

White Paper Sakai Project

A Comparison between the JISC and Sakai Frameworks

April 5, 2004 - v3

Mark J. Norton

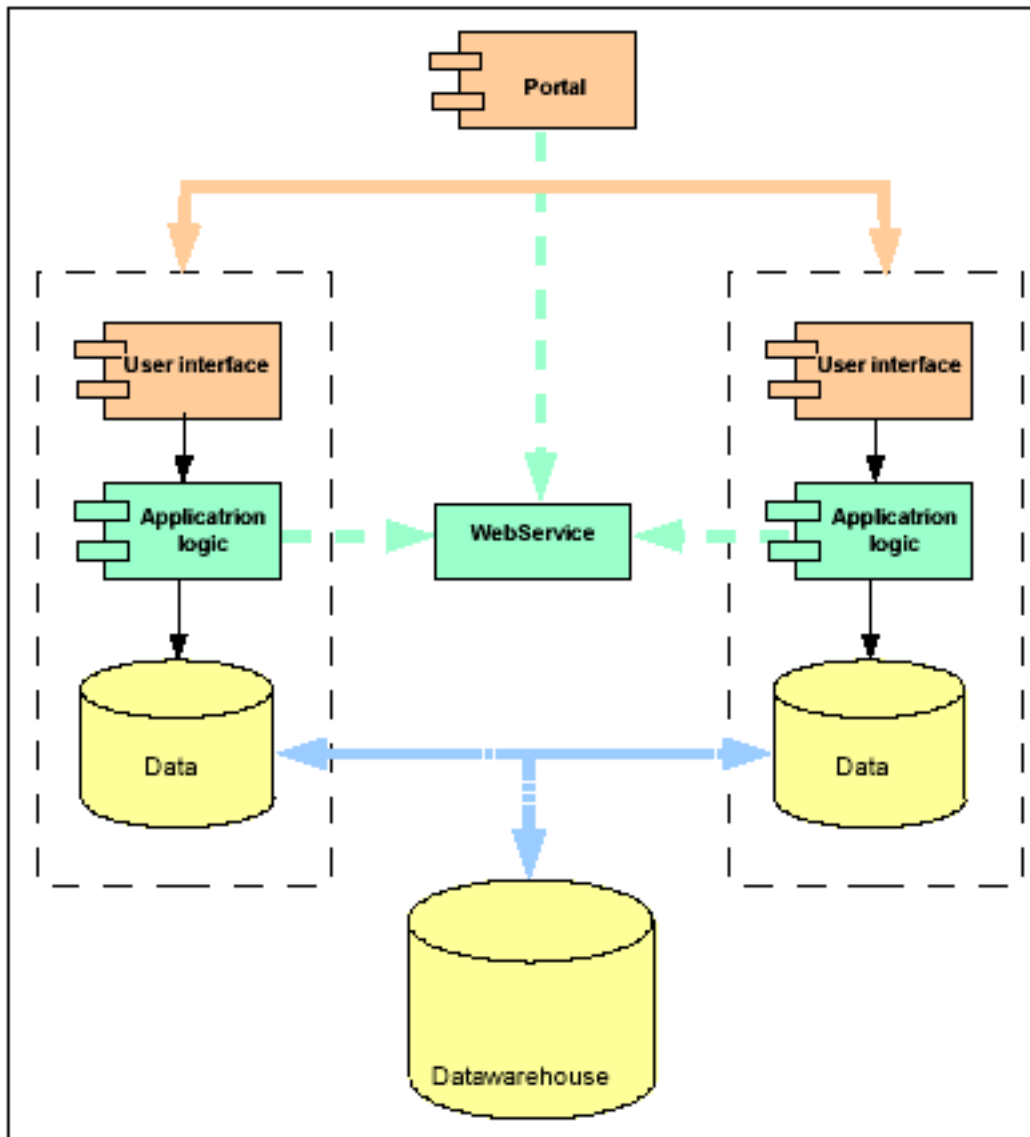
Comments to:
markjnorton@earthlink.net



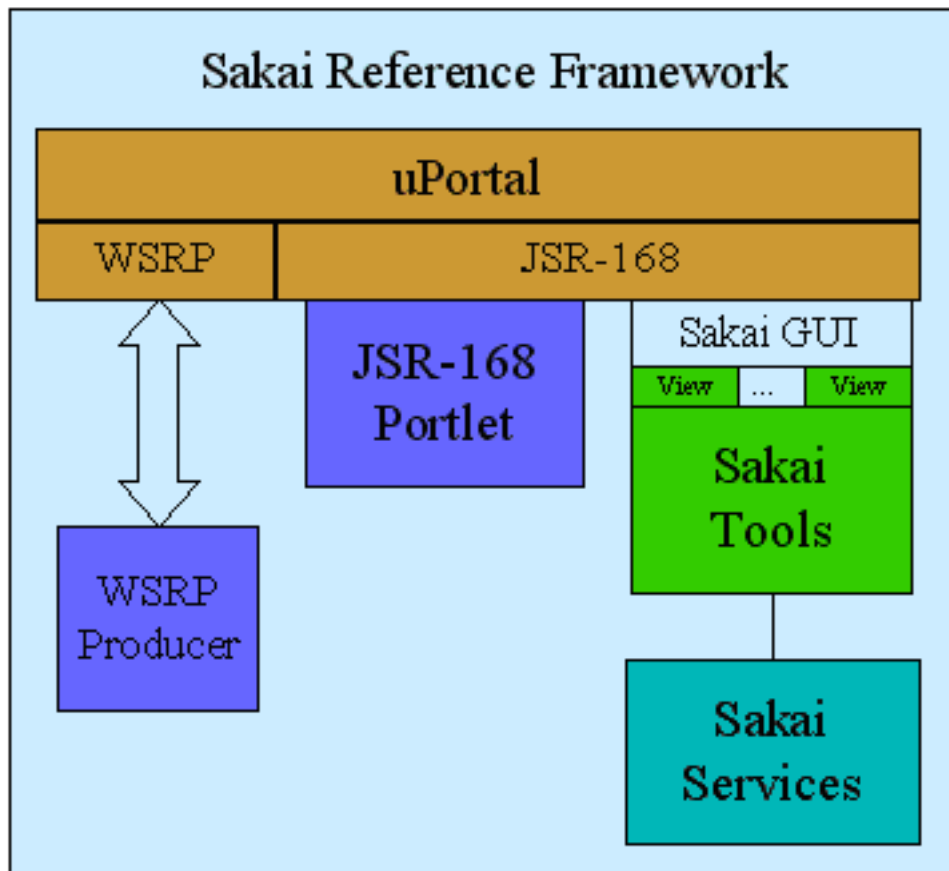
Earlier this year (March 17, 2004), Scott Wilson et al released a paper entitled, “A Technical Framework to Support eLearning”. The paper builds an excellent case for a layered architecture that avoids replicated data and fosters shared, reusable services. It is felt that this would reduce risk since modular services are largely self contained. This paper explores the differences between the Sakai and JISC Frameworks in attempt to better understand the JISC approach and to consider new ideas for inclusion in the Sakai Framework, some presented as recommendations.

The View from Near Orbit

From a 50,000 meter view, both the JISC and Sakai Frameworks look quite similar. Consider the following diagram from the JISC paper:



This diagram actually explores three different integration levels, but the elements are telling: the main point of integration is a portal system, UI logic is separated from application logic, applications can use web services, and data can be distributed or consolidated. These are pretty much the same elements in the Sakai Framework as well. This is a validation of both approaches. Consider the Sakai Framework diagram by comparison:



While this has been modified a bit to remove legacy interfaces, it shows pretty much the same things: a portal as the main integration point, GUI elements separate from the application logic and underlying services. In addition, this framework includes the notion of portlets. Web Service Report Portals (WSRP) is included to show how tools developed using other frameworks can be integrated into the Sakai portal environment.

In many ways, this is not a coincidence. The JISC framework authors are clearly in touch with industry trends in developing eLearning and collaboration systems as presented in their key concepts section of the paper. Building on prior work is essential if we are to leverage the success of the past and avoid the pitfalls that others have fallen into. Sakai shares the view of a unified solution with modular

implementation. With a little work on the part of JISC, Sakai, OKI and others, we might be able to develop a global solution which is flexible enough to include both similarities and differences. The JISC paper calls for open development, something shared by the Sakai project. The paper doesn't, however, mention any licensing terms, whereas the Sakai Open-Open license (based on the OKI license) makes open development terms very clear. Licensing issues aside, both Sakai and JISC are in a position to adopt and leverage software developed on both sides of the Atlantic. Key to interoperability between frameworks will be the use of standards and data models. The JISC paper has this to say:

Data must be presented in a format which is understood by services that are consuming it in the way that was intended by the services presenting it; this covers the way in which it is transported, the formats that are used and the vocabularies. Without standards for each of these each agreement between producing and consuming services would have to be ad hoc and this grows exponentially and rapidly become insupportable.

This is a concept that must be embraced by Sakai. Work has recently begun on shaping how data models will be defined and represented. Where ever possible, these models, data transport, use, and semantics should be based on existing industry standards. Even in those cases where there are competing standards, two are better any an arbitrary number of ad hoc solutions.

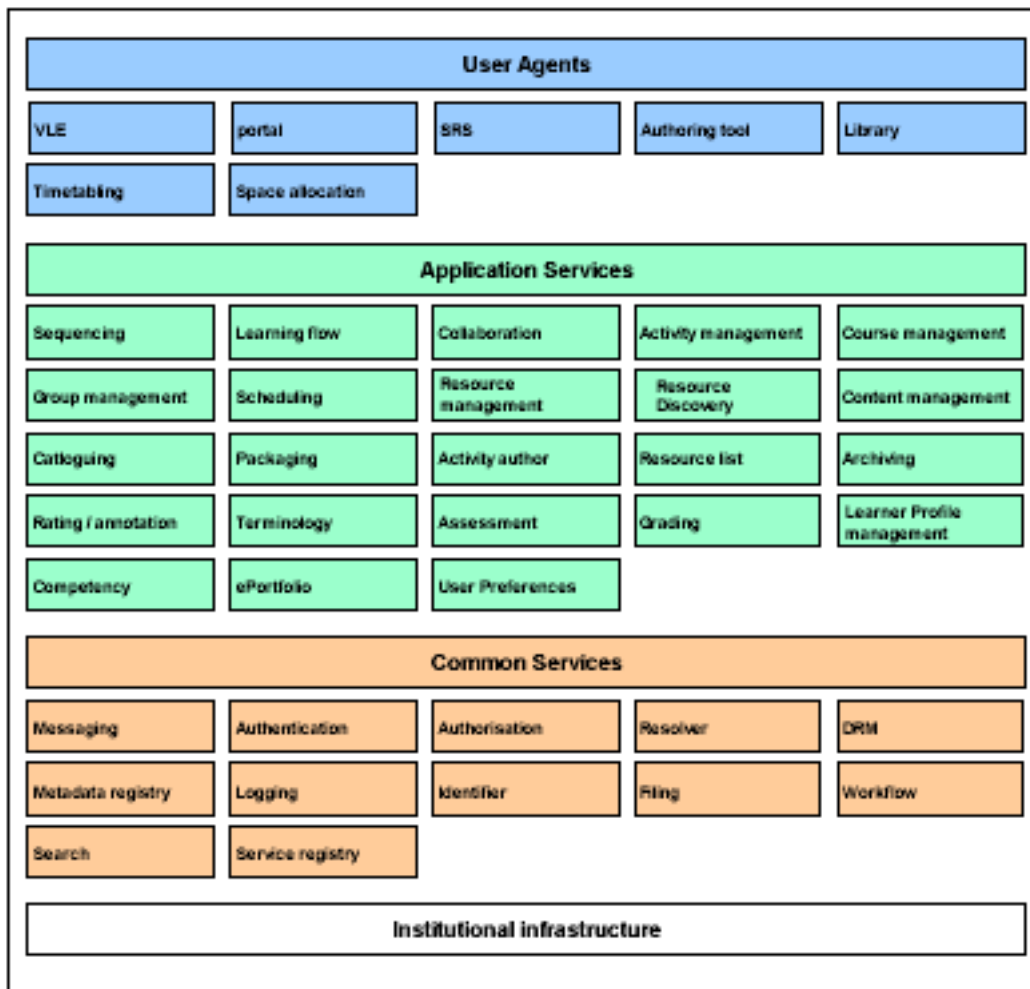
The JISC Framework presents the notion of encapsulating services in software components with web service interfaces defined by WSDL but recognize that other approaches are possible. The Sakai Framework proposes to encapsulate services in software components, likely JavaBeans. The components are combined with a set of user interface elements declared using JavaServer Faces and wrapped in a portlets. Portlets are combined into a portal page that allows control over look at feel. Presumably portlets could be distributed across servers and combined into a portal view. In terms of common services, Sakai favors a Java interface define by an OKI Open Service Interface Definition (OSID). While these could be implemented as web services, Java APIs are seen has having a lower overhead. Service distribution may be pushed lower in the stack, below the level of the OSID. Thus an authentication OSID implementation might use a shared LDAP server for campus wide single sign on. Single sign on, authentication, and authorization are also important topics for Sakai.

It is important to understand that Sakai will be a fully integrated product. As such, Sakai services are tied to tools which implement application and business logic. Sakai services manage functions and data associated with a Sakai data model. This layering of logic, services, and data allows Sakai to be internally flexible, extensible, and integrated with with other systems. These layers provide "perforation points" to

foster portability of tools and interoperability with data. This architecture results in the ability to support integration with other frameworks, such as those proposed by JISC. It also opens the doors to services and tools developed in languages other than Java. The selection of Java as the de facto implementation language simplifies the development effort and reduces deployment schedule risk. Regardless of that, I believe that the Sakai and proposed JISC frameworks are compatible.

The Layered Services Framework

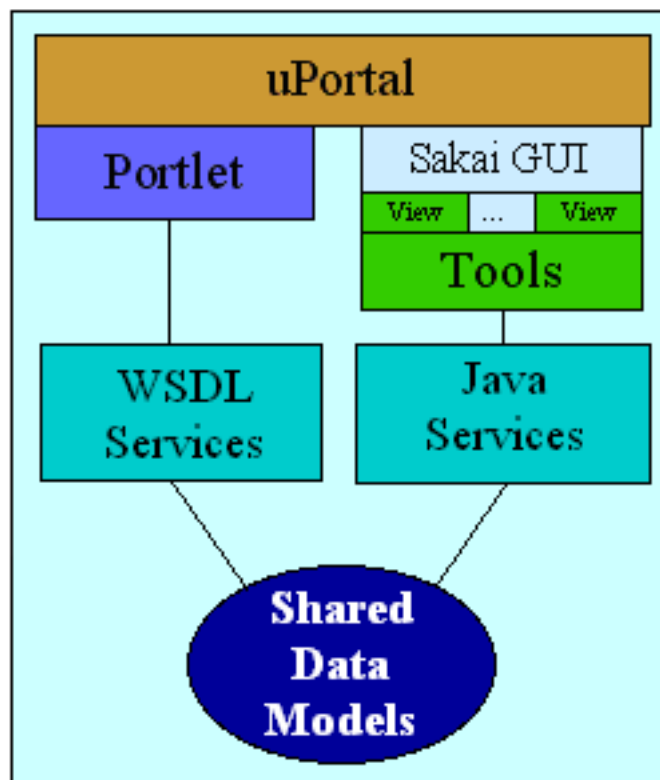
The bulk of the JISC paper is given over to a description of the various services that should be considered as part of a JISC Framework. The JISC Layered Services Framework is illustrated with the following diagram from the paper:



Before exploring these services in more detail, the idea of different kinds of user agents is one that Sakai should consider carefully. Most of the Sakai documentation assumes a web browser user agent, though handheld devices such as phones are

mentioned. Reading between the lines a bit (this is not really discussed in the JISC paper), the idea of different kinds of user agents is an important one. Consider the Authoring Tools box in the diagram above. While authoring can be done via browser interfaces, in some cases, this is extremely restrictive. Authoring can and will be done with special tools either off-line, or integrated with web services of some kind. Tools such as PowerPoint, Excel, Word, Visio, Open Office, Rational Rose, and emerging pedagogical tools such as the Tufts VUE concept mapping editor all have powerful editing paradigms. Sakai needs to carefully consider how these kinds of tools will be integrated into eLearning and collaboration. WebDAV is a good place to start, but some of the Sakai services may need to be revealed in order to enable service based tools which are client based, rather than server based.

Sakai services are roughly grouped into Common Services, which provide low level access to system and enterprise services via a shared interface, and Educational or Application Services. These application services factor out business and application logic into reusable APIs which can be combined at the Sakai tool level. All services in the proposed JISC framework are intended to be potentially distributed as web services. While that could be the case for Sakai services, most of the Application Services will likely be bundled into the Sakai Framework initially. Common Services, on the other hand, may very well be made available as web services (via WSDL, SOAP, UDDI, etc). Making common services more widely available will enable non-Java Tools to interface with Sakai.



Common Services are were most of similarity arises between these frameworks. For comparison purposes, the table below shows JISC and OKI Common Services, along with some common CHEF services.

JISC Common Services	OKI Common Services	CHEF Services
Authentication	Authentication	Authentication
Authorization	Authorization	Authorization
Logging	Logging	Logging
Messaging	Messaging	Messaging
Filing	Filing	Filing and Resources
	Scheduling	Scheduling and Calendar
Workflow	Workflow	
Search		Search
	Dictionary	Localization
Identifier	Identifier	
	Types	Roles
Resolver		
Metadata Registry		
DRM		
Service Registry		
Harvest		
	Database Connectivity	
	Hierarchy	
	SQL	
	Agent and Groups	
		Presence
		Notification
		Help
		Records
		Presentations

Sakai relies on the OKI Common Services and the JISC Framework is obviously informed by that work. Still some interesting patterns arise. Five services are common across the table: authentication, authorization, logging, filing, and messaging. In many ways, these are the no-brainers since all eLearning systems are going to need these services.

More interesting is the fact that Agent is missing from the JISC Framework. Towards the end of the paper, some discussion is made of a Group Management service. User Preferences are also included as an application service, but basic management of persons (agents) and groups is missing at the Common Services layer. Given the amount of data associated with Agents and Groups, I'd recommend that Sakai retain the Agent/Group service defined in the OKI OSIDs.

Several other services are present in OKI which seem to be missing the JISC Common Service layer. These include Scheduling, Dictionary, and Roles/Types. Again, the JISC Framework has support for event management at the application level so perhaps scheduling and calendars are included there. The dictionary service in OKI is a very limited approach to localization. It would be interesting to query the JISC authors on how they intend to handle internationalization and localization. Types play an important role across the OKI OSIDs. While it may not be necessary to call types out as a separate service (they are really just an object), agent roles are a good candidate as a separate service, if distinct from types.

Now we examine the new services proposed by the JISC Framework. Several of these are mentioned, including Resolver, Metadata Registry, Service Registry, Harvest, and Digital Rights Management (DRM). Identifiers in the JISC Framework support several standards for creating unique identifiers including Digital Object Identifiers (DOI), PURL, and Handle. The Resolver service, provides identifier resolution to find matching services, locations, and finding metadata. The Metadata Registry provides machine readable declarations of metadata schemas, mappings between metadata schemas, and human readable views of metadata schemas. This type of service will be especially useful to digital libraries and repositories. Since The JISC Framework is based on web services, a service registry is a natural service for it to provide. Should Sakai become more heavily invested in web services, it should consider a similar capability. Harvesting supports discovery of resources by copying metadata records. Finally, Sakai should strongly consider including some form of Digital Rights Management. There has been considerable progress in the standards world on DRM representation languages and is a topic of considerable importance to intellectual property vendors.

Above the common services layer is a layer which the JISC paper calls the application services and OKi calls the educational services. There is a rough correspondence to

CHEF Tool services as well. These are compared in the following table:

JISC Application Services	OKI Educational Services	CHEF Tools
Sequencing		
Learning Flow		
Collaboration		
Event Management		Event Manager
Course Management	Course Management	Course Management
Group Management		Group Management
Scheduling		Scheduler
Resource Management		Resource Manager
Cataloging		
Packaging		
Activity Authoring		
Resource List		
Archiving		
Rating and Annotation		
Terminology		
Assessment	Assessment	(Navigo)
Grading	Grading	
Learner Profile Manager		
Competency		
ePortfolio		
User Preferences		User Preferences
	Digital Repository	
		Chat
		User Controls
		Notification
		Announcements

It's at this level that the JISC factoring becomes most obvious. Many of these services translate directly in common eLearning tools: Course Management, Even

Management, Scheduling, Groups, Assessment, Grading, etc. More interesting are the new suggestions.

Sequencing is a service that arises out of JISC participation in SCORM and IMS work. IMS has produced two specification which allow sequencing of learning objects to be defined: Simple Sequencing and Learning Design. To my mind, sequencing is a valuable service, but more likely to be used by a higher level application service, such as content delivery. I would shift sequencing into a lower level and tie it to a single data model, either Simple Sequencing or Learning Design. Both have merits.

The JISC paper identifies learning flow as a service that handles complex learning scenarios over and above what sequencing handles. They indicate that these might merge at some point. I think that Sakai can safely defer adoption of this service until it is better defined.

Collaboration is where various synchronous and asynchronous tools are defined. These include chat, forums, instant messaging, etc. It also include collaborative document creation and collaboration workflow. This is an area which is still pretty much a research topic and can be deferred. The combination of IMS Learning Design and workflow has some very interesting possibilities for defining collaborative pedagogies, however.

Cataloging and archiving are concerned with maintaining and organizing information about digital assets largely through metadata. It is built on top of a metadata service, like the ones described in the common services above. These services is of concern to the DR and online library community and will have an impact on how users of Sakai interact with digital content. However, I think that this should be a library function and not part of Sakai.

Packaging provides support for getting things into and out of IMS/SCORM packages. This will be needed for any kind of migration plans for Sakai, so we should serious consider including it. However, I think it is a lower level service.

Activity Management and Authoring has to do with creating and managing Learning Design activities. This likely include the concept of assignments which is present in the OKI OSIDs, but goes further to include other kinds of activities such as scheduling an online chat session, forming a collaborative team, etc. It is capable of scheduling online, runtime capabilities. This is an area that Sakai is going to have to give some thought to. There is more than one way of organizing this kind thing. Sakai should consider creating a list of “things students do”, “things content creators do”, “things administrators do”, and see what commonality arises. The JISC Activity Manager should inform this process.

Resource Lists are largely concerned with reading lists and is the current focus of an IMS work group. I think this could be subsumed into other services. Low priority.

Rating and Annotation is also included as a JISC application service. While rating may not be all that important to Sakai, annotation might very well be, especially in a collaborative project. I also think annotation is an important part of ePortfolios. Sakai should consider including this capability.

The Terminology service provides a means to map one set of terms to another, and to define hierarchies of terms to create vocabularies, taxonomies, and ontologies. Again, this might have more relevance to the digital library crowd than directly to Sakai. Some aspects of this might be useful at low levels of Sakai, such as creating person role taxonomies and mapping them to local structures.

Learner Profile Management allows all information about a person to be organized and maintained. The JISC paper mentions that this would include personal development plans as well. Sakai needs some means to represent person data and the IMS LIP specification should be considered in the data model design.

Competencies have to do with personal skills and measuring them. This is not really in line with Sakai at this time.

The Portfolio service mentioned by the JISC paper is very much in line with what the University of Indiana folks are working on, along with r*smart and OSPI. Sakai should strongly consider what would constitute a good portfolio service.

Finally, the JISC report includes User Preferences as a service. The IMS LIP specification includes preferences in it, but doesn't really say way they are. User preferences are an important ability to allow a student (or user) to customize the environment to suit their own needs. Accessibility preferences also come up in this service. Sakai should consider having a user preference service.

Summary of Recommendations

Service	Recommendation
Agent / Groups	Go with OKI OSIDs
Resolver	Push off to a library effort.
Metadata Registry	Library effort.
Service Registry	Defer until Sakai supports web services.
Harvesting	Library effort.
DRM	Sakai should include this.
Sequencing	Sakai will need this for content delivery.
Learning Flow	Defer until better understood.
Collaboration	Defer until better understood.
Catalog / Archive	Library effort.
Packaging	Sakai will need this for content migration.
Activity Manager	Sakai will need some form of this.
Resource Lists	Merge with assignments.
Ratings	Not needed.
Annotations	Sakai should include this.
Terminology	Sakai should include low level support.
Learner Profiles	Sakai will need some form of this.
Competencies	Not needed.
Portfolio	Sakai will need some form of this.
User Preferences	Sakai will definitely need this.

Conclusion

The JISC framework paper has a lot in common with the Sakai framework, which validates both approaches. The JISC framework has factored services a bit differently from Sakai, which raises several interesting possibilities that Sakai should consider. JISC and Sakai should continue exchanges of information of this type to inform the further development of both.