst. Of Techology
33	Michigan State University
34	Monash University
35	Nagoya University
36	New York University
37	Northeastern University
38	North-West University
39	Northwestern University
40	Ohio Learning Network
41	Ohio State University
42	Pennsylvania State University
43	Portland State University
44	Princeton University
45	Rice University
46	Ringling School of Art and Design
47	Roskilde University
48	Rutgers University
49	Sakai Quebec
50	Simon Fraser University
51	Stanford University
52	State University of New York System
53	Stockholm University
54	SURF/University of Amsterdam
55	Syracuse University
56	Texas State University - San Marcos
57	Tufts University
58	Tulane University
59	Universidad Politecnica de Valencia
60	Universitate de Lleida
61	University College Dublin
62	University of Arizona
63	University of Auckland
64	University of British Columbia
65	University of California, Berkeley
66	University of California, Davis
67	University of California, Los Angeles
68	University of California, Merced
69	University of California, Santa Barbara
70	University of Cambridge, CARET
71	University of Cape Town
72	University of Chicago

73	University of Colorado at Boulder
74	University of Delaware
75	University of Hawaii
76	University of Hull
77	University of Illinois at Urbana-Campaign
78	University of Limerick
79	University of Melbourne
80	University of Michigan
81	University of Minnesota
82	University of Missouri
83	University of Nebraska
84	University of North Texas
85	University of Oslo
86	University of South Africa UNISA
87	University of Texas at Austin
88	University of Toronto
89	University of Twente
90	University of Virginia
91	University of Washington
92	University of Winnipeg
93	University of Wisconsin, Madison
94	Virginia Tech
95	Weber State University
96	Whitman College
97	Yale University

Corporate Partners

1 Apple Computer
2 Embanet
3 HarvestRoad, Ltd.
4 IBM
5 Oracle
6 Ostrakon, Ltd.
7 Pearson Higher Ed
8 the rSmart group
9 Stoas
10 Sun Microsystems
11 Sungard SCT
12 Unicon
13 Unisys

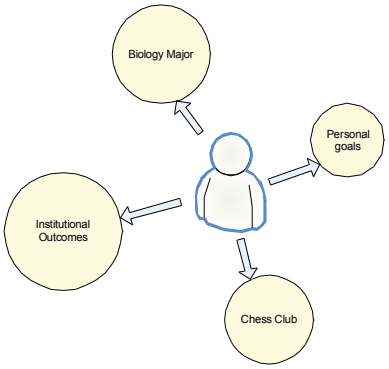
Appendix D

The OSPI Decision to Build on Sakai

The Open Source Portfolio (OSP) version 2: Built on the Sakai framework

John Ellis, OSP2 Development team, the r-smart group
Chris Coppola, OSPI Board, the r-smart group

Introduction



The Open Source Portfolio (OSP) version 2 is based on the idea that a user's interaction with the portfolio system is uniquely characterized by the groups they belong to. A user can belong to any number of common interest groups that define the user's experience as a member of that group. A common interest group defines templates, organizational structures, scaffolding to instruct and guide a user, and a set of tools to facilitate interaction, information retrieval, and more.
OSPI's desire to organize the software around groups of people with a common interest and purpose appeared to be very similar to Sakai's organization of collaboration and learning tools around course and project sites (worksites). In July of 2004, the OSPI leadership evaluated the Sakai technology as a means to provide some of the OSP 2 infrastructure and application services. We did this with a number of things in mind:

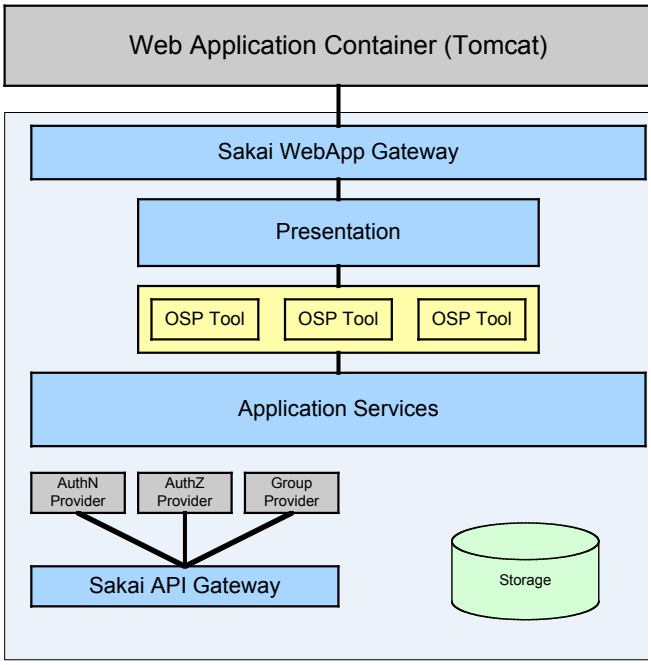
1. Using existing Sakai infrastructure could minimize duplication of effort between the projects and allow the OSP 2 development team to dedicate more energy to those things unique to the needs of portfolio.
2. Sharing code between the two projects will extend the number of developers familiar with the code of both projects and improve innovation potential & sustainability.
3. Both projects would benefit from the functional capabilities of the other to provide the respective communities with a more flexible, and more capable set of eLearning tools.
4. Because eLearning tools like Sakai are already in widespread enterprise use, Sakai would be required to perform under greater load, in more environments, and be subject to greater use. These conditions would lead to a more scalable and reliable application that OSP could leverage for it's own performance and reliability.

The evaluation in July resulted in a decision to build OSP 2 on the Sakai framework. This whitepaper explains what that decision means.

Portfolio tools running in the Sakai framework

The OSP software, like most enterprise applications, makes use of dozens of other libraries, frameworks, and tools. Leveraging existing software provides OSP with proven solutions to common application needs like data persistence, xml handling, and more. This approach also leaves more project resources to focus on the features and functions unique to the electronic portfolio. Similarly, OSP utilizes the Sakai framework to provide much of the application infrastructure and some features and functionality common to many learning-focused applications.

Sakai (www.sakaiproject.org) is really two things. It is an enterprise application that provides organized online space and tools for a host of collaborative activities like learning, teaching, and research—a collaboration and learning environment (CLE). It is also a framework that defines a set of APIs for tool development and portability—the Sakai Tool Portability Profile (TPP). OSP is designed as a collection of portfolio tools written to the TPP to run in the Sakai environment. This design provides OSP with many common application functions for user and group management, authentication, authorization, and more. It has also greatly expanded the set of tools available for use in a portfolio Common Interest Group. In addition to the portfolio-specific tools like the repository, the learning matrix, portfolio presentation and sharing tools, a portfolio Common Interest Group can leverage Sakai's collaboration tools like chat and discussion forums.



Tool portability is the focus of an emerging IMS specification that will make tools portable across many learning environments like Sakai, Blackboard, WebCT, and any other applications that support the specification. Sakai is playing a lead role in the new IMS specification. The OSP 2 design anticipates the evolution of these standards which will allow OSP 2 tools to run in any system that implements the specification. While the specification is still developing, one could implement the Sakai Services APIs to integrate OSP 2 tools with any LMS with some development effort.

The flexibility to be just what each institution needs

While OSP leverages the Sakai framework to create a robust enterprise portfolio system, it is important to note that OSP does not depend on the Sakai course, research, and collaboration tools. OSP does make use of common application services of the Sakai framework through the Sakai API's. In a similar way, OSP leverages many other open source projects such as the Java Standard Tag Library (JSTL) and Hibernate for object relational mapping. Though OSP does not depend on the Sakai collaboration and learning tools, some institutions may choose to take advantage of them. The flexibility to tailor the

OSP for a variety of institutional conditions makes it very powerful. There are three different approaches to implementing the OSP:

1. **An enterprise portfolio system.** The OSP can be implemented as simply an enterprise portfolio system without the features and functionality that makes the Sakai CLE function like a Learning Management System (LMS).
2. **An integrated learning environment with Sakai.** The OSP can also be implemented in a way that exposes the CLE features of Sakai such as quizzing, testing, assignment drop box, calendar, and the like to create a 'next generation' integrated learning environment.
3. **An integrated learning environment with a proprietary LMS.** With some development effort, the OSP tools may be integrated with an LMS like WebCT or Blackboard in a similar way to the pre-integrated Sakai environment.

The default OSP installation will leverage Sakai's "Quick Start" installation and will be preconfigured to expose just those features that make OSP a world-class enterprise portfolio system. Extending the basic installation to enable an integrated learning environment like that described in scenario two above is as simple as enabling those features using web-based configuration tools.
Schools that are already running Sakai will be able to quickly leverage the OSP tools without the overhead of creating or configuring new authentication mechanisms and other points of integration with systems on campus. Because OSP tools are built to use the common Sakai framework features like authentication (authN) and authorization (authZ), the OSP tools will just work.
Likewise, schools that are planning to implement OSP without the Sakai LMS functionality will still benefit from Sakai's scalable and reliable framework with application services such as AuthN and AuthZ.

The Bottom Line
The OSPI leadership is extremely excited about the benefits of leveraging the Sakai framework. Since we don't have to build many of the common application services already provided in the framework, the development and functional teams are able to focus on tools specific to electronic portfolios. Both the OSPI and Sakai projects benefit from greater functional capabilities as well as a wider pool of talent and contribution. Enhancements to one project benefit the other and vice versa.
In addition to each project feeding the other with a larger pool of talent and innovative new capabilities, the combination of the two will result in a larger number of platform combinations. This, in turn, will lead to improved performance and reliability.
Looking to the future, by remaining a strong supporter of the emerging IMS specification, OSPI tools will be enabled in an even wider set of learning-focused applications.

Appendix E

1st European Sakai Day

Come to Luebeck for the 1st European Sakai-Day
Shaping future E-learning in Higher Education

Sakai: Collaborative Learning Environment (CLE)
Free to use, free to develop, freedom for education

What is the European Sakai-Day?

Next Generation E-Learning is determined substantially by advanced systems integrating collaboration and learning. Covering these challenges will become a critical factor for competitiveness in higher education. Open source solutions for collaborative learning environment will on a strategic level allow universities to share competence and take advantage from existing state-of-the-art-solutions as well as to contribute to the future development of the environment – so enabling the community to participate and to join resources, efforts and knowledge in E-Learning.

Sakai is an open source environment for collaborative learning, initiated from the Universities Michigan, Indiana, Stanford and the MIT. It is one of the largest open source software projects in the world of E-Learning. By now approximately 100 institutions are represented in the partner programme for research and development. The Sakai Foundation has been formed to back up and govern the multitude of activities taking place within the worldwide community. The primary goal is to deliver the Sakai application framework and associated tools for learning, teaching and collaboration that are designed to work together.

Among the Sakai partners there is a growing community of European universities and institutes and there are already a number of Sakai implementations around Europe. As we are seeing more and more interest for Sakai in Europe, Luebeck University of Applied Sciences (LUAS) has taken the initiative to create a forum to focus this interest, and those interested, in the first European Sakai Day. LUAS and its E-Learning spin-off oncampus GmbH are delighted to welcome Sakai friends and newcomers to the world of Sakai on the first conference specifically designed around Sakai for a European audience.

Key objectives of the European Sakai Day are

- to promote the open source Sakai approach in collaborative learning and its architecture towards European stakeholders in E-Learning
- to build the European community of practice by exchange of experiences, developments and practices

- to adress the European perspective on Sakai adapting and deploying the framework in the context of the Bologna-process and Lisbon-strategy
- to encourage transatlantic community building for future expansion of Sakai both in use and in development

Appendix F

Sakai/uPortal Development Review and Plans

The purpose for funding uPortal within the Sakai project was twofold, first to support activities to integrate Sakai into uPoral and secondly to invest in continuing to support and grow the leadership of the uPortal community.

There were two phases to the technical requirements of the Sakai project.

The first phase was to implement support for JSR-168 and WSRP-Consumer in uPortal and deliver it in a released version of Sakai.

The second phase was to support development for the next version of uPortal (version 3.0) with strong support throughout for JSR-168, WSRP-Producer, and WSRP-Consumer.

The first phase of the uPortal requirements were completed by the summer of 2004. The currently delivered release of uPortal 2.4 now supports both JSR-168 and WSRP portlet standards. The uPortal team was able to quickly provide these two capabilities by installing adapters. These were the most important technical requirements for Sakai because uPortal provided Sakai with a production-quality stable testing environment to allow Sakai to build their JSR-168 and WSRP-Producer capabilities to integrate into uPortal. This rapid delivery is a testament to the talent and focus of the uPortal implementation team. Other early Sakai-funded work on uPortal 2.x included performance and scaling improvements. The uPortal team also was always available to help the Sakai team with technical questions about JSR-168 and WSRP.

The uPortal 3.0 rewrite took the remaining 18 months of the project. The final production release of uPortal 3.0 has not been released. A number of milestone releases were produced throughout 2004 and 2005 and a number of joint developer meetings were held. The uPortal 3.0 code is in a Alpha-ready state and is awaiting community resources willing to do pilot deployments of uPortal 3.0 to help refine the requirements and put the finishing touches on uPortal 3.0.

The uPortal 3.0 product contains a number of important innovations:

- Solid native support for JSR-168
- Solid native support for WSRP-Consumer and WSRP-Producer
- Refactored pipeline-style rendering architecture
- Support for Spring configuration for major uPortal components
- Refactored Data Access Objects using Hibernate 3 for persistence

At 2004 year-end Chuck Severance reported “During the first year, Sakai-uPortal has produced excellent results. The uPortal team has consistently delivered capabilities well in advance of when those capabilities were needed by the Sakai effort.”

Appendix G

Virtual Research Environment Projects using Sakai in the UK

This appendix lists some of the eResearch projects using Sakai in the UK.

Project Name: SocSciVre
Institution: Cambridge
To provide and support the Sakai platform for large, distributed social sciences research projects. The focus is on activities to investigate the needs of the TLRP researchers and to evaluate the extent to which Sakai meets those needs.
Technology: SakaiPortal, DSpace
http://www.jisc.ac.uk/index.cfm?name=project_sakai_edu_research

Project Name: PoliticsVre
Institution: East Anglia
To develop a virtual research and research skills development environment, capable of expansion and of facilitating multiple participation in the rapidly evolving field of the history of political discourse. Domain area: History of political discourse 1500-1800
Technology: SakaiPortal, AccessGrid
http://www.jisc.ac.uk/index.cfm?name=vre_political_discourse

Project Name: SakaiPortalVre
Institution: Lancaster
To address the requirement for a single point of access to a comprehensive set of Grid and collaboration services in a VRE.
Generic collaboration and research tools
Technology: SakaiPortal and WebServices
http://www.jisc.ac.uk/index.cfm?name=vre_sakai_portal

Project Name: GrowlVre
Lead Site: Cambridge
To build upon the existing prototype GROWL library to produce a truly lightweight extensible toolkit which complements other solutions. Scientific domain: Bio informatics, Chemistry and Stats
Technology: WebServices, AccessGrid and SakaiPortal
http://www.jisc.ac.uk/index.cfm?name=vre_growl

Project Name: CheshireVre
Lead Site: Liverpool
To identify and integrate a number of tools and technologies for the data grid and digital library communities which will support collaborative e-Research across institutions and domains using Sakai.
Technology: SakaiPortal and CheshireSoftware

Project Name: SilchesterVre
Lead Institution: Reading
To develop a system to facilitate rapidly developing and iterative archaeological research by synchronising the three processes of gathering information, co-ordinating expertise, and managing the resulting body of data.

Technology: SakaiPortal
http://www.jisc.ac.uk/index.cfm?name=vre_silchester

Appendix H

Sakai Project Budget Synopsis

The Mellon Foundation has been provided with a complete budget report from the University of Michigan. This synopsis is included here for reference.

y	Print Total Revenue? (y or n)
y	Print Total Invoice Amount Due? (y or n)
n	Does report require the manager's/associate director's signature? (y or n)
n	Are there budget deviations that require a project director explanation letter? (y or n)

The University of Michigan

Final Financial Report

Sponsor: Mellon, Andrew W., Foundation
Award Number: Awd ltr dtd 12/19/03
Project Title: 381663-The Sakai Project

Report Period: 01/15/04 - 01/14/06
Project Period: 01/15/04 - 01/14/06

Budget Category	Budget	Previously Reported	Current Expenses	Expenses to Date	Balance Remaining
Salaries & Wages	390,696.00	0.00	370,683.02	370,683.02	20,012.98
Administrative Salaries	0.00	0.00	22,882.55	22,882.55	(22,882.55)
Clerical Salaries	0.00	0.00	0.00	0.00	0.00
SRO Salaries	0.00	0.00	0.00	0.00	0.00
Fringe Benefits	105,454.00	0.00	98,382.16	98,382.16	7,071.84
Tuition Waiver	0.00	0.00	0.00	0.00	0.00
Consultants	0.00	0.00	4,924.00	4,924.00	(4,924.00)
Respondents/Subjects	0.00	0.00	0.00	0.00	0.00
Stipends	0.00	0.00	0.00	0.00	0.00
Student Aid	0.00	0.00	0.00	0.00	0.00
Patient Care	0.00	0.00	0.00	0.00	0.00
General Supplies	0.00	0.00	12,816.10	12,816.10	(12,816.10)
Research Supplies & Services	3,850.00	0.00	9,791.26	9,791.26	(5,941.26)
Data Processing	0.00	0.00	33,335.00	33,335.00	(33,335.00)
Postage and Shipping	0.00	0.00	84.90	84.90	(84.90)
Telephone Services	0.00	0.00	2,720.72	2,720.72	(2,720.72)
Pass Through Under 25K	50,000.00	0.00	0.00	0.00	50,000.00
Pass Through Over 25K	1,770,000.00	0.00	1,819,443.14	1,819,443.14	(49,443.14)
Alt & Renovations	0.00	0.00	0.00	0.00	0.00
Travel-Domestic	48,000.00	0.00	82,165.43	82,165.43	(34,165.43)
Travel-Foreign	20,000.00	0.00	6,507.58	6,507.58	13,492.42
Travel-Trainee	0.00	0.00	0.00	0.00	0.00
Hosting	0.00	0.00	9,755.06	9,755.06	(9,755.06)
Equipment	12,000.00	0.00	0.00	0.00	12,000.00
Computing Center	0.00	0.00	0.00	0.00	0.00
ULAM	0.00	0.00	0.00	0.00	0.00
Other Spec Svc Fac	0.00	0.00	0.00	0.00	0.00
Unallocated	0.00	0.00	0.00	0.00	0.00
University Investment Pool	0.00	0.00	(32,586.20)	(32,586.20)	32,586.20
Transfers	0.00	0.00	(40,904.72)	(40,904.72)	40,904.72
Total Indirect Costs	0.00	0.00	0.00	0.00	0.00
Total	**2,400,000.00**	**0.00**	**2,400,000.00**	**2,400,000.00**	**0.00**
U/M Cost Sharing	0.00	0.00	0.00	0.00	0.00
Amount over Authorization	0.00	0.00	0.00	0.00	0.00
Total to Report	**2,400,000.00**	**0.00**	**2,400,000.00**	**2,400,000.00**	**0.00**

Total Revenue Received: $2,400,000.00

Total Amount to Return: $0.00
Total Unpaid Invoice Amount (doesn't include final): $0.00
Total Cost Sharing: $1,945,491.03

**This report prepared by Brenda L. Phillips, blphil@umich.edu, (734) 647-3796.

NOTE: Project Director will send a letter explaining budget deviations.

HARDIN,JOSEPH B
Project Director — Date

Kay E. Bressler
Accounting Supervisor II — Date

U/M Project/Grant # N005340-381663 Summary

#