

January 4, 2006

David Porush  
Executive Director  
SUNY Learning Environments  
State University Plaza T-305  
Albany, New York 12246

Re: Response to SUNY's Request for Comment on SLN2.0

Dear David:

I have prepared the attached response to your request for public comment to your Technology Strategy Report dated October 24, 2005.

According to your web site the date for comments was to be November 14, 2005. I submit these comments because of several personal communications with SLN team members encouraging people to share their opinions on the document. You are welcome to make this document part of the public record of your investigation. I understand that based on the lateness of the document it may not receive full consideration in your process.

I am responding as an individual so to make it very clear that this response is not a consensus position of the Sakai community - but instead one person's opinion. I am the Chief Architect of the Sakai Foundation and a board member of the Sakai Foundation but this still simply my own opinion.

I start with a review of the Technology Components of the Proposed [SLN 2.0] Solution from the Strategy Report, ask some rhetorical technical questions and point out some concerns and risks in the approach that is proposed in the Strategy Report.

I then propose my own roadmap that is a variant of the approach proposed in the SLN 2.0 report. As you might expect, Sakai features more prominently in my roadmap. My roadmap is driven by a desire to balance between two competing themes:

- The need to be conservative and deploy enterprise level solutions that meet end-user needs and are seen as an improvement upon SLN 1.0 by both the adopting campuses and the end users of the software.
- The desire for SLN 2.0 to be "more innovative" than a basic-off-the-shelf Learning Management System. SLN is interested in deploying supporting solutions that are best-in-class for teaching and learning.

This conflict between "safe"/"production-ready" software and the simultaneous desire for innovation is in part what fuels the entire Sakai effort. I will readily admit that the Sakai project has focused on the "safe part" of the LMS problem space for its first two years.

While my comments are tuned toward the SLN 2.0 RFC, at some level, this document represents how I think that increasingly teaching and learning software will have to operate - that is we will be moving from a monolithic "Learning Management System" to a set of composable and reusable "Learning Lego Blocks".

I hope that you and your team will take these comments in the context that I intend them. Because my analysis is somewhat technical I had to make assumptions about details that were not present in the Strategy Report. Some of my analysis is based on my assumptions about the report and not the report itself. As such, my analysis and comments may miss the mark. I apologize in advance for any incorrect assumptions.

It would have been easier if I had the opportunity to interact with the team who prepared the report to clarify some of my assumptions. I do feel that to be fair, like the other respondents to the Strategy Report that I should go on record with my complete comments before engaging in a more interactive discussion with the SLN team. I will be quite happy to revise this document in light of any feedback I receive from the SLN team.

I am quite sure that the SLN team will be successful regardless of which combination of solutions are chosen and look forward to supporting the team in any way I can. If SLN chooses to go with its original proposed solution (uPortal + LAMS + parts of other applications), I see the effort of SLN as quite valuable helping move the entire field forward towards a more component-based approach.

I look forward to any questions and/or further interaction with you or other members of the SLN team.

Sincerely,

Dr. Charles Severance  
csev@umich.edu  
<http://www.dr-chuck.com/>

## Introduction

In this document I review and comment on the SUNY Learning Network (SLN) Technology Strategy Report dated October 24, 2005. While this response is very much about Sakai, I am only responding as an individual who happens to be the Sakai Foundation Chief Architect and a Sakai Foundation Board member.

None of my comments in this document should be perceived as anything other than one person's honest opinion and recommendation after reading the SLN Technology Strategy Report. In particular, none of this should be viewed as any type of promise from "Sakai". The future directions of Sakai are really up to the members of the Sakai community who contribute to Sakai. As Chief Architect, I try to influence some of the investment the community makes in Sakai, but I can only give community members advice - not give them orders.

My basic approach to this document was to try to imagine what I would do if I were working for SLN for the next few years.

This is great time for SLN 2.0 to start. With Moodle, Sakai, uPortal, LAMS, Open Source Portfolio, and other projects reaching basic levels of maturity, we are in a position where open source solutions are emerging to solve not just our infrastructure needs (Linux and Apache) but also our application needs. While open source application solutions are still growing and maturing there is software that is available today that is roughly competitive to commercial offerings.

Some of my analysis is based on assumptions I make by "reading-between-the-lines" in the SLN strategy report in terms of some missing technical detail. Perhaps some details are simply not yet known. Those parts of my analysis that are based on incorrect assumptions on my part can easily be wrong and I hope that you will ignore them.

This analysis in this document is *only* intended to be applied to the SLN request for comments. None of my comments within this document should be taken out of context. I am happy to clarify anything in this document or have further discussions with SLN or other organizations about the contents of this document. I prepared this document quickly, to be as responsive as possible to timeline of the SLN team. If the document is inaccurate or mistakes in the document come to light, I would like the opportunity to revise it.

## SLN 1.0 Review

SLN is currently based on a Lotus Notes/Domino with significant local extensions and improvements that is reaching the end of its useful life. The SLN team presents several important conclusions in the Strategy Report:

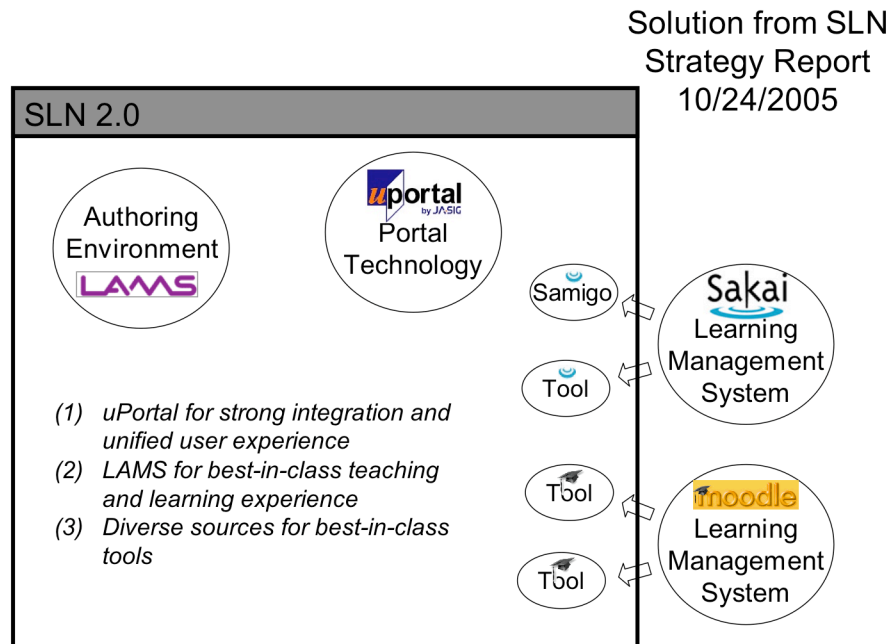
- There is only one other institution in the world that is using Notes/Domino for LMS applications
- The technology underlying SLN1.0 is increasingly difficult to scale to the levels of users needed in terms of performance
- It is increasingly difficult to evolve Notes/Domino to meet new end-user requirements (as an example, it would be rather difficult to add WebDav to Notes/Domino)
- IBM has informed SLN that there is no logical follow-on product to Notes/Domino that fits the SLN application needs

This is a pretty strong case that something needs to be done for SLN to continue and flourish in 2006 and beyond.

Interestingly, the University of Michigan faced the identical set of problems with Lotus Notes in 2000 that led UM to begin the development of the CHEF project that was one of the founding technologies for the Sakai Project in 2004. Many of the current Sakai tools take design inspiration from Lotus Notes.

## Review of Proposed SLN2.0 Solution

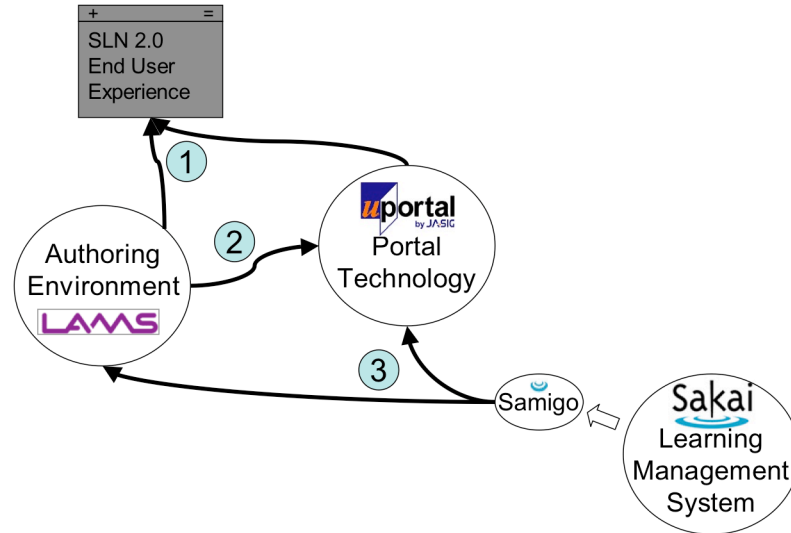
From the Technology Strategy Report, the Proposed Solution for SLN 2.0 is a combination of LAMS, uPortal, and a selection of best-in-class tools from a number of sources including possibly Sakai.



The Strategy Report leaves out some important details as to how this will all be accomplished; leading to some questions I pose in the following section.

## Questions Regarding the Proposed SLN2.0 Solution

The primary area that I am concerned with the above diagram is how the user will perceive a seamless user experience.



**(1) Will the LMS users go directly to the LAMS site, or will the LAMS site be rendered through the portal?**

This question will be a key determiner of the "seamless" user interaction part of the design. I assume that the ultimate goal is to render everything through the portal and use the portal as the place where the elements are assembled together. [This is an important assumption. If this assumption is wrong, many of my comments below are irrelevant.]

**(2) How will LAMS student view be rendered in the portal?**

Certainly, it is not practical for LAMS authoring activity (a desktop application) to operate through the portal. But for the learner view and instructor tracking and interaction view, this is a browser-based operation so this could be done using either an iFrame portlet or a custom-written JSR-168 portlet. At a minimum, to make this appear seamless would require some modifications to the LAMS web-based interface so it would look natural as a pane in the portal's interface.

Sakai has had the capability to "put Sakai into a Portal" for a long time using an iFrame as shown in Appendix with some screen shots from UC Merced. In those screenshots, Sakai has been configured to eliminate any screen real estate dedicated to login or branding and while it is usable and used by the students, instructors prefer to go directly to Sakai when they do any significant work.

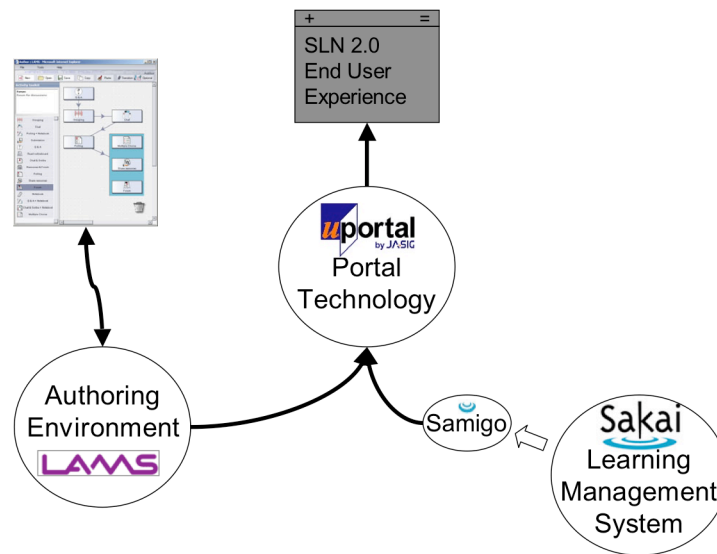
**(3) Will the assorted tools (such as Samigo) be rendered to the user through the portal or will these tools be integrated into LAMS? If so, will this be done via an iFrame or using a custom-designed JSR-168 portlet?**

If you are setting LAMS up to be the primary learning experience that instructors author and students experience, the most natural interaction would be to make Samigo (or other non-LAMS tool) appear *\*within\** LAMS. This is an ideal approach but quite technically challenging. Sakai is planning on working over the next year to align our efforts in such a way as to align Sakai and LAMS to make this possible, but it is just not practical for the SLN team to do this quickly as an add-on or "glue".

Sakai is working to make our tools and capabilities available to LAMS run-time environment [LAMSSakai1] because we see this is a valuable direction forward for Sakai, but this is not something that will likely be done for the average best-in-class tool that you will encounter. Connecting the LAMS run-time engine with Sakai tools is a task that I have allocated a year of investigation just to "understand" the nature of the problem.

So, I assume that you intend to bring LAMS and Samigo (and other non-LAMS) tools together in the Portal by placing them "near" one another in the portal and using iFrames or JSR-168 portlet placements.

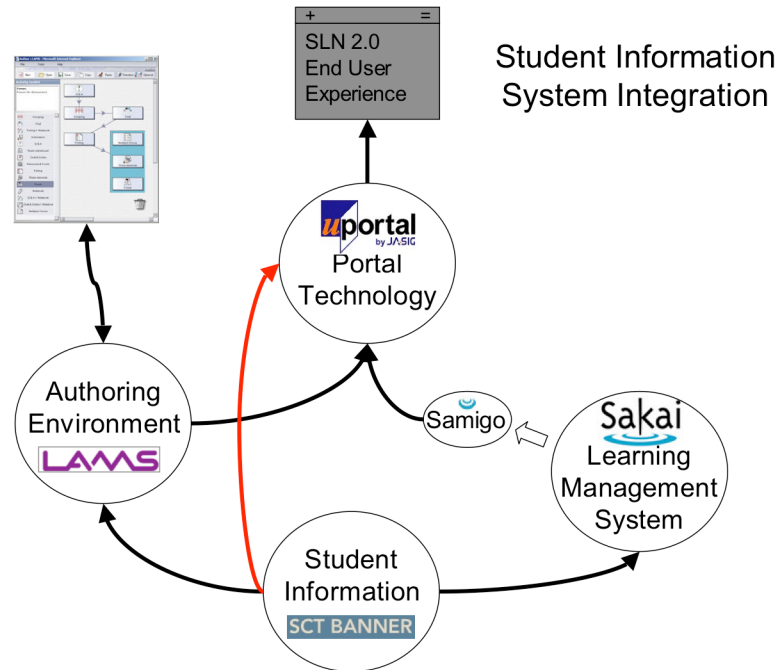
Working with my basic assumption that the portal is how the students and instructors will see SLN2.0, the implementation diagram looks roughly as follows:



If this is the proposed architecture, then another technical concern arises. At some level there will need to be some way to automatically provision these three technologies so that they all know about all of the courses, rosters, students and teachers. This is further complicated by the fact that SLN serves multiple campuses and multiple administrative

domains. But for now, let's assume that there is a single source of Student Information (presumably from the SUNY-wide Banner Initiative).

To produce a seamless user experience, you will need to integrate roster and instructor information into each of the components (Sakai, uPortal, and LAMS). Since LAMS and Sakai are already designed as learning environments this is a relatively straightforward technical problem.

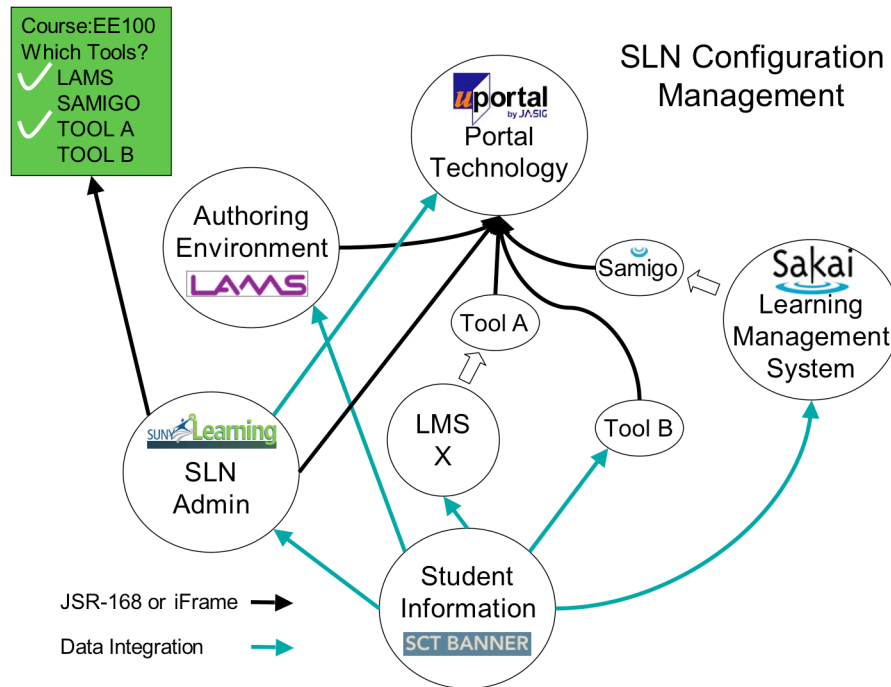


However, for the portal this is more challenging. While uPortal has a very flexible Groups and Permissions structure (GAPS) in terms of portlet layout, it ultimately wants to organize independent, unrelated rectangles (portlet placements) - mostly under the control of the portal administrator.

If you look at the UC Merced examples in the Appendix, you see a set of tabs across the top. These portal tabs were set up by the portal administrator and not the end users. The view is pretty much the same for all users (with some personalization). Within each tab, there is user-specific information - when the user goes to the Mail tab - they see the Mail application and their mail. When the user goes to the UMCROPS (Sakai) tab, they all see Sakai, but Sakai shows each user exactly the courses and tools that they are authorized to see/use based on their roles within the roster in the Student Information System. UC Merced's approach uses portal navigation for the simple basic navigation *between* applications hosted in the portal, but allows each application to display the right set of information personalized to each user *within* the portlet frame.

If SLN's intent is to use the uPortal to bring the different tools from different sources together, then portal navigation will have to be dynamically generated based on both the roster information and which tools the instructor or SLN staff have selected for the

particular course. This begins to add a layer of management between Banner and uPortal as shown below.



This begins to become an SLN Administration tool that auto-configures the portal and "pushes" the instructor's chosen configuration into the portal. In the long run, this sounds pretty clever - there are a few caveats that you should look into fully before committing to this approach:

- Each of the blue lines (Data Integration) can be quite challenging - each new application will require its own data integration. This will slow your ability to deploy new applications. Some of the tools which are not already part of an LMS (Tool B) you will want to integrate will actually not even have the notion of courses and instructors and so you will have to map the Course/Student/Instructor data models onto the application's internal data models as part of the integration of the Tool
- The pushing of dynamic fragments of portal navigation is an advanced use of portal technology - you should look closely into the technical detail and build a prototype with pushed fragments and see if you can produce an interface (including the instructor's interface for course setup) that you feel will satisfy your users.

This section poses some technical questions and raises some technical concerns - much of this section's analysis depends on a critical assumption that the portal will be the way that SLN delivers the "unified user interface". If this assumption is not true, then many of the concerns in this section may not apply.

## Comments on the Proposed Solution

The use of a portal as a way to collect capabilities for your user community is a good direction. By adopting uPortal, you get a good collection of open source portlets developed by and for higher education. The direction forward for uPortal is controlled by higher education.

By adopting a portal and building your new capabilities to the JSR-168 Portlet standard you allow yourself to retain software elements across infrastructure changes.

As you will see in my proposed approach in later sections, the portal continues to be a central feature of my proposed approach as well.

### ***Risks/Concerns I See with the Proposed Solution***

The risks I see with the proposed solution is as follows:

You are not using any of the applications in their operational "sweet spot".

- uPortal is an enterprise portal - not a learning portal. The uPortal community is primarily made up of campus IT organizations deploying the portal for a campus and placing campus-wide services such as e-mail, calendar, and library in the portal.
- You are not using LAMS as a way to deploy Learning Design - but instead you are using it as nearly your entire LMS.
- You are not using Sakai as an LMS, but extracting tools from Sakai.

This will ultimately put you somewhat out of step with the communities for each product and force you to produce local solutions that cannot easily benefit from efforts going on elsewhere.

Each new capability requires a brand new integration starting from scratch. One of the stated goals of SLN 2.0 is to be able to accrete innovation more rapidly and not be limited to capabilities produced/integrated by SLN.

LAMS is clearly the leading open source example of the use of Learning Design techniques in teaching and learning. However it does not provide the "basic LMS" capability equivalent to Sakai, WebCT, Moodle, Blackboard, Angel. My biggest concern is that LAMS organizes things into flows with some sense of order, whereas the more traditional LMS systems simply provide capabilities/tools that are always there and can be used in any order and with no need for the instructor to "make a script".

This is not to suggest that one approach is "better" than the other - but simply to suggest that at times, faculty and students will prefer the simpler unordered structure of a non-learning design interface to learning capabilities.

I wonder what you have thought about the process of converting existing courses hosted in SLN1.0 into LAMS. Since I have not had the opportunity to do an in-depth analysis of SLN1.0 I will have to make another assumption - I am guessing that much of the existing SLN1.0 course material will not map naturally onto sequences. I am guessing that SLN1.0 has a course calendar, threaded discussion, area for sharing files, and other features from Lotus/Notes. While some of the SLN value-add beyond Lotus/Domino may be sequence oriented - I am guessing that there is also heavy use of these non-LD features that are part of Notes.

The primary risk that SLN will face if LAMS is the only learning management system that you deploy to your users is that effectively it puts SLN in the position of convincing people that the LAMS-way is the right way. It puts the SUNY campuses in the position where they can either go with SLN and accept LAMS as their primary LMS or install their own WebCT, Moodle, Sakai, or Blackboard.

Please don't take from these comments that I think that the choice of LAMS (or uPortal for that matter) are bad choices on the part of the SLN. The opposite is actually true. I think that both LAMS and uPortal are the strongest players in their respective fields. Sakai has been working with uPortal for over three years and is planning on increasingly close collaboration with LAMS over the next year to align efforts between the projects [LAMSSakai1].

The key is that LAMS provides capabilities that are not available in Sakai. Sakai has a choice - either build some type of Learning Design layer on top of Sakai or simply work with LAMS so that Sakai can be "plugged into" LAMS as a set of tools. My opinion is that it would be a colossal waste of time for Sakai to try to compete with LAMS in the space where it is the clear leader amongst open source projects.

The mistake in the SLN design is the lack of including of an "old-school" LMS system. I can understand the desire to make the "future" happen immediately - one way to force oneself to move forward is to operate without a "safety net". However because SLN is doing things "the hard way" - before the respective communities are producing the solutions that SLN needs, the path that SLN has chosen will take far longer than you currently imagine. You will be well served to have a safety net.

## **Alternate View of SLN 2.0**

My primary change in the design of SLN 2.0 is to add Sakai as the "safety net" which will provide a base-level functionality that will allow you to meet the immediate needs of your users and quickly transition from your Lotus Notes/Domino system to a system with new technology.

This will also provide SLN with a ready answer to the questions such as "Why not WebCT?" or "Why not Moodle?". And while Sakai may initially be only a safety net, SLN may find that Sakai capabilities are a better match for some of the SUNY campuses than LAMS. By allowing those campuses to make use of Sakai, SLN is better serving its constituency.

I break down my plan for SLN into three phases:

- SLN2.0 - Quickly deploy LAMS, Sakai, and uPortal. uPortal and Sakai are the production ready and available to begin the transition from SLN1.0 by Fall 2006. Sakai is integrated into uPortal and uPortal is the "single point of contact". LAMS is also fully integrated and production ready but not fully integrated into the portal. LAMS is aimed at a large pilot effort.
- SLN 2.1 - Fully integrate LAMS into the Portal. Connect Sakai to LAMS. LAMS is capable of being used as an alternate view knitting Sakai and non-Sakai capabilities together.
- SLN 2.X - This is simply a vision of how complex the ultimate component based learning system might be and the different elements that can be brought to bear to provide the most innovative learning environment possible. By the time SLN2.X comes to pass, likely things will change significantly, but it does not hurt to peer into the crystal ball from time to time to align our present efforts with some possible futures.

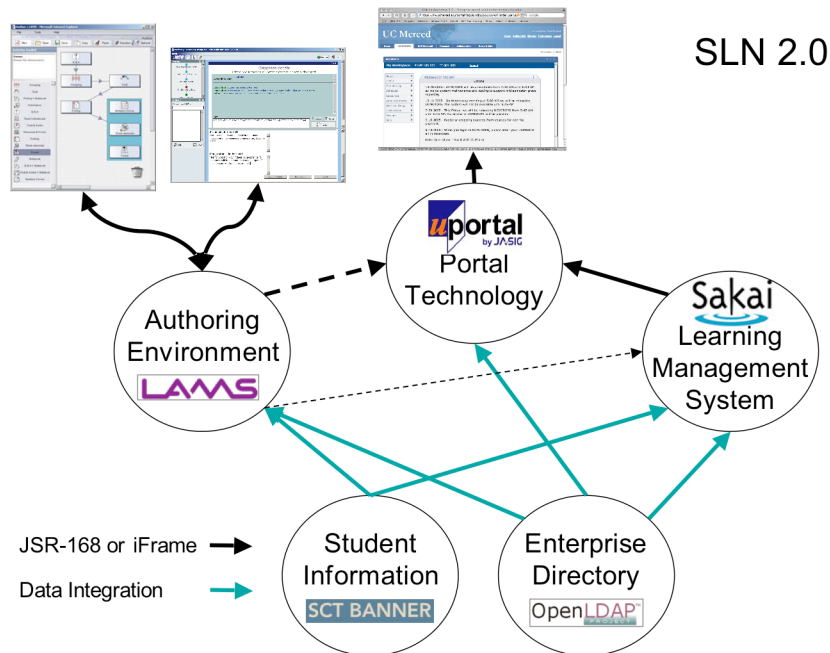
In the following sections, I provide some technical detail as to how each of the phases would work out.

### ***SLN 2.0 - Conservative Deployment***

My suggestion for SLN 2.0 is to simply deploy Sakai, uPortal, and LAMS out of the box with no major modifications to any product beyond enterprise data integration, and the application of the SLN look and feel.

By not investing in major innovative software and using each of these products in similar fashion to other higher educational institutions it allows SLN to focus on other transition issues such as: deployment, performance, data conversion, system administrator training, instructor training, and student training.

There is no reason that you could not roll out SLN 2.0 for initial test and use by Fall 2006.



## Technical detail

- Sakai is fully integrated into uPortal using JSR-168. uPortal is the only way that your users see Sakai. This is done using a series of completely standard JSR-168 portlets written as part of Sakai that will run in any JSR-168 compliant portal (uPortal or otherwise). These portlets are well underway in development and should be ready for your use well in advance of Fall 2006. (See appendix for more detail).
- LAMS is accessed by both the students and teachers directly. This allows them to make full use of LAMS and does not constrain LAMS deliver to that which can be delivered via the portal. It would be nice to do a very simple integration between LAMS and the portal (dotted line) that allows those who want to have a LAMS dashboard on their portal to do so. At some level you might be able to convince the LAMS foundation to produce a JSR-168 portlet that provides a LAMS interface in the portal. But I would not depend on that dashboard providing a rich enough experience for your users.
- You can see how such a LAMS portlet dashboard might work by looking at the integration between Sakai and LAMS (small dotted line in the above diagram). You can see a demonstration of the LAMS-Sakai integration at [LAMSSakai2]. I would expect that a JSR-168 portlet would operate in a similar fashion. The complexity in any of these "Dashboards" is the coordination of the rosters and roles between the applications. You can seem some of the technical detail of the integration by looking at the post installation configuration of the LAMS / Sakai connector [LAMSSakai3].

- Directory information (identity) is integrated into all three applications. The Strategy Report mentions system-wide LDAP. Integrating LDAP into these three applications will be pretty straightforward.
- Student information (rosters and roles) is *only* integrated into Sakai and LAMS. Since Sakai and LAMS are learning applications, their internal data models are quite ready to accept information from Banner and alter their user interface accordingly. As I mentioned above, it is not difficult to get student information into uPortal using uPortal's GAPS capability, it is more challenging to get uPortal to produce the highly dynamic user interface based on course role and membership. My SLN2.0 approach avoids this difficult problem.

The goals of SLN 2.0 are as follows:

- Quickly begin the process to retire SLN 1.0 using Sakai as the destination for the typical user
- Bring LAMS to your campuses and users in a low stress environment (i.e. you are not forcing users to use LAMS). If the SLN user community is as interested in Learning Design as you suggest in the Strategy Report they will love LAMS, use it heavily and give you great feedback as to how it could be improved. Also you can test LAMS with those members of your community who are not so pro-LD and determine if there is a set of features that will allow LAMS to satisfy their needs as well.

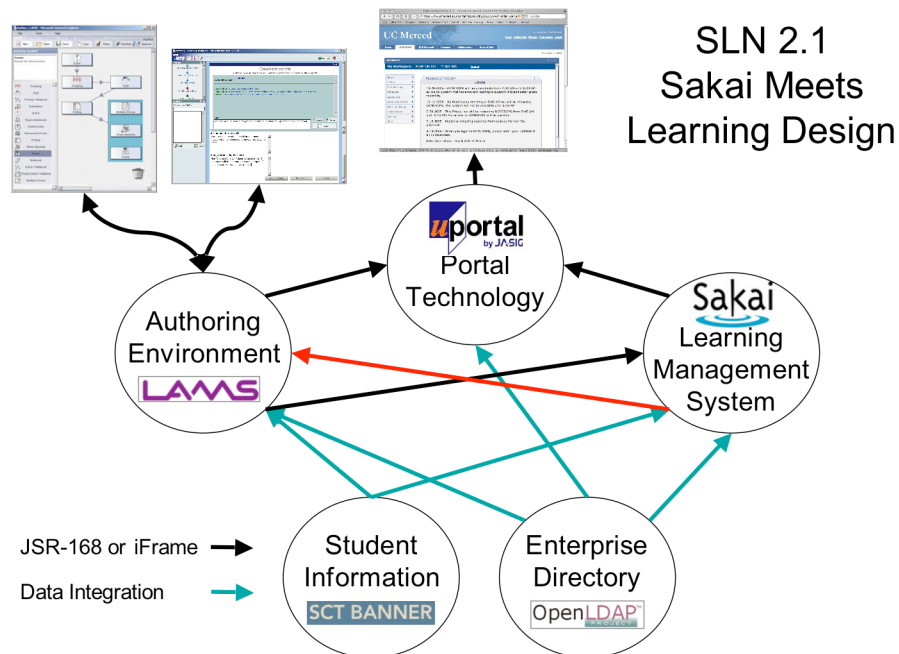
The primary benefit of SLN 2.0 is that SLN has moved from technological isolation to being part of three vibrant developer communities. Instead of finding yourselves at odds with the communities, SLN is deploying and using the software being built by those communities in a virtually identical manner to the other members of those communities.

The additional benefit is that SLN is placed squarely in the overlap between the Sakai and LAMS projects and is in a position to dramatically affect that relationship going forward, both based on your knowledge and use of the two products, but also because SLN possesses developer expertise that can be applied to bringing those two software elements together [LAMSSakai1].

## ***SLN 2.1 - Innovation and Learning Design***

The goal of SLN 2.1 is to transform both Sakai and LAMS so as to allow LAMS to function as a LD layer on top of Sakai capabilities. When this effort is complete users should be able to move seamlessly between the "old school" tabs-and-buttons style of teaching and learning to a scripted/organized/workflow-oriented view of teaching and learning.

The goal is to provide an environment that combines the best of both worlds in a seamless manner. This is a great challenge and will likely take about two years. It requires a significant internal refactor for Sakai which will allow fully compliant Sakai components to be fully scriptable and usable in both the LAMS authoring and run-time environments. Once this work is done in Sakai, it will be done in such a way so as to ultimately allow Sakai to work with other Learning Design authoring environments and run-time engines such as CopperCore [CopperCore] and Reload [Reload]. We hope and expect that the patterns of interaction between Sakai and LAMS result in a set of standards for this type of interaction between a set of tools and a Learning Design environment<sup>1</sup>.



So while this is a great challenge, the ultimate goal (represented by the red line in the above diagram) is the ability to use any application operating within Sakai to be made available to LAMS (and ultimately other Learning Design engines).

Looking at this from an SLN perspective, SLN gains great advantage by quickly deploying both LAMS and Sakai and developing a deep understanding of both products during the SLN2.0 phase. By deploying and supporting both products SLN staff will learn the strengths and weaknesses of both products and be well equipped to contribute

<sup>1</sup> This would be a good place to remind the reader that I have no ability to promise anything on behalf of either Sakai or LAMS. As chief architect, I can only try to convince communities that these are good directions to go. The ultimate delivery of these capabilities depends on community resources being brought to bear on these problems.

strongly to the efforts to perform the integration in a way that will most benefit the SLN community.

If you look at this from a time perspective, during the time period January through September 2006, SLN worries about just getting basic off-the-shelf software into production for the SLN so that the transition from SLN1.0 to SLN 2.0 can begin. At the same time, Sakai and LAMS hope to work together to understand the requirements of a Sakai / LAMS connection and begin to develop a design strategy for going forward. Toward the end of 2006, just as SLN 2.0 is calming down, the development effort for the Sakai / LAMS connection will be ramping up and SLN technical staff should have time available to participate in both communities and help produce the ultimate solution.

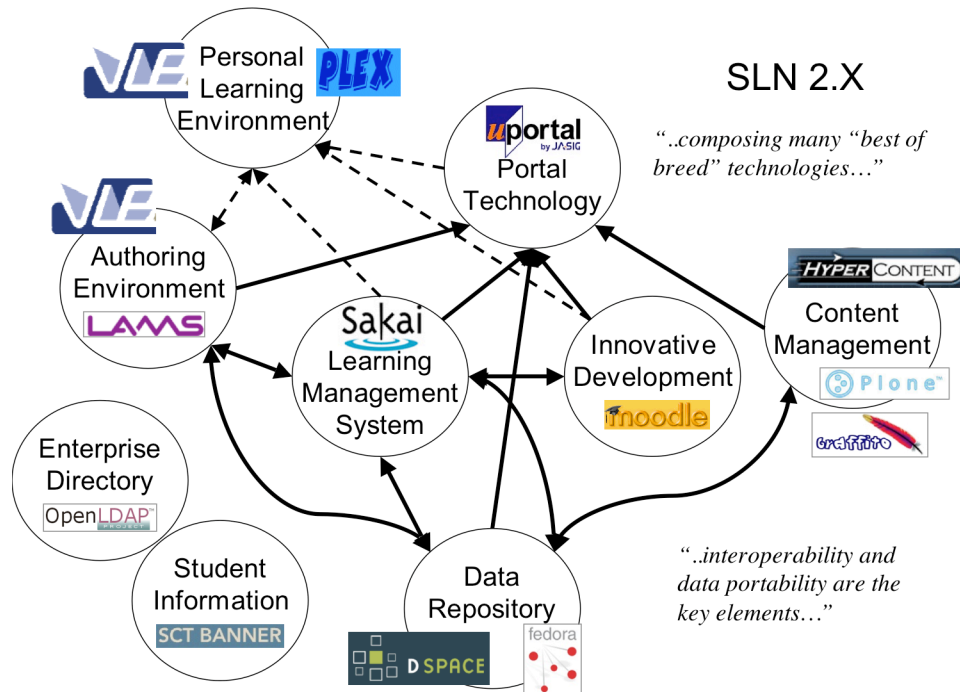
I would hope that with sufficient resources and investment we would see initial results of this effort by Fall 2007 that can be deployed in SLN2.1.

SLN 2.1 holds out the promise that LAMS could possibly be the sole user interface that you provide to your SLN community. But this will require a lot of effort and some iterative development over several years.

### ***SLN 2.X - A Component-Based Future***

There are more dimensions to teaching and learning than just improving our use of Learning Design approaches. This section is a bit fanciful, but is intended to get the SLN team thinking more broadly about the kinds of services you should be thinking about deploying to your community over the next five years and how the SLN technical team will invest its resources to see some of it come to pass.

Here is the architecture that I recommend for SLN in the 2009/2010 timeframe.



This is a pretty busy diagram so I will describe the elements at a high level.

## Portal

The portal is becoming increasingly important as we deploy more and more applications to our user base. We cannot afford users maintaining yet one more bookmark to a new service (say Content Management) which we deploy Fall 2008. As we roll out new functionality we need a place for users to find out about new functionality and then use that functionality. It is important to note that we are not trying to force-fit each of these new applications into the portal. We will increasingly use technologies such as Web Services for Remote Portals (WSRP) rather than JSR-168 to bring new applications into the portal. We hope that new applications will simply come off-the-shelf with the proper WSRP adapters built right in.

## Learning Management

Sakai and LAMS are still there from SLN 2.1. Of course in the course of several years, Sakai has many new tools that have been developed and integrated by the Sakai community and all of these tools are immediately available in LAMS.

## Authoring Environment

There may be a number of different "views" on how to present information to learners that will emerge and be tested over time. The Visual Understanding Environment [VUE]

project is looking at teaching and learning from the knowledge representation perspective. Whereas LAMS characterizes learning as a "sequence of activities", VUE looks at building "knowledge maps" and "knowledge pathways" which allow the student to move throughout a knowledge space in either a directed or undirected manner. There are many efforts beyond VUE attempting to build "knowledge" from information based on relationships between different concepts rather than directed sequences. We should be open to the idea that there will be multiple ways of presenting information to students beyond the sequence paradigm proposed by LD.

## **Software Development in the Hands of Teachers**

The Moodle project demonstrates that there is a strong desire among some teachers for teachers to be in complete control of the learning experience. Because Moodle is so simple and elegant, the teachers do the programming in PHP and the teachers maintain the servers. This has created a wonderful source of programming talent that drives the Moodle community forward. I would expect that you have a number of areas in the SUNY system that would love to be able to write their own software and use it in their classes. Sakai, Angel, WebCT, LAMS, and other enterprise systems are not likely to meet the need for an easy-to-program environment for the average teacher. At the same time, most IT organizations are not ready to adopt Moodle as their core LMS. In SLN 2.X we need the best of both worlds. Sakai and LAMS provide the enterprise-wide capabilities but Moodle is so well integrated that users can build and host a tool under their desktop and have it appear seamlessly within SLN as just another tool. This is work that is being considered but I know of no resources currently being applied to the problem in either the Sakai or Moodle community. Perhaps sometime between now and 2009 some institution will get the urge to deploy a hybrid solution and invest some effort in solving the technical aspects of a Sakai / Moodle interaction.

## **Institutional Repository**

There will be an increasing need to have an institution-wide repository such as DSpace or Fedora so that materials can easily flow between applications. Like the portal at the top of the picture unifies the disparate user interfaces, an institutional repository can unify the long-term data storage and data exchange for the different applications that make up the enterprise. The institutional repository also allows for the curation of important retained data even after an application has been retired. It would be nice to have a place to retain all of the SLN1.0 course materials in their original form without the need to maintain the Notes/Domino infrastructure for the next 20 years. While DSpace and Fedora are active projects, there is still work to do in this area for them to be suitable for supporting a very wide range of applications. Much of this new work is around more flexible models using technologies such as Web Ontology Language [OWL] and Resource Definition Framework [RDF]. These projects are already aware of and working on these new use cases for development and delivery in the next few years.

Once these repository projects deliver rich products, each application (Authoring, LMS, etc) will need to invest some effort to make their internal data and structures available to

the institutional repository using OWL and RDF. This also is a complex and long effort with the need to develop a new generation of data-representation technologies and standards.

## **Content Management and Publishing**

Another often-overlooked aspect of a learning enterprise is the content management and publishing aspects. This publishing activity can take many forms. One activity is the publishing of reusable courses and learning objects. Another activity is the communications between the enterprise and its users via the organizational web site, press releases, newsletters, etc.

There are current products such as Mambo and Plone in this space, but they are intended to operate as stand-alone applications in support of a web site. The next generation of these products will likely be heavily influenced by standards such as JSR-170 (Java Content Repository) and be more suitable for complete integration into an enterprise environment.

## **Personal Learning Environments**

Even as we can imagine reaching a level of success in our ability to deploy a unified user interface for our enterprise using a portal, the next-generation interfaces are beginning to show some promise.

For me the best model of what a Personal Learning Environment might be is to look at Apple's iTunes software. I would describe iTunes as a "Personal Listening Environment". iTunes primary value is that it allows each user to organize their music according to their own personal tastes. iTunes is enhanced by central resources such as the iTunes Music Store or the Cddb (The database of each album's title tracks) but ultimately its sole purpose is to customize the listening experience for each user.

There are tools that are in development aimed at producing the same type of software, except for the teaching and learning environment. The Personal Learning Environment [PLEX] is a CETIS funded project to develop just such an environment. Appendix C shows a prototype integration of Sakai into PLEX as well as experimental Desktop versions of Sakai written for Apple and PC.

The key issue with Personal Learning Environments is to understand that while the Portal is a critical technology for building a unified web presence for the elements which make up the SLN Enterprise, but also, to think beyond the portal.

## **Conclusion**

My overall goal of this document is to try to get SLN to view its technical efforts in the context of a larger picture. Everything in this space is a moving target and there will be wonderful new capabilities built over the next few years. Reading the Technology Strategy Report, I got the sense that SLN felt that with enough careful analysis, SLN could "pick" the right combination of things and then make that combination work.

The problem is that the combination that SLN picked and the way that was described connecting the components together (some of which I have made assumptions on) is a challenging path forward. SLN has chosen to use technologies in ways quite different than other organizations. I appreciate trying to jump "ahead" at the moment where you are considering new technology so as to skip as many intermediate steps as possible.

However, when I look at migrating a large (and relatively happy) user base from one technology to another, I get pretty conservative. I don't like promises of future features from any vendor (commercial or open source). I like to make my decisions on what I can see, download, install, and use today.

I prefer to innovate in an iterative fashion, always working from a safe production environment. You may think that faculty and students want rapid innovation and to use the best-in-class technologies all the time. My experience suggests that this is not generally the case. Solid production environments that evolve and improve slowly over time are what make users the happiest.

If SLN makes the right choices and invests their development talent wisely, SLN can have a dramatic impact on the still evolving Open Source teaching and learning field all the while keeping your faculty, staff, and users very happy with solid production services based on the best available open source solutions.

One final and very important note: None of my comments should be construed as criticism of any of the products I mention in this document. uPortal and LAMS are excellent products and excellent projects. I have good working relationships with these projects and look forward to closer and deeper collaboration with them. My concern is limited to how SLN is intending to integrate, deploy and use these technologies.

## References

[Academus] [http://www.unicon.net/products\\_519.html](http://www.unicon.net/products_519.html)

[CopperCore] <http://coppercore.org/>

[LAMSSakai1] <http://www.lamsinternational.com/news/#7>

[LAMSSakai2] <http://www.lamsfoundation.org/integration/sakai2>

[LAMSSakai3] <http://lamsfoundation.org/integration/sakai2/sakai-code/README.txt>

[Reload] <http://www.reload.ac.uk/>

[OWL] [http://www.w3schools.com/rdf/rdf\\_owl.asp](http://www.w3schools.com/rdf/rdf_owl.asp)

[Pluto] <http://portals.apache.org/pluto/>

[RDF] <http://www.dlib.org/dlib/may98/miller/05miller.html>

[SLNTechReport] <http://le.suny.edu/sln/rpc/sln2tsr.pdf>

[SunGard] <http://www.sungardsct.com/>

[VUE] <http://vue.tccs.tufts.edu/>

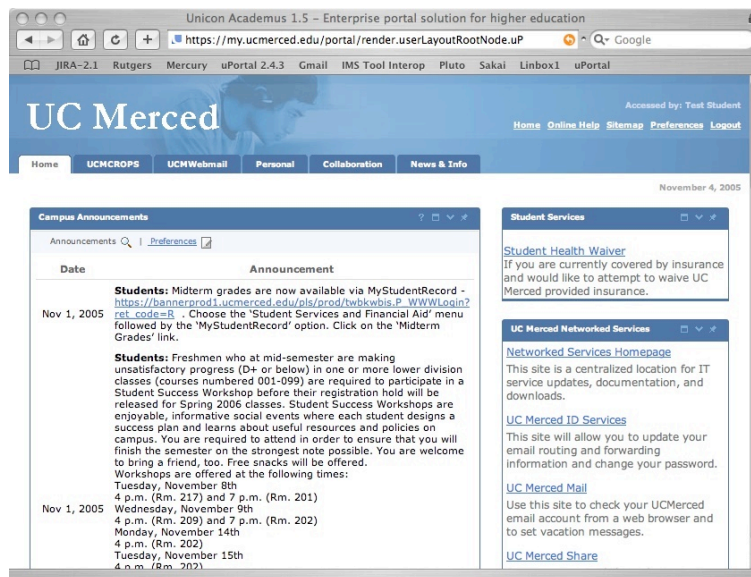
## Appendix A - Screenshots of UC Merced

UC Merced is a Sakai Site. UC Merced is a brand new campus (started in 2003) in Merced, California. Merced uses Sakai for its LMS, the Academus [Academus] variant of uPortal for its enterprise portal, and SCT Banner [SunGard] for finances.

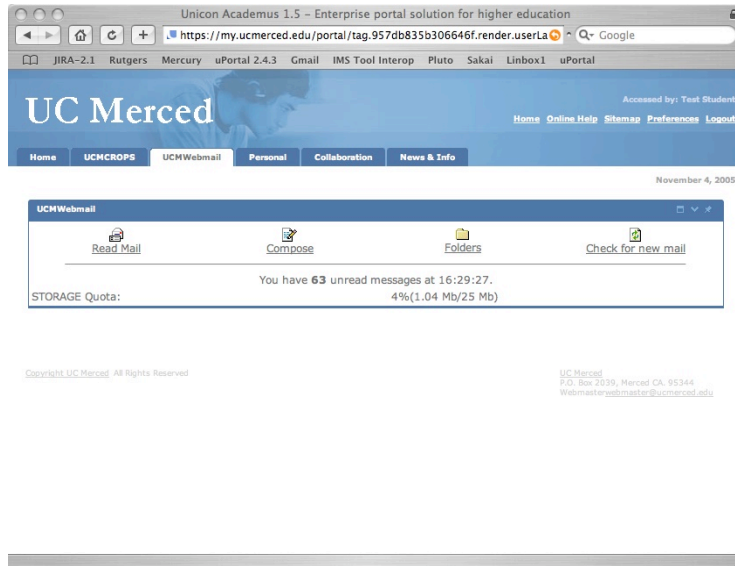
The integration between these elements was relatively straightforward and accomplished in a few months by a small team of talented individuals. UC Merced chose to use all of these tools in their "out-of-the-box" configuration so as to simplify their own integration efforts.

I include these screenshots to give an example of how a learning management System can be viewed easily in a portal. There is a significant difference putting the entire learning management system into the portal as a single portlet and trying to use the portal's internal administrative tools to "assemble" a bunch of portlets to form a reasonable learning management system.

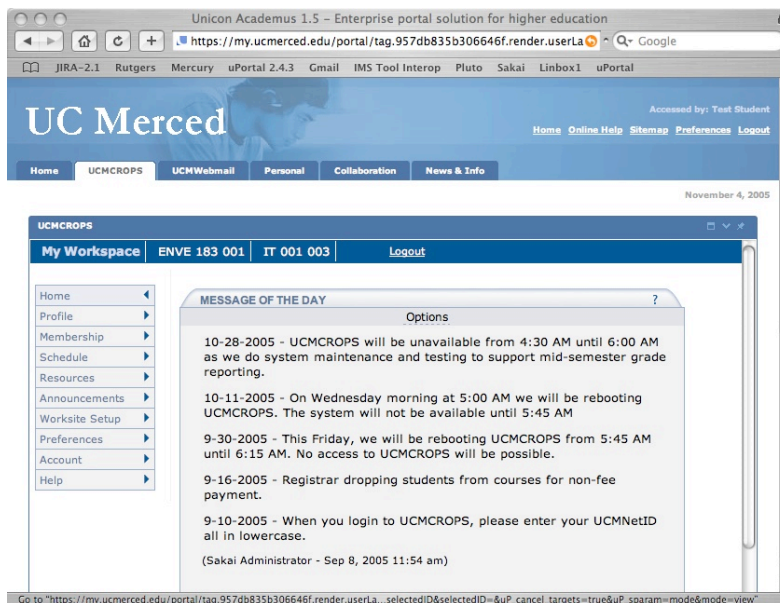
This is an example of the main page of the UC Merced Portal. There are tabs across the top and portlets arranged in columns and rows. Each portlet shows relatively independent materials. The arrangement is not reflected in the programmatic linking of the portlets - the arrangement is just how the system administrator chose to lay out the site.



This is UCB Mail (an open source uPortal channel) in the UC Merced Portal. This is an example of one of the well-developed portlets that are available for uPortal. This portlet is not a JSR-168 portlet - it is an iChannel portlet. A strong advantage of uPortal is that you can use both iChannel portlets and JSR-168 portlets in uPortal. JSR-168 may be the standard, but as of now, there are not a lot of well-developed portlets for higher education available for JSR-168.



This shows how the UC Merced Students see Sakai. Sakai is a single portlet (iFrame in this case). The portal is not aware of any of the course rosters or roles - the Banner integration is only to Sakai. For each user, Sakai displays the right courses *within* the portlet.



## Appendix B - Screen Shots of The Sakai JSR-168 Portlet

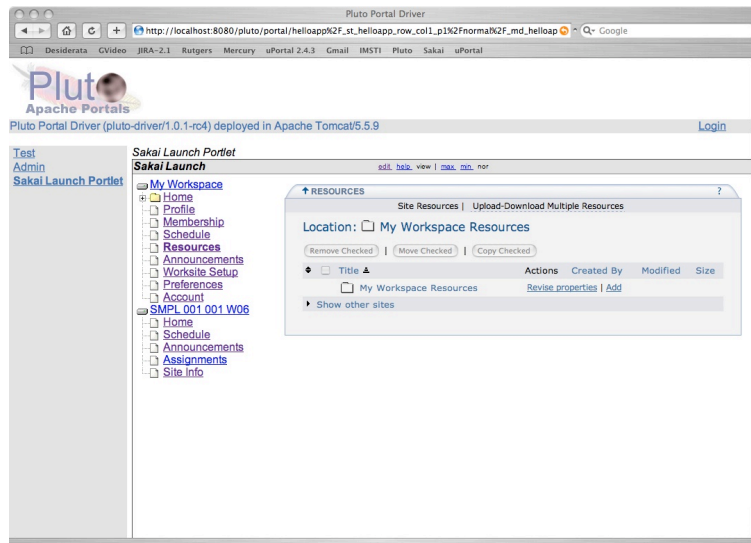
These screenshots are taken using the Pluto portal. Pluto [Pluto] is a simple portal intended for developers to use as a test environment. So the look and feel of the Pluto portal is pretty poor. Most portals actually place the buttons across the top (See Appendix A). The Sakai Portlet is designed to look best in a portal with the buttons aligned across the top.

The goal of this appendix is to show the Sakai portlet capabilities. The Sakai portlet has features that automate sign on between the portal and Sakai as well as features to configure each portlet placement to either be all of Sakai, or a single Sakai tool.

The challenge that limits the use of these Portlets as a complete learning management system within the portal is solving the "provisioning" problem. Each of these portlets must be properly placed and configured and the portal groups and roles must be properly aligned with Sakai's groups and roles. This is something that is being worked on but the solution to the cross-provisioning problem is not something that is part of the JSR-168 portlet spec and as such must be done with custom Sakai logic for each portal product.

These portlets are purely JSR-168 compliant and do not require any special integration to run in a JSR-168 compliant portal.

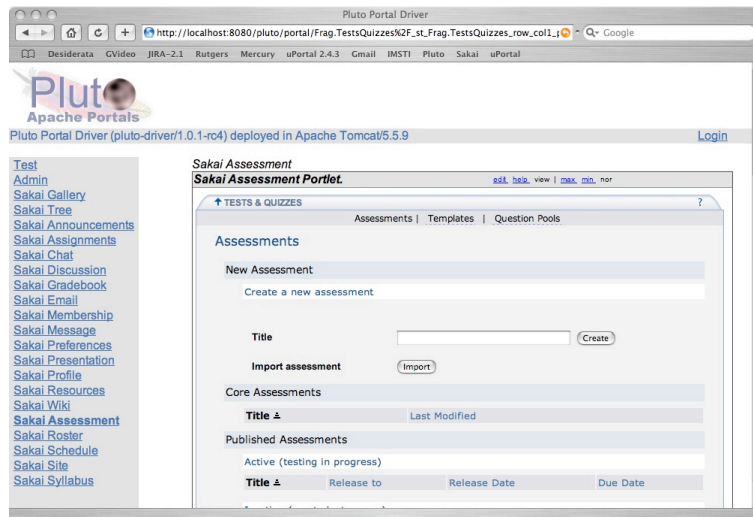
This is an example of a user view of a Sakai Server using a tree view.



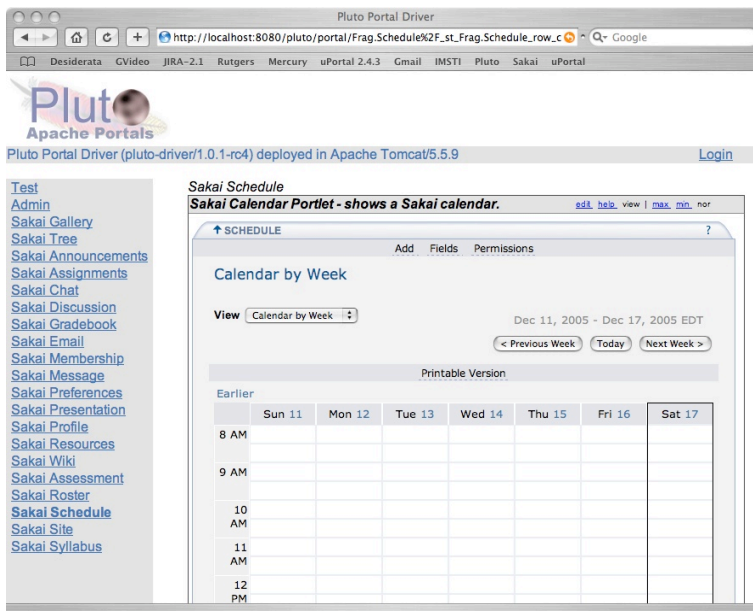
The most common application of the portlet will be to include all of Sakai under a tab in the portal. This bypasses the challenge of cross provisioning of course roster role information between the portal and Sakai.

The portlet also supports the ability to place a single Sakai tool within the Portal. These portlets are best thought of as a "proxy" for each Sakai tool. Unless the cross-provisioning problem is solved, each of these placements must be hand-placed and configured by the system administrator or portal users.

This is an example of Samigo running within a JSR-168 portlet.



This is an example of the Sakai Calendar running within a JSR-168 Portlet.



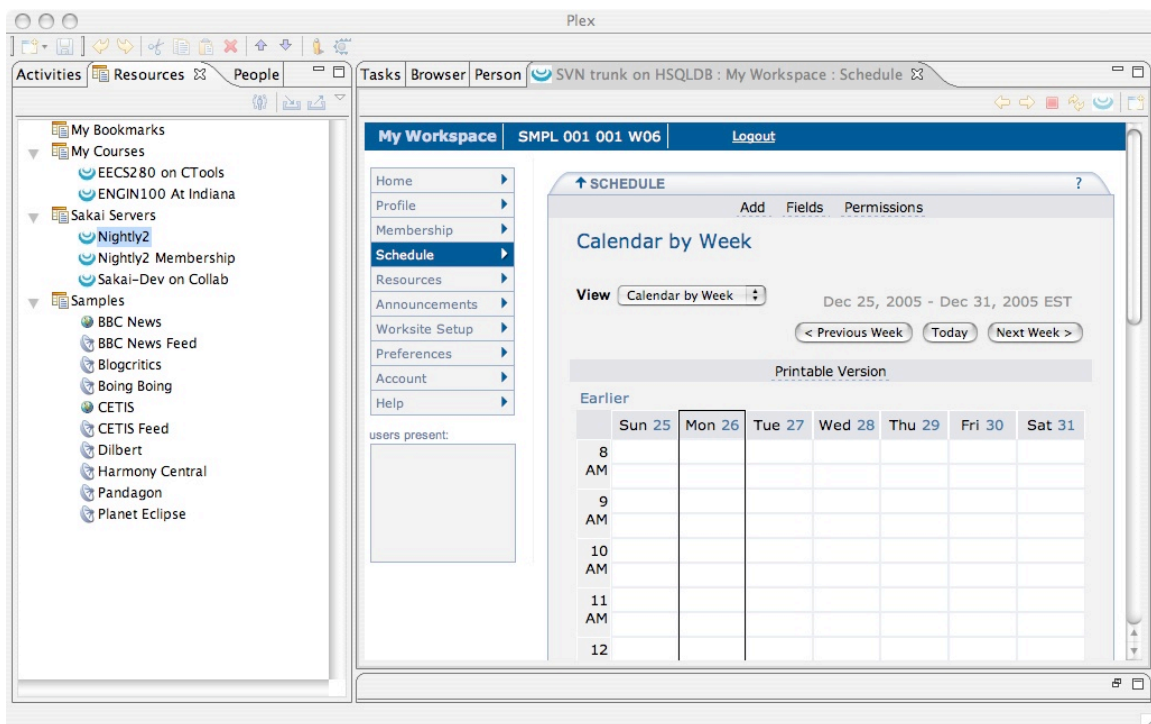
## Appendix C - Sakai in a Personal Learning Environment

This appendix shows some nascent efforts to place Sakai in a Personal Learning Environment. This shows three basic views of Sakai in a PLE.

The first example is Sakai place in the PLEX software that is being developed by a CETIS project (<http://www.cetis.ac.uk/members/ple/>).

PLEX supports People, Activities and Resources and allows the user to build and annotate organizational structures on top of those core resources.

What you are seeing is a prototype implementation adding a "Sakai" type resource. PLEX allows you to keep smart bookmarks to different Sakai servers, and even allows the creation of new resources while using Sakai that are pre-navigated to a particular tool or site within a Sakai server. PLEX also maintains credentials for the user for the different Sakai servers so that navigation is seamless between Sakai resources regardless of how the user chooses to organize their learning environment.



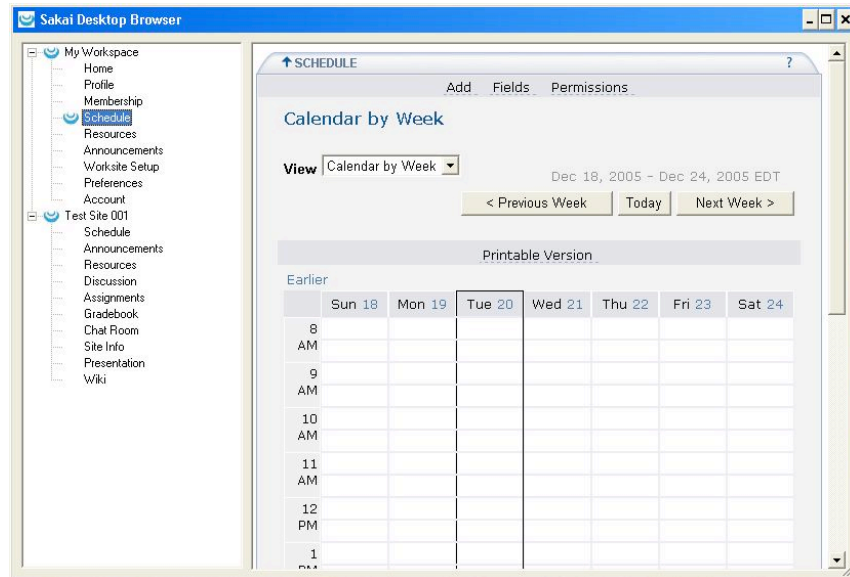
The screenshot displays the PLEX software interface. The main window is titled "Plex" and shows a "My Workspace" view for "SMPL 001 001 W06". The interface includes a navigation menu on the left with categories like "My Bookmarks", "My Courses", "Sakai Servers", and "Samples". The "Sakai Servers" category is expanded, showing "Nightly2" and "Sakai-Dev on Collab". The "Samples" category is also expanded, listing various resources like "BBC News", "Blogcritics", and "CETIS".

The main content area shows a "Schedule" view for the week of Dec 25, 2005 - Dec 31, 2005 EST. The view is set to "Calendar by Week". The calendar grid shows the following structure:

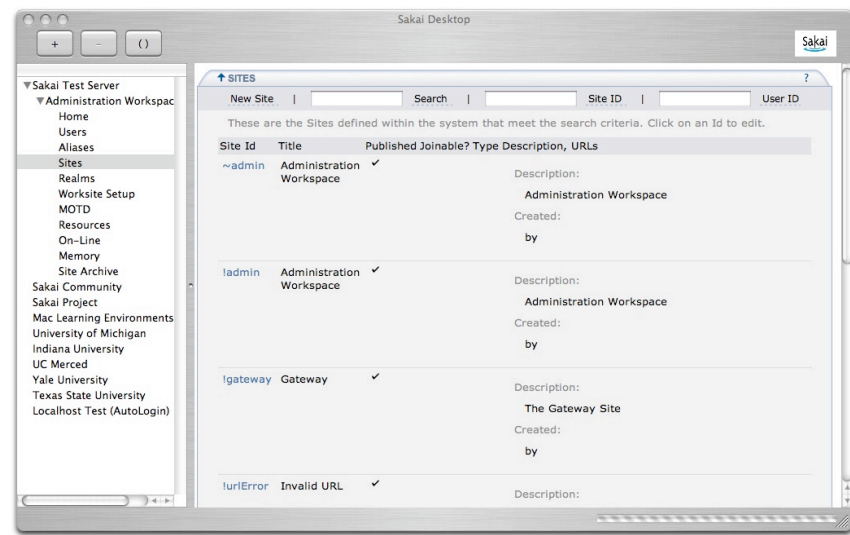
Earlier	Sun 25	Mon 26	Tue 27	Wed 28	Thu 29	Fri 30	Sat 31
8 AM							
9 AM							
10 AM							
11 AM							
12							

The interface also includes a "Printable Version" link and navigation buttons for "Previous Week", "Today", and "Next Week".

This view shows a Sakai Desktop Tool written in Visual Basic that is presenting a set of Sakai Sites and Pages in a tree view and allowing the user to navigate between them.



This is a version of the same idea written for the Apple Desktop. This handles multiple Sakai Servers and allows the user to drill down into the sites and find pages.



None of these tools are at a maturity level to replace an organizational portal in terms of the primary interface to a teaching and learning environment. But these tools *are* at a maturity level where they work well enough to make screenshots.