

*Publisher's Note: These notes were prepared by Sakai Board Chair Joseph Hardin and distributed to the Sakai Board.*

Sakai Strategic Goals, Draft -- 11/10/2004

(1) Create a well-documented, robust framework for a collaborative learning environment, with the necessary functionality to support the development of full-featured collaborative learning environment (CLE), and enough documentation and sample tools to enable developers to extend Sakai and to demonstrate best practices for building on the framework. This framework will be called "Framework2 (F2)" for the remainder of this document. The existing legacy framework will be called "Framework1 (F1)"

(2) Produce a full-featured collaborative learning environment (CLE) sufficient in scope to replace Stellar, Ctools, OnCourse, and CourseWork. While much of the new development will be using F2, we understand that legacy tools will provide a significant portion of the functionality in the CLE at least through the 2.0 release and Fall 2005 deployments.

(3) O.K.I. intends their OSIDs to define interfaces for integrating systems like Sakai into institutional infrastructures and promote system independence. Sakai needs its system to be flexible in the same way, to provide pure portability and customizability, across institutions, systems, and time. Therefore Sakai will Develop the F2 framework based on a service oriented architectural model using OSID definitions.

(3a) Tools built to the F2 framework will be portable among Sakai F2 frameworks. This means that tools can be packaged for sharing and can be installed by Sakai system administrators with relative ease and without the need for coordination with or among software developers.

(3b) However, the F2 framework shall be flexible enough to allow institutions to choose to incorporate, with varying degrees of interoperability, tools not built to Sakai's tool best-practice specifications.

(4) Produce at least one tool that is completely free of any F1 based code and demonstrates the full capabilities of F2. (See Notes)

(5) An attempt will be made to maintain compatibility between the legacy F1 framework and the F2 framework for the duration of the project. In other words, it should not be apparent to an end user which are legacy tools and which are F2 tools. (See Notes)

(6) Sakai will produce a set of OSID 2.0 implementations that expose Sakai F2 framework services to tool developers and publish the Sakai out-of-band agreements so as to form a possible starting point for standardization activities around the recommended best-practices for an out-of-band agreement for Higher Education.

(7) OSIDs will be the mechanism to integrate enterprise systems with Sakai. For those institutions wishing to replace "out-of-the-box" Sakai service implementations with local campus infrastructure services, Sakai will provide a mechanism for OSID implementation plug in.

(8) Sakai will produce a convenience layer of language-dependent tool-facing methods which can be implemented through portable code which calls standard OSIDs. This addresses issues relating to easing the tool development process for Sakai developers. (See Notes)

(9) The Sakai service model may require capabilities in support of use cases which cannot be handled by the current 2.0 OSID definitions. These requirements will be articulated to OKI as they are encountered and close communication will be maintained between Sakai and OKI so as to insure that the OKI APIs are evolved to have improved functionality and possible new areas for API standardization based engineering experiences of the Sakai effort. The Sakai API will add these capabilities in appropriate ways.

(9a) In addition to portable implementations implementing the Sakai API using an OSID implementation 'plugin', there may be implementations provided by Sakai which directly implement the Sakai convenience layer in order to perform more efficiently.

(10) DELETED

(11) MOVED TO (9)

(12) Sakai must present its capabilities in a wide range of standards compliant Portals (WSRP and JSR-168). Sakai will co-Evolve with uPortal to insure that the WSRP and JSR-168 implementation are robust and well tested. Sakai will engage in continuous communication between the uPortal and Sakai development teams so as to exchange designs and technologies as appropriate. Areas of overlap between OSIDs and uPortal APIs will be identified with the goal of providing uPortal API implementations built on Sakai APIs and OSIDs.

(13) Sakai must demonstrate interoperability through adoption of industry standard elearning content specifications. For instance, support for the import and export of IMS Content Packages should be supported in a future version of Sakai to help achieve data exchange interoperability with other systems that support that spec.

(14) Sakai must also demonstrate interoperability through exposing its own services for consumption by external systems that wish to integrate with Sakai. At many institutions a Sakai instance will represent significant educational infrastructure that could be of value if integrated with other applications. An example of this might be a client based content authoring tool that can open and save content directly into Sakai's educational content repository.

Notes:

\* The Sakai Framework and Service Team is charged with developing detailed consensus-based requirements and designs that will be used to guide the framework development (Target #1 above).

\* The Sakai Tools Team is charged with developing detailed consensus-based requirements that will be used to guide development efforts in fulfilling the requirements of the Sakai CLE for the core schools (Target #2 above). Note that the primary driver for the definition of "sufficient in scope" in Target #1 will come from the Sakai Tools team.

\* There is a debate as to whether Samigo / Gradebook should be the "poster child" for F2 \*\* Come back to this - perhaps discuss but not decide (Target #4 above)

\* Backwards compatibility may be a problem and require significant work. It may not be practical, for instance, to update the look and feel of legacy tools to meet the requirements of the style guide. (Target #5 above)

\* Many of the gap items in 1.5 will be coded against F1 (Target #5 above)

\* The primary goal of the O.K.I. OSIDs is not developer convenience, but interoperability. It is true that many developers who use OSIDs already create convenience classes to help handle cumbersome and repetitive tasks. It is a noble goal of Sakai to create such a convenience layer to make life easier for all Sakai tool developers whose tool requirements fit with the Sakai OBAs, etc. We agree that while these goals are not mutually exclusive, they are achieved through different means (Target #8 above)