

# Understanding the Open Source Portfolio, Version 2.1, 2.2

Published: December 2005  
(Editorial Update: October 3, 2006)

Chris Coppola  
[chris.coppola@rsmart.com](mailto:chris.coppola@rsmart.com)

with significant contributions from:

Nate Angell, Portland State University  
Darren Cambridge, George Mason University  
John Ellis, rSmart  
Jay Fern, Indiana University  
Tom Lewis, University of Washington  
Wende Morgaine, Portland State University  
Dawn Nidy, DePaul University  
Hannah Reeves, rSmart  
Janice Smith, rSmart  
Tom Smith, DePaul University  
Paul Treuer, University of Minnesota Duluth  
Josh Baron, Marist College



## Table of Contents

Table of Contents .....	2
Introduction.....	3
What is the OSP?.....	5
OSP History.....	5
OSP Community .....	6
Tools for community collaboration .....	7
OSP Community Library .....	8
OSP software .....	9
Comparing OSP 1.5, 2.0, and 2.1 .....	10
Activity-based descriptions of OSP 2.1 .....	16
Detailed descriptions of OSP 2.1 .....	19
Standards and portability .....	28
Technical details .....	28
Licensing.....	28
Roadmap – OSP’s future .....	29
Get involved.....	30
Get support.....	31
Self-help resources.....	32
Community Support.....	32
Commercial Support .....	32
Appendix A - Glossary .....	33
Appendix B – OSP Forms and XML Schemas.....	37

**Editorial Note:** This whitepaper, *Understanding the Open Source Portfolio, version 2.1*, was originally published in December 2005. As a result of the integration of the OSP and Sakai communities, information regarding online resources and participation in the OSP community became outdated. To address this issue, an updated version of the document with valid reference information has been released (October 2006).

## Introduction

*Understanding the Open Source Portfolio, Version 2.1* is written to help newcomers to the community understand the OSP software, and the community that produces it. It attempts also to describe the OSP version 2.1 in sufficient detail to build a common understanding of the software, and common language to describe it among veteran members of the OSP community. While it is difficult for one document to satisfy the needs of newcomers and veterans alike, this document provides a comprehensive introduction, as well as a foundation on which the OSP community can build and extend. Hopefully that the content of this document will find its way into other ongoing forms such as improvements to the Website, documentation, and conference presentations.

*Understanding the Open Source Portfolio, Version 2.1* begins with a conceptual framework for understanding OSP in the section **What is the OSP?** which is immediately followed by a history of the project – **OSP History**. The next section, **OSP Community**, is written to help reduce the barriers to participation. It describes the tools and practices our community uses to collaborate and get things done. Some of the **OSP Community** section describes well-established tools and practices, codifies practices that have been tacit, and introduces new forms of contributions being developed to keep the OSP community strong and sustainable.

The **OSP Software** section begins with the guiding principles that shape the software. These principles will stand the test of time and help OSP stay on course over multiple releases of the software. This is immediately followed by two tables that provide a snapshot of OSP capabilities presented in a way that will help new users get a sense for what OSP offers. Also, users of previous releases can see how the 2.1 release will compare. The **Activity-based Descriptions of OSP 2.1** build on the conceptual framework introduced in **What is the OSP?** with a concise discussion of each of the key activities and how the various user roles play their part. The next subsection of OSP Software is a set of **Detailed Descriptions of OSP 2.1**. This section is organized by major groups of functionality and describes key features and functions that support each of the OSP roles.

The next section, **Standards and Portability**, describes the OSP community's long-standing commitment to inter-institutional portfolio portability, and is followed by **Technical Details** for developers about the technical architecture and frameworks used by the OSP. Finally, **OSP Licensing** rounds out the OSP Software section.

**Roadmap – OSP's future** looks beyond version 2.1 with some statements about future functionality that has been described and slated for work in future releases.

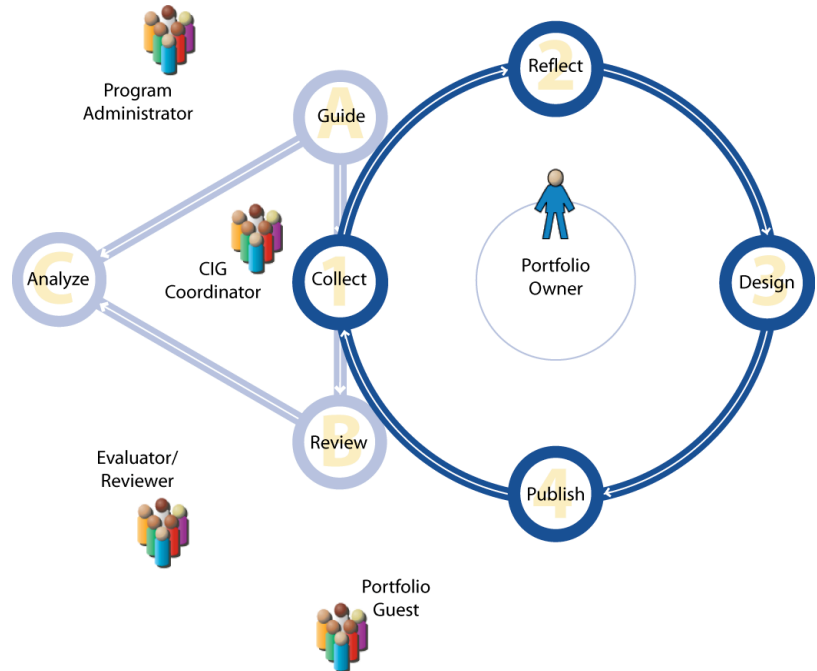
The final two sections, **Get Involved** and **Get Support**, provide a path for where to go from here and will make transparent to newcomers how to engage with the OSP community.

As both a vibrant community and world-class enterprise software, the Open Source Portfolio is poised to power the next generation of portfolio practice in higher education. Reading and referring to *Understanding the Open Source Portfolio, Version 2.1* is one of many ways for you to become involved in using and contributing to OSP. While this document describes the history of OSP, its capabilities, and the authors' ideas about how it might move forward, the future of OSP will be shaped by all of its users. We hope that you will be a part of it.

## What is the OSP?

The OSP supports portfolio-based activities by providing an environment in which a person, as a portfolio owner, is able to exhibit their work. A portfolio owner is provided tools: to collect items that best represent their accomplishments, their learning, or their work; to reflect upon these items and their connections; to design a portfolio that showcases the best selections of this work; and to publish the portfolio to a designated audience.

Tools are also available for coordinators of Common Interest Groups (CIG) or program evaluators and administrators that allow users in these roles to provide structure and guidance for portfolio owners regarding portfolio development. CIG coordinators (also referred to as Program Coordinators), evaluators, reviewers and portfolio guests are able to review published portfolios and provide formal evaluation or informal feedback and comments. Tools for analysis of portfolio items in aggregate also make it possible for CIG coordinators, administrators or program evaluators to measure program effectiveness or educational outcomes.



## OSP History

The Open Source Portfolio (OSP) initiative is an education community collaborating on the development of the best non-proprietary, open source electronic portfolio software available. The OSP began as a project in January 2003 to make the University of Minnesota's electronic portfolio system available to the world as open source software. The University of Delaware and The rSmart Group joined the University of Minnesota in this cooperative project and released the first version of the OSP in July 2003. The project had two key objectives: (1) prepare the software that had been in use at UMN for more than 5 years for other schools to use; (2) develop a community that would shape the future of OSP and sustain it in years to come.

In April 2003, the first OSP community meeting was convened at CSU Monterey Bay. Roughly 20 people attended and began to shape the future of the project with six priorities:

1. Usability and Customization
2. Assessment and Accreditation
3. Integration Interfaces
4. Scholarship of Teaching and Learning

## 5. Templates

## 6. Portability

In June of 2003, just prior to the first release, the project formalized a simple governance structure to enable community growth beyond the project founders at Minnesota, rSmart, and Delaware. Soon after the release, interested people at institutions all over the world began to download and pilot the OSP. Within a few short months the OSP community had grown from three institutions to more than 1,000 people in 77 countries. In addition, Virginia Tech joined the active development effort and began working with University of Delaware and rSmart to begin addressing some of the “Monterey Priorities.” In parallel with this new development effort, University of Rhode Island and the Carnegie Foundation for the Advancement of Teaching got involved and began to attract leading portfolio thinkers around the country and to shape the project’s direction.

Two very important things happened in December of 2003. The Mellon Foundation provided OSP with a \$518,000 grant that matched contributions from Indiana University and The rSmart Group for a total of \$1M in working funds for the next major release of OSP and to further develop a sustainable community. At the same time, the Sakai project ([www.sakaiproject.org](http://www.sakaiproject.org)) was born. Sakai began as a \$6.8M project with \$2.4M from the Andrew W. Mellon Foundation, \$1M contributions each from University of Michigan, Indiana University, Stanford, and MIT, and \$300K from the William and Flora Hewlett Foundation. By December of 2005, it had been joined by more than 100 other educational and commercial organizations. Sakai is a scholarly collaboration and learning environment (CLE) that supports the collaborative needs of teaching, learning, research, and administration.

In July of 2004, just after the release of OSP 1.5, the OSP leadership determined that the OSP would benefit tremendously from the common software infrastructure and services of the Sakai framework and the growing Sakai community. In order to realize the potential technical and community benefits, OSP 2.0 and beyond would be developed on the Sakai framework. In addition, the OSP community would work toward common practices and coordination with the Sakai community. In June of 2005, OSP 2.0 was released to the community with substantial new features for customization and assessment.

Since the 2.0 release in June, the OSP community has continually integrated more closely with the Sakai community and vice versa. The Sakai project has grown from a collaborative grant-funded project to an independent member sustained not-for-profit foundation. OSP and Sakai are mutually beneficial. OSP benefits from the rapidly maturing community, legal entity, conference planning, and more that comes from Sakai. Sakai benefits from the rich pedagogy, tools, and community that come from OSP.

## OSP Community

OSP is a volunteer-driven organization. There is no paid development staff, only the dedicated contributions of organizations like the Carnegie Foundation for the Advancement of Teaching, DePaul University, George Mason University, Indiana University, Portland State University, The rSmart Group, Syracuse University, University of Amsterdam, University of Minnesota Duluth, University of Washington, and Virginia Tech.

Leadership for the project has been provided by a governing Board of Directors consisting of representatives from the Carnegie Foundation for the Advancement of Teaching, George Mason

University, Indiana University, The rSmart Group, University of Edinburgh, University of Minnesota, University of Rhode Island, and Virginia Tech.

As the OSP project has become more closely aligned with the Sakai project, many of the responsibilities of the Board are increasingly fulfilled by the Sakai Foundation and the OSP leadership is likewise in transition. Many of us actively engaged in the leadership of the project frequently meet in person, hold conferences, and work collaboratively over the phone, via video conference, and by using the same online collaboration tools that make up the OSP and Sakai software.

The primary means of organizing OSP work is through Workgroups. Workgroups are chartered by OSP leadership with specific objectives and milestones. The team of portfolio thought leaders that developed the vision and software requirements for OSP 2.1 were organized as a Workgroup called “FX” and the development team is organized as a Workgroup.

Learn more about OSP and access a demonstration version of OSP 2.0 at: [www.osportfolio.org](http://www.osportfolio.org). (Please Note: Some of the information on this web site regarding community collaboration is outdated, please use the information below instead. The web site will be updated shortly.)

## Tools for community collaboration

Our community makes heavy use of online tools that enable collaboration and distributed work. The key tools are described below:

**OSP Website** – The OSP Website is located at [www.osportfolio.org](http://www.osportfolio.org). The website uses the Mambo open source content/community management system and is currently where you can access a demonstration version of OSP 2.0. (Please Note: Some of the information on this web site regarding community collaboration is outdated, please use the information below instead. The web site will be updated shortly.)

**Documentation** – OSP uses a Wiki called Confluence for documentation. As a result of the integration of the OSP and Sakai communities, OSP information is now located on the Sakai Confluence site (<http://bugs.sakaiproject.org/confluence/display/OSP>). We use Confluence for technical and functional documentation, collaboratively developed documentation of community practices, software design and testing documents, and more. The documentation is all available without logging in; however if you establish an account on the Sakai Confluence site (see link in “welcome” text at the top of the page) you will be able to comment and edit shared documentation as well as the additional abilities such as setting up notifications of changes.

**Newbie Tip**: A good way to learn how the system works is to review the OSP System Test Scripts and take advantage of the many associated system component test files (e.g. Forms, Layouts, Styles, etc.). When in Confluence, do a search for “OSP System Test Scripts” to locate these materials. When you are ready, completing these tests and reporting on them is a great way to help the project!

**Mailing lists, shared documents, etc.** – OSP uses the Sakai collaboration tools along with the others mentioned here to collaborate over time and distance. To join the OSP Discussion Group (DG) and associated mailing list, you will need to access the Sakai Collaboration site (also called “Collab”) at



[collab.sakaiproject.org](http://collab.sakaiproject.org). If needed, you can create a new account using the “New Account” link off the left tool bar of this site. Once you sign-in, click on the “Membership” link in the left tool bar (while in My Workspace) and then the “joinable sites” link. Then “join” the discussion group titled “DG: Portfolio”.

**Guided Tour** – The OSP community maintains a self-paced, guided tour of the OSP designed to illustrate the benefits, features, and potential of the software. The tour uses real user scenarios to explore the OSP from several different user perspectives. The guided tour is conveniently accessed from [www.osportfolio.org](http://www.osportfolio.org).

**Case Studies** – Several institutions have thoroughly documented OSP implementations as case studies using the Carnegie Foundation for the Advancement of Teaching’s KEEP toolkit. Case studies are documented at [kml.carnegiefoundation.org/OSP/](http://kml.carnegiefoundation.org/OSP/).

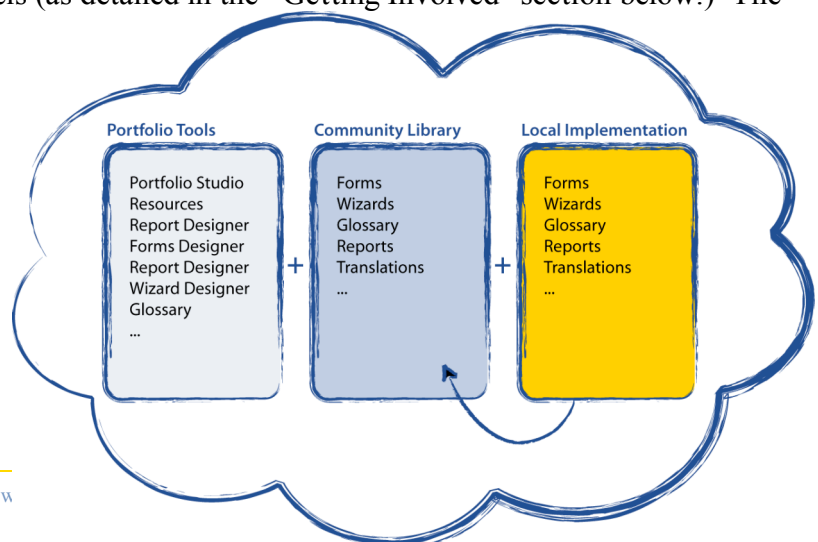
**Demo Software** – Our community maintains an online implementation of the most current production release of the OSP. Located at [demo.osportfolio.org](http://demo.osportfolio.org), the demo environment allows individuals to log in, and explore OSP as the same personae identified in the guided tour. The demo server is reset nightly to make sure everyone has an opportunity to experience the software the way it was designed.

**Issue (bug) Tracking** – The community maintains an organized log of software defects (bugs), change requests, and ideas for new features in software called Jira. While it may seem like this is a system for programmers, the software is actually easy to use and available to everyone at: [bugs.sakaiproject.org](http://bugs.sakaiproject.org). One of the most important contributions nearly everyone in the OSP community can make is taking the time to log issues and ideas for improvements in Jira. Doing so requires no special technical skills. Like the documentation, your Sakai Confluence account provides you with access to create and monitor issues.

**Source code repository** – OSP software source code is maintained in a source code repository called Subversion(SVN). Anyone may access the OSP source code from SVN using an SVN client installed on your computer or directly through a web browser at: <https://source.sakaiproject.org/svn/osp/trunk/>

## OSP Community Library

The strength of open source software lies in the community that develops and uses it. The OSP community is built on and made richer by the contributions of community members just like you. Contributions can be made a variety of levels (as detailed in the “Getting Involved” section below.) The OSP community is strengthened by the *volume* of contributions, not by the *size* of the contribution. In other words, the value of a lot of individuals making small, meaningful contributions will far exceed that of a few people making large contributions. Often newcomers to this and other open source projects will say “I wish *they* would...” There is no *they*. We are a community of like-minded





individuals with a shared mission. We can accomplish great things, and we have. But remember that when we say *we*, that includes you.

One of the primary ways OSP community members have contributed historically is by sharing the components developed for local implementation back to the OSP community. Often those templates and forms inspired new functionality, and components that build on the prior work. One contribution from one person at one institution can be the paradigm shift that revolutionizes portfolio use in higher education. That one person from that one institution making that one contribution is you. Remember - *There is no "they."*

With version 2.1, a selections of components—such as forms, wizards, layouts, and styles—will be available in the Community Library which is under development in the Sakai Confluence site. But as each institution uses OSP version 2.1, additional components for local implementation needs will be developed. Sharing those components with the rest of the OSP community—through the Community Library—is vital to the continual improvement of OSP and to a sustainable community. “They” will not be making additional components. You will. *There is no "they."*

In addition to facilitating the sharing of new components, the Community Library will provide user-friendly ways for the OSP community to rate and comment on templates, ultimately vetting them for inclusion into the basic templates available with each new version of OSP. This process of using, sharing, commenting on, rating, and ultimately developing future versions of OSP is at the heart of our community practices. We all take part in using, creating, sharing, and commenting on OSP because *there is no "they."*

In conjunction with the case studies and wiki documentations, the Community Library will additionally provide a space for the OSP community to share non-technical instructional and professional development materials, research results, and implementation experiences, which may also inspire us to create new components and functionality.

## OSP software

The objectives of the OSP are a direct response to the needs of the academic community to provide educational tools geared toward supporting both the individual and the institution. While the portfolio is designed for the primary purpose of supporting an individual’s work, the framework within which an individual can be supported is suggested and in many cases prescribed by the academic institution.

The objectives below describe what the OSP software hopes to achieve in response to the needs of the academic community.

- Establish an environment that enables an individual to present and reflect upon original, authored work in a manner that facilitates multiple modes of learning, is readily accessed and easy to use
- Provide means for individuals to discover and articulate relationships within and between their works within a collection and to discover patterns within those relationships that help them make sense of the whole.
- Enable owners to control the visual and navigational design of a portfolio of authored work

- Protect the intellectual and personal property rights of individuals who present work through the software
- Provide quality information about learning outcomes and rubrics to individual portfolio owners, course instructors, degree program administrators, and institutional administrators
- Provide access at the institutional, departmental, and course level to assessment data crucial to evaluating the strengths and weaknesses of efforts to promote learning and participation in the credentialing and accreditation processes
- Provide information to learners and teachers seeking to understand the learning process
- Provide a tool for faculty and staff development, as well as the hiring, promotion, and tenure processes
- Offer default components based on best practices for gathering content, managing workflow, and presenting content for a variety of professional uses
- Enable collaboration and exchange of ideas among members of a portfolio community in response to presentation of work
- Provide for seamless lateral movement of content between institutional departments, as a transition between roles within an institution, and in moving between institutions, and in managing relationships with multiple institutions simultaneously
- Provide an environment built in accordance with national and international standards
- Interoperate with human resource, student information, course management, learning management, and other enterprise-wide systems
- Provide an environment for students to display their work to potential employers

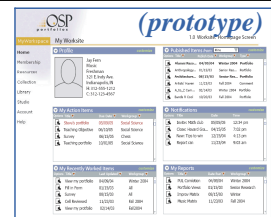
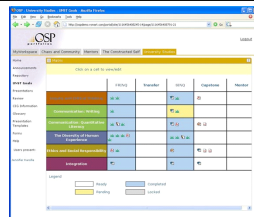
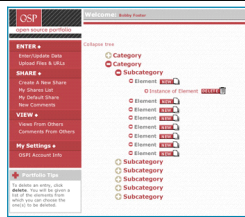
## Comparing OSP 1.5, 2.0, and 2.1

The open source portfolio (OSP) version 2.1 is a substantial new release that builds on the strong foundation of OSP 2.0 and brings forward unique capabilities from OSP 1.x that didn't get included in the first 2.x release. The following table summarizes OSP 2.1 in the context of the portfolio activities described above, and from the perspective of both OSP 2.0 and OSP 1.5. The table uses terminology that may be unfamiliar to many readers. Please use the glossary and other sections of this document to complement the use of the following table.

# OSP 1.5

# OSP 2.0

# OSP 2.1



Collect	o	~	~
Reflect	o	o	~
Design	TM	o	?
Publish	o	o	?
Guide	?	o	~
Review	o	o	?
Analyze	o	TM	?
Customization	o	?	~
Portability	o	TM	?
Integration	o	?	?
Localization	TM	o	?

Table 1 - Big picture comparison

The following detailed comparison in Table 2 below expands on the snapshot in Table 1 above with details about capabilities comparing OSP 1.5, 2.0, and 2.1.

Activity	OSP 1.5	OSP 2.0	OSP 2.1
<b>Collect</b>	Re-usable elements and files	Web-based file system with WebDAV to store and organize Items—very flexible re-use of files and Forms.	Everything from 2.0 +  Combined with Sakai Resources for ease of sharing with CLE tools. Wizard guided collection. Copyright management.
<b>Reflect</b>	Hard-coded into some elements.	Guided reflection in Matrix. Limited customization.	Highly customizable guided reflection may be triggered in Wizards.
<b>Design</b>	Customizable Presentation Templates. Limited user design choices.	Customizable Presentation Templates. Limited user design choices.	Portfolio Studio, Custom Layouts and Styles, Portfolio Templates, Multiple page portfolios, Internal and external hyperlinking, End-user creative freedom.
<b>Publish</b>	Share Hierarchy (selective), share with system users, by email address, by public Web address.	Publish to CIG members by name/role, publish to Web (public), publish to any email address.	Same as 2.0 + Publish to any system users.  Offline copies of Portfolios may be saved to removable media.
<b>Guide</b>	Fixed Hierarchy, Elements, and Tips	Matrix, Forms, Presentation Builder, Glossary	Hierarchy, Sequential, and Matrix Wizards, Forms, Guidance, Portfolio Studio, Glossary

Activity	OSP 1.5	OSP 2.0	OSP 2.1
<b>Review/Evaluate</b>	Commenting	Commenting, Limited Review/Evaluation	Commenting on Portfolios, Custom Review Forms in Wizards, Custom Evaluation Forms in Wizards.
<b>Analyze</b>	System utilization reports		Advanced reporting engine, Report Builder uses XML report definitions, data warehouse, Snapshot/Export, Included reports
<b>Non-functional characteristics</b>			
<b>Customization</b>	Hierarchy/Elements customizable at implementation time. Custom Presentation Templates	Skins, Re-usable Forms, Multi-site organization, Limited Matrix/Evaluation customization, Customizable Glossary, Presentation Templates	Everything from 2.0 + Wizards, Forms used for reflection, comments. Wizards used for Evaluation. Highly customizable Matrix, Portfolio Layouts & Styles, and Reports.
<b>Portability</b>	Save human-viewable snapshot to CD	Forms may be developed that map to XML standards	Comprehensive baseline set of re-usable, portable forms mapped to IMS ePortfolio Specification.  Offline copies of Portfolios may be saved to removable media.
<b>Integration</b>	Custom	Re-usable integration interfaces, CMS Integration	Re-usable integration interfaces, Out of the box Sakai CLE/CMS Integration

Activity	OSP 1.5	OSP 2.0	OSP 2.1
<b>Sakai Integration</b>		One release back, duplicative tools	Synchronized release, better tool integration
<b>Localization (Internationalization)</b>	Required substantial code changes	A bit more separation (MVC).	Better use of resource bundles to support reusable translations.
<b>Re-use of components</b>		Export/Import for Forms, Matrices, Presentation Templates	Export/Import for: Styles, Layouts, Forms, Wizards, Glossary, Portfolio Templates, Language Resource Bundles  Community library of components.
<b>Accessibility</b>	Unknown.	<p>The Sakai and OSP communities have made accessibility a priority. The Sakai Accessibility Style Guide provides consistent direction to exceed the minimum and recommended standards of the Web Content Accessibility Guidelines (WCAG 1.0). The community is currently conducting a rigorous evaluation to determine progress toward Section 508 requirements and WCAG priorities. The following adaptive technology products have been found to be compatible with Sakai: JAWS 6.1, Window-Eyes x. The following evaluation tools were used to assess Sakai compliance: A-Prompt, WebXACT, W3C Validator.</p> <p>Although there are still unknowns, and work underway, the community has shown significant progress and an ongoing commitment to accessibility.</p>	

Activity	OSP 1.5	OSP 2.0	OSP 2.1
<b>Usability</b>	Highly intuitive, easy to use. Activity based.	Easy and efficient to use after initial learning curve. Entirely tool based.	Implementing an activity-based UI for improved intuitiveness. Substantial addition of guidance. Many usability enhancements informed by structured usability testing.

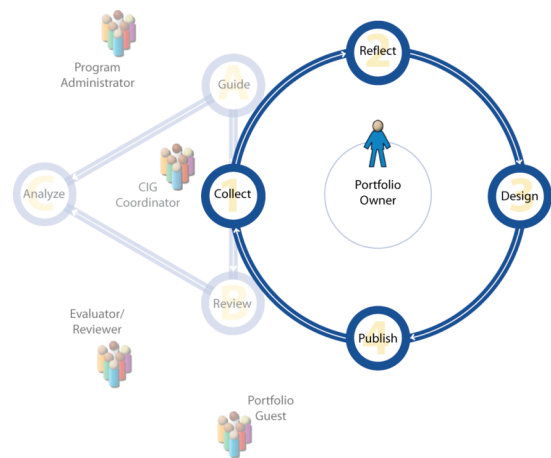
**Table 2 - Detailed Comparison**



## Activity-based descriptions of OSP 2.1

### Collect

Each Portfolio Owner has a personal filing system of folders and sub-folders which they use to collect and organize any digital items they choose. These items may be files such as audio, video, and documents, or they may be information collected in a web-based Form they have completed within the portfolio system. OSP 2.1 uses the Resources Tool to collect, organize, and access items. Once items are stored in the Resources Tool, the Portfolio Owner may access them from anywhere within the system and reference them in any number of activities.



Wizards facilitate the construction and organization of information by providing a Portfolio Owner with guidance and structure. In OSP 2.1, Wizards may present the Portfolio Owner with activities as a linear sequence of steps, in hierarchical form, or as a set of rows and columns in a Matrix. Wizards provide important guidance, rationale, and examples which can be in the form of rich formatted text, and associated files of any type.

### Reflect

Often Portfolio Owners will reflect on various aspects of their work, thinking carefully about relationships between learning experiences. OSP version 2.1 supports reflective activities throughout a Portfolio Owner's experience by providing tailored reflection Forms and guided reflection activities (through Wizards) that encourage and guide the Portfolio Owner through reflective activities. Virtually every component of OSP can be used reflectively. For example, owners may reflect through the visual relationships they compose between items on a page.

### Design and Publish

Another key group of activities for Portfolio Owners is selecting from a diverse collection of items and designing a Portfolio that presents those items to a particular audience for a particular purpose. The OSP system provides a rich set of tools that give the Portfolio Owner creative freedom to represent herself in different ways over time to various audiences. The Portfolio Studio is where a Portfolio owner creates and manages her Portfolios. Any Portfolio Owner can create new Portfolios. New Portfolios may be created from pre-defined templates, or by using the Portfolio Designer. Pre-defined Portfolio Templates control layout and style for consistency but allow the Portfolio Owner to select appropriate content and customize various aspects of the presentation. The Portfolio Designer gives the Portfolio Owner much more creative freedom to design multi-page Portfolios based on a customizable selection of layouts and styles.

Portfolios may be published and shared with others in 4 ways:

1. Portfolios may be published to other users of the portfolio system by role or by name.
2. Portfolios may be published to anyone by email address.
3. Portfolios may be published to a publicly accessible URL that anyone can see using a Web browser.
4. Portfolios may be published to removable media such as CD-ROM, DVD, or USB drives.

## Guide

Program Administrators and CIG Coordinators use the OSP system to provide Portfolio Owners with essential guidance throughout the Portfolio Owner's portfolio experience. The OSP 2.1 includes a rich set of tools to provide guidance in the form of structure, content, and process. The most granular building blocks of guidance Program Administrators and CIG Coordinators can provide for Portfolio Owners are Forms. Forms are a collection of fields that provide a space for user entry of data. Forms can be designed to collect virtually any kind

of information desired, and they contain instructions to help users complete them. Forms can also be arranged in Wizards to provide higher level structure and process that facilitates the completion of multiple Forms. These wizards can be organized as a linear sequence, a hierarchical organization, or a set of rows and columns that form a matrix. Guidance associated with Wizards can include rich text instruction, rationale, and examples as well as file attachments of any type.

Pre-defined portfolio templates, and the layouts and styles in the Portfolio Designer also provide guidance for the Portfolio Owner.

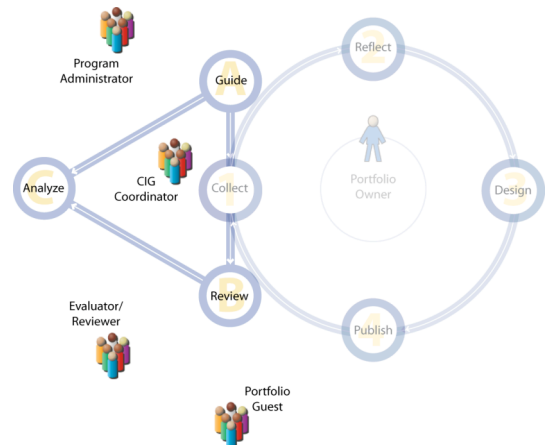
Throughout the OSP system, terms defined in the OSP Glossary are highlighted for the user providing quick and easy access to the definitions. Terms in the OSP Glossary may be defined system-wide, and terms may be overridden in a particular CIG.

## Review and Evaluation

Portfolio Reviewers, Evaluators and Guests interact with Portfolio Owners by providing Portfolio Owners with formative feedback, formative and summative evaluations, and comments. CIG Coordinators and Program Administrators are able to customize comment, feedback, and evaluations by designing Forms and Wizards to be used for those activities.

## Analyze

OSP 2.1 introduces completely new reporting functionality to support analysis of portfolio data. The OSP reporting engine is based on an XML report definition that articulates the source and format of data, and the style and layout of the report. Report definitions may have multiple style and layout options that accommodate, for example, the different needs of print and screen formats.



To generate reports, Program Administrators and CIG Coordinators will use the OSP Report Designer. The Report Designer allows selection of a report definition and prompts the user to fill-in and select parameters. Reports may then be saved to run again with the same parameters, or saved in a snapshot form which will preserve the data and still provide layout and style options which can be chosen when the report is viewed. The results of any report may be exported to a file suitable for import into a spreadsheet program such as Open Office or MS Excel.

## Detailed descriptions of OSP 2.1

This section describes the capabilities planned for OSP 2.1. Keep in mind that these descriptions reflect current plans and may change between the time this document is written, and the actual release of OSP 2.1.

### OSP Forms

In OSP 2.1 Forms are the essential building blocks on which all other functionality relies. They are the smallest re-usable unit of information in the system. Forms are a collection of one or more optional or required fields which can be published to Portfolio Owners who then fill in those fields and save the resulting item in their Resources for later use. Portfolio Owners may create any number of specific Items from a Form to capture different information (i.e. Multiple Items from a “Work History” Form that contain specific information about one’s work history).

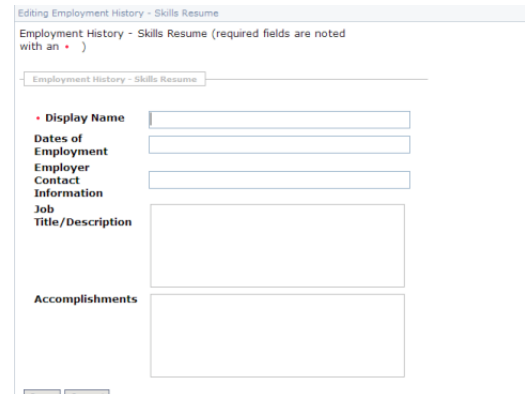


Figure 1 - Sample Form

Forms may contain fields as described in Table 3 below. The fields in Table 3 may be arranged in groups, be required or not, have length restrictions, and permit multiple occurrences to allow users to add one or more of a particular set of fields (i.e. a home phone, cell phone, and work phone).

Field Type	Description
<b>Text</b>	Simple string of letters, numbers, and symbols presented to the user on a single line
<b>Text Area</b>	Text of any length presented to the user on multiple lines
<b>Integer</b>	Input of integer values
<b>Rich Text</b> <sup>(new)</sup>	Text of any length presented to the user with the standard text-editing toolbar for style and formatting
<b>Date</b>	Day, month, year presented as fill-in blanks and also a calendar date selection pop-up
<b>Select</b>	User is prompted from a fixed list of choices to select one from a drop-down list
<b>Boolean (yes/no)</b>	User is prompted with a checkbox to either select or de-select
<b>File Attachment</b> <sup>(new)</sup>	Ability to associate any file from one’s resources with a Form

Table 3 - Supported Field Types

OSP Forms are based on XML Schemas. The best way to learn more about OSP Forms is to understand XML Schemas. See Appendix B – OSP Forms and XML Schemas.

In addition to the fields and other characteristics previously described, Forms can either inherit the default system style, the default style for a particular CIG, or the style of a Wizard.

The following table illustrates key features and functions of OSP Forms from the perspective of each role:

User Role	Feature Descriptions
<b>Portfolio Owner</b>	<ul style="list-style-type: none"> <li>• Completes Forms presented through Wizards, and other portfolio related activities;</li> <li>• Stores, organizes and retrieves completed Forms (Items) in her repository;</li> <li>• Views Forms completed by others that represent comments, feedback, and evaluations;</li> </ul>
<b>Program Administrator</b>	<ul style="list-style-type: none"> <li>• Designates global Forms;</li> <li>• Approves Forms submitted for global publishing;</li> </ul>
<b>CIG Coordinator</b>	<ul style="list-style-type: none"> <li>• Creates Wizards that incorporate the use of Forms;</li> <li>• Submits Forms to be published globally;</li> </ul>
<b>Reviewer</b>	<ul style="list-style-type: none"> <li>• Uses “Review Forms” to review Portfolio Owner progress and provide feedback;</li> <li>• Views the information content of completed Forms in Portfolios and Wizards;</li> </ul>
<b>Evaluator</b>	<ul style="list-style-type: none"> <li>• Uses “Evaluation Forms” to review Portfolio Owner progress and provide feedback;</li> <li>• Views the information content of completed Forms in Portfolios and Wizards;</li> </ul>
<b>Guest</b>	<ul style="list-style-type: none"> <li>• Uses “Comment Forms” to comment on shared Portfolios;</li> <li>• Views the information content of completed Forms in Portfolios;</li> </ul>
<b>System Administrator</b>	<ul style="list-style-type: none"> <li>• Imports Forms for use by Program Administrators, and CIG Coordinators</li> </ul>
<b>Instructional Technologist</b>	<ul style="list-style-type: none"> <li>• Constructs custom Forms using XML Schema Definitions (XSD)</li> </ul>

**Table 4 - Role-based features for OSP Forms**

## OSP Glossary

The OSP 2.1 Glossary provides a set of terms or phrases and definitions that can be set for the entire system, and inherited or overridden in a particular CIG. CIG Coordinators (also called a Program Coordinator) and Program Administrators define terms in the Glossary which are then highlighted to users as hyperlinked items wherever they appear in appropriate areas of the OSP system including: Wizards, Reports, and Designing Portfolios. The OSP Glossary supports a short definition which appears to the user when he/she moves the mouse cursor over a defined term. The Glossary also supports a longer definition which is presented to the user as a pop-up window when the hyperlinked term is selected.

User Role	Feature Descriptions
<b>Portfolio Owner</b>	<ul style="list-style-type: none"> <li>• Hovers mouse cursor over defined terms to see short definition</li> <li>• Selects hyperlink of defined terms to see complete definition</li> </ul>
<b>Program Administrator</b>	<ul style="list-style-type: none"> <li>• Defines global glossary terms</li> </ul>
<b>CIG Coordinator (also called a Program Coordinator)</b>	<ul style="list-style-type: none"> <li>• Defines unique CIG glossary terms and rendering preferences (first/all occurrences)</li> <li>• Overrides global glossary terms if necessary</li> </ul>
<b>Reviewer</b>	<ul style="list-style-type: none"> <li>• Hovers mouse cursor over defined terms to see short definition</li> <li>• Selects hyperlink of defined terms to see complete definition</li> </ul>
<b>Evaluator</b>	<ul style="list-style-type: none"> <li>• Hovers mouse cursor over defined terms to see short definition</li> <li>• Selects hyperlink of defined terms to see complete definition</li> </ul>

**Table 5 - Role-based features for OSP Glossary**

## Resources

The Resources tool in OSP 2.1 is the result of combining the Sakai 2.0 Resources tool with the OSP 2.0 Repository tool. OSP 2.1 and Sakai 2.1 now share a common tool for storing and organizing Items.

Resources provides Portfolio Owners with a personal file system. The Portfolio Owner's Resources are conveniently accessible anywhere throughout the OSP and Sakai system. Similar to the file systems we're accustomed to on our Windows, Mac, or Linux desktop, the Resources tool provides the virtual place to store, organize, and access electronic resources such as documents, audio, video, URLs, and Items that result from completing OSP Forms.

Resources also provides Common Interest Groups with a common file space to share Items among members of the group.

User Role	Feature Descriptions
<b>Portfolio Owner</b>	<ul style="list-style-type: none"> <li>Creates/Modifies/Deletes Items (files from local computer, URLs, HTML content) in the repository</li> <li>Set copyright and access permissions for Items</li> <li>Organize Items into folders and sub-folders</li> <li>Retrieve Items by browsing the folders and sub-folders of Resources either through the Web interface, or using a local desktop interface such as the Windows Explorer, or Mac Finder (WebDAV)</li> <li>Select specific Forms to complete directly from Resources</li> </ul>
<b>Program Administrator</b>	<ul style="list-style-type: none"> <li>Selects Styles, Layouts, and other components used to import and/or design Forms, Wizards, Portfolio Templates, and more</li> </ul>
<b>CIG Coordinator (also called a Program Coordinator)</b>	<ul style="list-style-type: none"> <li>Creates/Modifies/Deletes Items (files from local computer, URLs, HTML content) in the shared repository for the CIG</li> <li>Selects Styles, Layouts, and other components used to import and/or design Forms, Wizards, Portfolio Templates, and more</li> </ul>
<b>System Administrator / Instructional Technologist</b>	<ul style="list-style-type: none"> <li>Adds Items to CIG repositories used to create Forms, Portfolios, Wizards, etc.</li> </ul>

**Table 6 - Role-based features for Resources**

## Wizards

Wizards are new in OSP 2.1. They provide structure and guidance to inform the Portfolio Owner's experience. A Program Administrator or CIG Coordinator uses the Wizard Designer to create a Sequential, Hierarchy, or Matrix Wizard which the Portfolio Owner will then use to complete Forms, associate existing Forms and Files, Reflect, and receive Feedback and Evaluation. A Portfolio Owner will generally access Wizards in a particular CIG, however the Portfolio Owner may also create their own Wizards in their Personal Workspace.

Wizards contain pages which are presented to users. Each of the three types of Wizards (sequential, hierarchy, and matrix) has a unique way of organizing and presenting the pages, but pages in all Wizards are essentially the same. A Wizard Page can contain Guidance, Forms, Feedback, Evaluation, and Items which the Portfolio Owner has associated (Files and completed Forms). A Sequential Wizard presents pages in a linear sequence, a Hierarchy Wizard organizes pages into categories and sub-categories, while a Matrix organizes pages into cells which represent the intersection of a series of rows and columns.

Any user with the Reviewer role may provide Feedback to the Portfolio Owner for each page in a Wizard. Feedback is provided in the form of rich text. Evaluators explicitly assigned when the Wizard was designed, can use an Evaluation Form to evaluate the Portfolio Owner's work. Evaluation Forms can be consistent for an entire CIG, Wizard, or customized for each Wizard Page. When a Wizard Page is designed to include Evaluation, the Portfolio Owner is able to submit the page for evaluation. Once submitted, the page cannot be altered to ensure the integrity of the review.

All Wizards track Portfolio Owner activities such as start time, and status which can be used in Reports. The contents and attributes of Wizards may also be used in Portfolios to share them with other Portfolio Owners, Reviewers, and Guests.

Table 7 describes the key features common to all Wizards:

User Role	Feature Descriptions
<b>Portfolio Owner</b>	<ul style="list-style-type: none"> <li>Views Wizard Instruction, Rationale, and Examples</li> <li>Interacts with Wizard navigation to complete Wizard objectives:               <ul style="list-style-type: none"> <li>Filling in Forms</li> <li>Reflecting</li> <li>Submitting for Evaluation</li> <li>Creating Portfolios (* potential for 2.1)</li> </ul> </li> </ul>
<b>Program Administrator / CIG Coordinator</b>	<ul style="list-style-type: none"> <li>Designs Wizards for Portfolio Owners to interact with               <ul style="list-style-type: none"> <li>Select Forms</li> <li>Select Style</li> <li>Create Guidance (Instruction, Rationale, and Examples)</li> <li>Sets other Wizard attributes as appropriate</li> </ul> </li> </ul>
<b>Reviewer</b>	<ul style="list-style-type: none"> <li>Review's Portfolio Owner's work and provides rich text feedback</li> </ul>
<b>Evaluator</b>	<ul style="list-style-type: none"> <li>Uses Wizard's Evaluation Form to Evaluate Portfolio Owner's work</li> <li>Browses Evaluation To-Do's</li> <li>Submits Evaluation Forms</li> </ul>

**Table 7 - Role-based features for Wizards**



OSP 2.1 supports three types of Wizards: Sequential, Hierarchy, and Matrix. Each type of Wizard is described in more detail in the following sections.

### Sequential Wizard

A Sequential Wizard provides the Portfolio Owner simple navigation through a linear sequence of Wizard Pages. The Portfolio Owner is first presented with an introduction page that contains basic information like the title and description, and any Guidance placed on the introduction page. The Sequential Wizard presents a single page at a time to a Portfolio Owner, with navigation that allows movement forward and back, as well as skipping directly to any page in the sequence. Each page may present the Portfolio Owner with Guidance, Forms, Items, Feedback, and Evaluation.

Sequential Wizards deal with Evaluation on the Wizard as a whole, versus individual Wizard Pages. Therefore, if the Sequential Wizard is designed with Evaluation, the Wizard will have a *submit* step where the Portfolio Owner is expected to submit their work for Evaluation. Sequential Wizards may also have a ‘create Portfolio’ capability if the Wizard designer associated a Portfolio Template with the Wizard<sup>1</sup>.

User Role	Feature Descriptions
<b>Portfolio Owner</b>	<ul style="list-style-type: none"> <li>• Navigates sequentially through the Wizard pages</li> <li>• Views Guidance pages</li> <li>• Creates new Items from Forms</li> <li>• Selects pre-existing Items from repository which match the allowed Form type for a page</li> <li>• Edits selected Items</li> <li>• Submits Wizard for Evaluation</li> <li>• Publishes a Portfolio directly from the Wizard based on the designated Portfolio Template (* potential for 2.1)</li> </ul>
<b>Program Administrator / CIG Coordinator</b>	<ul style="list-style-type: none"> <li>• Designs Sequential Wizards for Portfolio Owners to interact with               <ul style="list-style-type: none"> <li>• Arranges pages of Forms and Guidance in sequential order</li> <li>• Selects Evaluation options and Form</li> <li>• Selects Feedback options and Form</li> </ul> </li> </ul>
<b>Reviewer</b>	<ul style="list-style-type: none"> <li>• Browses a specific Portfolio Owner’s Sequence by selecting the Portfolio Owner by name</li> <li>• Provides Feedback on a Wizard Page (rich text area)</li> </ul>
<b>Evaluator</b>	<ul style="list-style-type: none"> <li>• Evaluates Wizard using the designated Evaluation Form for the Wizard</li> </ul>

**Table 8 - Role based features for Sequential Wizard**

<sup>1</sup> This functionality may make it into the 2.1 release, but isn’t certain to be in scope.

## Hierarchy Wizard

A Hierarchy Wizard is an organizational structure for Wizard Pages which is presented to the Portfolio Owner like an outline. Each item in the outline may either be: 1) a category which contains other categories or pages; or 2) Wizard Pages which present Guidance, Forms, Items, Feedback, and Evaluation. CIG Coordinators and Program Administrators design the organization of categories, sub-categories, and pages to convey relationships among the information, and a navigational structure for Portfolio Owners using the Hierarchy Wizard. Portfolio Owners interacting with the Hierarchy Wizard use the outline to navigate between categories, and to Wizard Pages. Once on a page, the Portfolio Owner is presented with Guidance (if present), Forms to be completed, Items which she has already associated with the page, and any Feedback that has been given. The Portfolio Owner may create new Items from the Forms, edit or remove existing Items, and select Items from her Resources that match the required Form types.

User Role	Feature Descriptions
<b>Portfolio Owner</b>	<ul style="list-style-type: none"> <li>Views the 'home' screen for the Wizard which contains a title, guidance, and the outline of categories used to navigate to the pages of the hierarchy</li> <li>Navigates the categories and sub-categories of the outline</li> <li>Views pages which contain guidance, one or more Forms, and Items already associated with the page</li> <li>Creates new Items from Forms</li> <li>Selects pre-existing Items from her repository which match the allowed Form types</li> <li>Edits associated Items</li> <li>Removes associated Items</li> </ul>
<b>Program Administrator / CIG Coordinator</b>	<ul style="list-style-type: none"> <li>Designs Hierarchy Wizards for Portfolio Owners to interact with               <ul style="list-style-type: none"> <li>Creates arrangement of categories and sub-categories</li> <li>Creates pages which may include one instance of Guidance, and one or more Forms                   <ul style="list-style-type: none"> <li>♣ Pages may have one required/optional Reflection</li> <li>♣ Pages may have one required/optional Evaluation Forms</li> <li>♣ Pages may allow Feedback</li> </ul> </li> <li>Designates Evaluators for Matrix, Rows, Columns, or individual Cells (Wizard Pages)</li> </ul> </li> </ul>
<b>Reviewer</b>	<ul style="list-style-type: none"> <li>Browses a specific Portfolio Owner's Matrix by selecting the Portfolio Owner by name</li> <li>Provides Feedback on a page (rich text area)</li> </ul>
<b>Evaluator</b>	<ul style="list-style-type: none"> <li>Evaluate Wizard Pages using the designated Evaluation Form for the page</li> </ul>

Table 9 - Role based features for Hierarchy Wizard

## Matrix Wizard

A Matrix Wizard is an organizational structure for Wizard Pages which is presented to the Portfolio Owner in a visual representation of rows and columns. The intersection of a row and column—a cell—is a Wizard Page which provides a point of interaction for the Portfolio Owner, Reviewers, and Evaluators. In general, the matrix provides a means of supporting longitudinal progression, with access to each Form determined by the completion of the Form that immediately precedes it. Navigation for a matrix may move horizontally across a row, vertically down a column, or independent of Form relationships within the Matrix. The specific workflow, as well as other attributes, is designed by a Program Administrator or CIG Coordinator.

User Role	Feature Descriptions
<b>Portfolio Owner</b>	<ul style="list-style-type: none"> <li>• Views the Matrix ‘home’ state which contains a title, description, rows, columns, and cells which illustrate associated work by icon, and status by color</li> <li>• Navigates to individual cells to:               <ul style="list-style-type: none"> <li>• View Guidance associated with a cell</li> <li>• Associate Items from Resources</li> <li>• Complete required/optional Reflections based on designated Reflection Form for a cell</li> <li>• Submit cell for Evaluation</li> <li>• View Feedback on cell made by reviewer</li> <li>• View Evaluation made by evaluator</li> <li>• Edit / Remove Items</li> </ul> </li> </ul>
<b>Program Administrator / CIG Coordinator</b>	<ul style="list-style-type: none"> <li>• Designs Matrix Wizards for Portfolio Owners to interact with               <ul style="list-style-type: none"> <li>• Creates arrangement of rows and columns</li> <li>• Creates pages which may include one instance of Guidance, and one or more Forms                   <ul style="list-style-type: none"> <li>♣ Pages may have one required/optional Reflection</li> <li>♣ Pages may have one required/optional Evaluation Forms</li> <li>♣ Pages may allow Feedback</li> </ul> </li> <li>• Designates Evaluators for Matrix, Rows, Columns, or individual Cells (Wizard Pages)</li> </ul> </li> </ul>
<b>Reviewer</b>	<ul style="list-style-type: none"> <li>• Browses a specific Portfolio Owner’s Matrix by selecting the Portfolio Owner by name</li> <li>• Reviews progress in each cell of the Matrix</li> <li>• Provides Feedback on a cell (rich text area)</li> </ul>
<b>Evaluator</b>	<ul style="list-style-type: none"> <li>• Evaluates submitted cells in the Matrix using the designated Evaluation Form for the cell</li> </ul>

**Table 10 - Role based features for Matrix Wizard**

## Portfolio Studio

The Portfolio Studio is where a Portfolio Owner designs and manages Portfolios—unique presentations of Items selected and designed by the Portfolio Owner for a particular audience. Portfolios can contain multiple pages, and each page may incorporate a design limited only by what can be rendered in a web browser. Portfolio Owners may use the Portfolio Designer which provides the flexibility to design multi-page Portfolios by choosing layout, style, and content for each page and choosing a navigational scheme to link the pages together. The Owner may choose to have OSP provide navigation or may design their own through including hyperlinks within the pages of the Portfolio. New layouts and styles may be developed by users with advanced HTML/CSS and basic XML skills, putting these activities within reach of Web and graphic design students. Alternatively, Portfolio Owners may use pre-defined templates that determine layout, style, and navigation but permit the Portfolio Owner to choose the content.

A Portfolio Owner may have any number of Portfolios which can be published to: other users or roles within the OSP system, Guests by email address, a public URL for unrestricted access, and removable media.

The Portfolio Studio may be accessed from a particular CIG, or from their Personal Workspace. If accessed from the owner's Personal Workspace, the Portfolio Studio is configurable to include Portfolios from any CIG which the Owner is a member.

User Role	Feature Descriptions
<b>Portfolio Owner</b>	<ul style="list-style-type: none"> <li>Creates new Portfolios               <ul style="list-style-type: none"> <li>Uses Portfolio Designer to select layouts, styles, arrange and link pages, select content, and publish the creative work</li> <li>Selects Portfolio Templates and available content to publish work using layout and style of pre-defined templates</li> <li>Selects audiences and methods of publishing (other users, roles, guests, public url, removable media)</li> </ul> </li> <li>Browses existing Portfolios to view/edit/delete</li> </ul>
<b>Program Administrator / CIG Coordinator</b>	<ul style="list-style-type: none"> <li>Imports Portfolio Templates for use within the OSP System</li> </ul>
<b>Guest</b>	<ul style="list-style-type: none"> <li>Views Portfolios which have been shared with the Guest</li> <li>Comments on Portfolios</li> </ul>
<b>System Administrator</b>	<ul style="list-style-type: none"> <li>Imports Layouts and Styles for use within the OSP System</li> <li>Imports Portfolio Templates for use within the OSP System</li> </ul>
<b>Instructional Technologist</b>	<ul style="list-style-type: none"> <li>Designs XHTML Layouts and CSS-based Styles</li> <li>Designs Portfolio Templates</li> </ul>

Table 11 - Role based features for Portfolio Studio

## Reports

Reports are all new in OSP 2.1. OSP reports are based on XML Report Definition files which are highly flexible, yet require technical expertise to create. These Report Definition files are imported into the OSP System to give non-technical users the ability to create Reports. An XML Report Definition file may specify parameters which are presented to Program Administrators, CIG Coordinators and other users with access to create Reports through the Report Designer. The user designing the report begins by naming the Report to be created, describing it, and selecting the Report Type (from a list of imported Report Definition files). Next the Report Designer prompts the user to select options and fill-in parameters to create the report. These parameters may be used to filter and/or select data. The Report Designer also prompts the user to select from an available list of output styles.

Once created, reports may be saved so they can be re-run at any time with the same set of options and parameters. In addition, the data can be exported to Microsoft Excel format, and Report Snapshots may be saved which preserve the actual report data at the time the Report Snapshot was created, and still allow style and layout to be chosen when the report is viewed.

User Role	Feature Descriptions
<b>Portfolio Owner</b>	<ul style="list-style-type: none"> <li>Views and exports applicable pre-designed Reports for which she has access</li> <li>Creates new reports using the Report Designer</li> </ul>
<b>Program Administrator / CIG Coordinator</b>	<ul style="list-style-type: none"> <li>Creates new reports using the Report Designer</li> <li>Views and exports applicable pre-designed Reports for which she has access</li> </ul>
<b>Reviewer</b>	<ul style="list-style-type: none"> <li>Views and exports applicable pre-designed Reports for which she has access</li> </ul>
<b>Evaluator</b>	<ul style="list-style-type: none"> <li>Views and exports applicable pre-designed Reports for which she has access</li> </ul>
<b>System Administrator</b>	<ul style="list-style-type: none"> <li>Imports Report Definitions and Styles</li> <li>Views and exports applicable pre-designed Reports for which she has access</li> </ul>
<b>Instructional Technologist</b>	<ul style="list-style-type: none"> <li>Designs XML Report Definitions</li> </ul>

**Table 12 - Role based features for Reports**

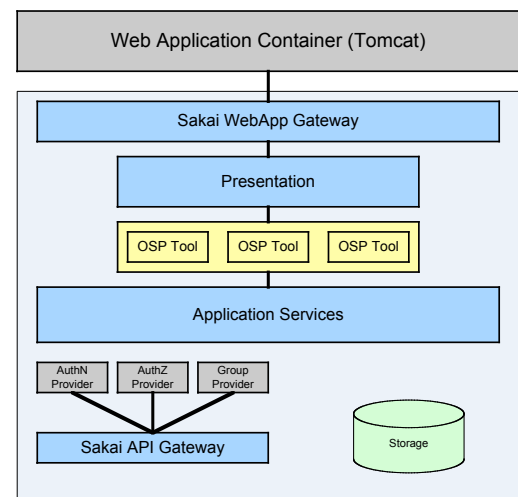
## Standards and portability

OSP is committed to fully supporting open standards and specifications so that portfolios may be portable between OSP and other systems and between institutions that use OSP. A member of the OSP leadership co-lead the development of the international IMS ePortfolio Specification, which was designed to address this need for portability. OSP will implement this specification. However, for portability to be meaningful, it must be accompanied by agreements about basic data elements and vocabulary terms. Because OSP is already in widespread use, establishing consensus on these matters is considerably more challenging than simply building the mechanisms to import and export data in a given format. OSP 2.1 will include a basic set of Forms that together constitute the core of a shared data model institutions across the OSP community may use. Each of these forms will be mapped to the appropriate data structure within IMS ePortfolio. Once these common elements have been proven in practice, an import/export tool will be added to the software. However, this tool will not be included in Version 2.1. Such a tool will make it possible to also support import and export of portfolio information in other standard formats, such as HR-XML.

## Technical details

OSP is an enterprise Java (J2EE) application that runs on a server and is accessed by users through a Web browser such as Mozilla Firefox. OSP is built on the Sakai framework which provides institutions with a common set of software services for teaching, learning, research, and collaborative project work.

If you plan on joining the development effort as a software engineer, you'll want to get familiar with the Sakai architecture, technologies such as XML, XSL, JSP, and JSF and frameworks like Spring and Hibernate. Getting involved in development does not necessarily require Java skills. The OSP development effort requires individuals with XML, XSLT, XHTML, SQL and testing skills as well.



## Licensing

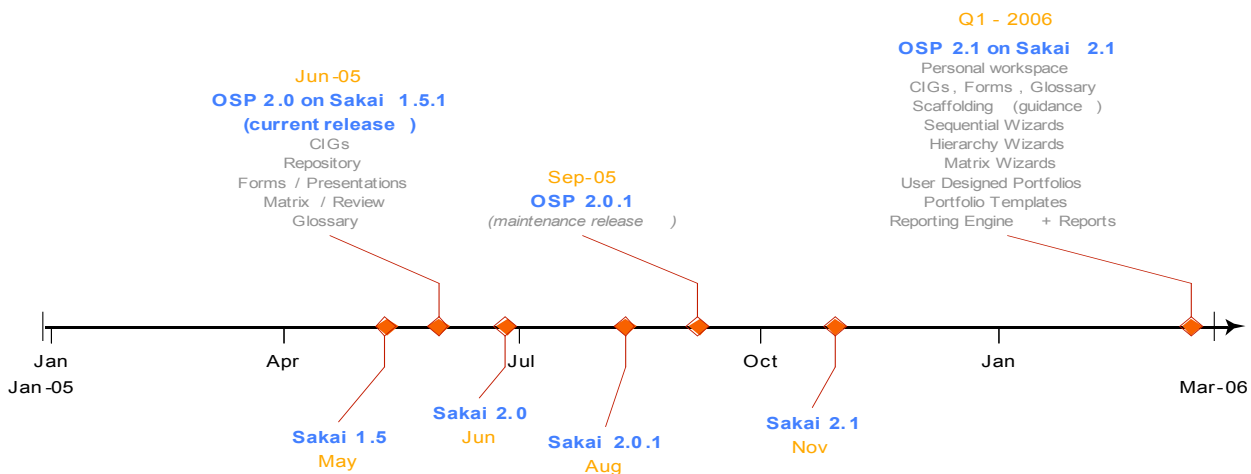
The Open Source Portfolio (OSP) is [OSI Certified open source software](#). It is licensed for use pursuant to the Educational Community License v. 1.0 (ECL). The ECL is a commercial-friendly BSD-style license that grants anyone the right to use the software for any purpose, modify the software, and share the software and any derivatives with others. You can read the [ECL on the OSI website](#).

OSP now follows licensing and contribution practices modeled after the Apache Software Foundation (ASF). The ASF has a long, successful history managing licensing and intellectual property for the good of the Apache community. Recently the Sakai Foundation was formed as a not-for-profit organization. OSP, by affiliation with the Sakai Foundation, is now able to manage intellectual property and software licenses with the same rigor as the Apache Software Foundation.

The ECL and our licensing and contribution practices help us achieve our goal of providing reliable and long-lived software through collaborative open source software development. In all cases, contributors retain full rights to use their original contributions for any other purpose outside of OSP or Sakai while providing the Sakai Foundation and its projects the right to distribute and build upon their work.

## Roadmap – OSP’s future

OSP was first released to the community as the Open Source Portfolio in July 2003. Version 1.5 followed in June of 2004 when work had already begun on version 2.0, which was first released in June of 2005 as depicted on the timeline below.



OSP 2.1 is in development as this document is being written. The scope of OSP 2.1 is a subset of a substantial body of work that articulates the most advanced portfolio system thinking anywhere in the world. Portfolio thought leaders organized as an OSP workgroup called “FX” have worked tirelessly over the past 18 months to craft the OSP software vision.

That entire vision has yet to be mapped onto a calendar to produce a detailed roadmap but the following is a small preview of what’s to come:

**IMS ePortfolio specification based portability** – The 2.1 release is the first step toward portability with data mapped to the specification. In subsequent releases, OSP will implement import and export functions to seamlessly transfer portfolios from one system to another.

**Increased granularity in re-usable components** – In OSP 2.1, Forms are the smallest re-usable building blocks of information. A Form is a grouping of fields presented to a Portfolio Owner on a single page. When completed by a Portfolio Owner, a Form becomes an Item stored in their Resources area. In future releases of OSP, a more discrete component called an Element will be introduced to improve re-use and portability. Elements will be re-usable groups of fields that can be arranged on a Form. This increase in granularity will permit greater re-use of information within multiple Forms. Portability of information through the IMS specification will be simplified as a more direct relationship to IMS components (through Elements) is possible.



**Increased use of Builder tools** – The ability to customize OSP will have two contexts – one being the interface experience of using the application itself, the other being the ability to use Builders to create Items for use within the system including Portfolios, Wizards, Forms, Reports, Layouts and Styles.

**Collection** – The Collection will provide a means for a Portfolio Owner to organize and locate items within the portfolio application based upon where items are used, the type of items collected, and when a set of items was added to a collection. This functionality, commonly known as “free-tagging,” allows the Portfolio Owner to work with items without having to keep track of where an item is stored within a file structure, or make decisions regarding the ‘best’ place to store an item that may be used for many purposes. Through the use open standards often employed in social software, such as RSS, the Collection will enable OSP to interoperate with blogging, social bookmarking, and social networking tools.

**Reporting and Analysis** – The Reporting environment will be extended to include access to all information collected through elements, as well as the context through which the information is used. When using a Report Builder, the information presented for selection for reporting is presented based upon the information gathered by the CIG in which the report is built. Individual fields will also provide, when available, valid field values when the field includes a valid value set, as well as text search when the field includes an open value set.

**Searching, Sorting, and Filtering** – The same meta-data awareness that supports a robust report builder experience is also implemented as part of a search engine, coupled with the Collection to allow all Portfolio Owners to easily find items, group items for easy reference, and to restrict selection of items. This feature is provided to assist a Portfolio Owner in the selection of items for use within a portfolio, or to support completion of a form within a wizard presented for a CIG.

## Get involved

First, identify yourself to the rest of the community. Create an account at [bugs.sakaiproject.org/confluence](http://bugs.sakaiproject.org/confluence) that you will use to access online tools. It’s easy, fast, and free to create one account to become a recognized user of all the online tools the OSP community uses to collaborate.

### Get to know the software and how others are using it

The OSP website ([www.osportfolio.org](http://www.osportfolio.org)) is the best resource to learn more about the OSP software. If you’re just getting started with OSP, follow the three steps below to get familiar with the software and how others are using it:



GUIDED TOUR



ONLINE DEMO



CASE STUDIES

[www.osportfolio.org/tour](http://www.osportfolio.org/tour)

[demo.osportfolio.com](http://demo.osportfolio.com)

[kml.carnegiefoundation.org/OSP/](http://kml.carnegiefoundation.org/OSP/)

In addition, visit [bugs.sakaiproject.org/confluence](http://bugs.sakaiproject.org/confluence) for a growing library of technical and user documentation.

**Create bookmarks for easy access:**

- OSP Website: [www.osportfolio.org](http://www.osportfolio.org)
- OSP Discussion Group: [collab.sakaiproject.org](http://collab.sakaiproject.org)
- Bug tracking database: [bugs.sakaiproject.org](http://bugs.sakaiproject.org)
- Documentation: [bugs.sakaiproject.org/confluence](http://bugs.sakaiproject.org/confluence)

Next log into the OSP Discussion Group site on “Collab” and join the appropriate groups. Joining groups gives you access to the community and to other people in those groups. Much of the discussion among the OSP community takes place in the mailing lists of the various groups. You might also benefit from joining one or more Sakai groups also available at [collab.sakaiproject.org](http://collab.sakaiproject.org).

There are many ways to contribute to OSP. It’s a mistake to think that you have to be technical or have a lot of institutional resources to add value to the community. We value all sorts of contributions including:

- Download and use the software
- Ask questions on the lists
- Log bugs and ideas for improvements in the bug tracking database (Jira)
- Add to and improve the documentation of the OSP software
- Help plan or organize a conference or meeting
- Present your experience at a conference session
- Contribute a component like a Wizard, Form, Portfolio Template, or language translation
- Participate in testing the OSP during the iterative development of a release
- Contribute a developer, designer, technical writer, or another role to the development of a new release

These are just a few of the ways members regularly contribute back to the OSP community. Once you get involved you’ll see lots of other ways you can contribute to the longevity and success of OSP. You may even have a new idea for a contribution. If everyone in contributes, the OSP community is strengthened and sustained.

## Get support

There are three complementary sources for getting help with the OSP: self help resources, community support, and commercial support. Each is described further in the following sections.

## Self-help resources

The OSP community maintains online resources such as documentation in the Sakai Confluence site ([bugs.sakaiproject.org/confluence](http://bugs.sakaiproject.org/confluence)), archives of the mailing list discussions ([collab.sakaiproject.org](http://collab.sakaiproject.org)), and Issue documentation and resolutions in our defect tracking tool ([bugs.sakaiproject.org](http://bugs.sakaiproject.org)). Each of these resources can be a valuable source for information and self-help.

## Community Support

### Work/Discussion Groups

The OSP Discussion Groups provide online tools for collaboration. One of the most frequently used tools is the email lists which anyone can join. These are accessible through your account at [collab.sakaiproject.org](http://collab.sakaiproject.org). Through these group mailing lists, members of the OSP community pose questions, post comments, and receive answers every day.

Remember, there is no *they*. Sometimes you'll be the one asking questions, other times you'll be answering them. Don't be shy. Although many OSP community members start out posing questions on the lists, it doesn't take long begin adding comments and answering others' questions.

### Conferences

Time and again, members of the OSP and Sakai community comment on how much support they receive from attending the bi-annual conferences. Usually one summer and one winter, conferences are attended by developers, administrators, and faculty—representatives from all sectors of higher education from around the world. Interactive sessions, demos, and keynotes at these conferences address critical issues from implementation strategies, to pedagogical methodologies, to coding practices, architecture, and APIs. But as with all conferences, it is the hallway contacts and dinner conversations that provide the most meaningful support connections.

## Commercial Support

Commercial support of open source projects like OSP is an essential complement to self-help resources, and community support. Many institutions require the services, accountability, and support typically available for commercial software. Commercial support providers like The rSmart Group offer technical support, assistance with installation, configuration, training, integration with other systems, customization, and application hosting. With a company like rSmart you get the best of both worlds – the freedom and control over your destiny that comes with open source, and the assurance of reliable professional organization at your disposal.

## Appendix A - Glossary

**Associate** – The act of creating a relationship between two items that exhibit a dependency that is relevant in the portfolio application.

**Association** A semantically-valuable relationship between two items. Items in OSP may be associated with cells in a Matrix, Forms, and other constructs.

**Audience** – An individual or group of individuals that have been identified as the target of a published Portfolio.

**Cell** - The intersection of two dimensions within a matrix.

**CIG** – Common Interest Group. Most commonly, a Worksite where members interact and collaborate. The CIG serves as the container and provides context for their interaction. A CIG may also exist solely for reporting purposes or where members do not interact and are unaware of one another

**CIG Coordinator** - A CIG coordinator is responsible for managing the storage of items provided for use by a CIG workgroup. The CIG coordinator is also responsible for managing template items, such as Forms and Wizards intended to gather information from CIG participants. The CIG coordinator may need to assist in managing participant's items that have been locked due to activities completed within the CIG.

**Collect** – The act of storing an Item within the Portfolio Owner's Resources.

**Collection** – *See Resources.*

**Completed Form** - A completed form is an instance in which one or more fields have been populated. This is generally referred to as an Item.

**Component** – Items users may add to OSP that shape how a portfolio is composed, including styles, layouts, forms, wizards, glossary, portfolio templates, language resource bundles. Components are portable between instances of OSP.

**Coordinator** – See CIG Coordinator

**Element** - An element is a future-looking term that describes the smallest re-usable collection of data in the OSP. In the future, OSP Forms will consist of re-usable Elements, which consist of one or more Fields. In OSP 1.5, an Element was equivalent to what OSP 2.x defines as a Form.

**Evaluator** – See Portfolio Evaluator

**Feedback** – Rich text written about something in a portfolio, usually by someone other than the Portfolio Owner.

**Field** - A field is an area within a Form in which a user is prompted to enter information. A Form consists of one or more Fields.

**Form** – An OSP Form is a group of one or more Fields which are presented to the Portfolio Owner to be filled in with data. Once completed, a Form is stored in the Portfolio Owner's Resources and referred to as an Item.

**Form Builder** – A tool within the portfolio application that supports the creation of Forms.

**Guest** – See Portfolio Guest

**Guidance** – Assistance for a portfolio owner through descriptive instructions, examples of expected results and rationale for inclusion of an item or completion of a task.

**Hierarchy Wizard** – A Hierarchy Wizard is an organizational structure for Wizard Pages which is presented to the Portfolio Owner like an outline.

**IMS ePortfolio Specification** – A technical specification that defines a data structure for portfolios, enabling portability. This term refers to the most recent version of the IMS ePortfolio Specification available here: <http://www.imsglobal.org/ep>.

**Item** – A file, completed Form, or URL stored in Resources.

**Layout** – A layout defines the position of content presented within an individual page of a Portfolio. It may include one or more item “placeholders” into which certain types of Items may be placed or rich text may be entered directly. A layout is similar to a slide design in PowerPoint.

**Matrix** – A type of Wizard consisting of a two-dimensional grid. Matrices are typically used to facilitate Association, Reflection, Feedback, and Evaluation of Items with Criteria along a longitudinal progression.

**OSP** – Open Source Portfolio.

**Owner** – See Portfolio Owner

**Personal workspace** – The Sakai CLE is primarily organized into sites. Sites are a collection of users, roles, tools, and generally a space for interaction. Sites can be of different types such as course sites, project sites, and portfolio sites which OSP calls Common Interest Groups (CIGs). Each user of Sakai and the OSP has a Personal Workspace called “My Workspace” which provides access to Portfolios, Reports, Resources, etc. which span individual sites or CIGs. When a user first logs in, they are presented with their “My Workspace” home.

**Portability** – The ability to import and export data between systems.

**Portfolio** – An organized collection of one or more Portfolio Pages which are intended to be published and shared with others.

**Portfolio Evaluator** – An evaluator is responsible for evaluation of items which have been submitted by the Portfolio Owner for evaluation. In OSP 2.1, Wizard Pages can be submitted for evaluation. The Portfolio Evaluator is explicitly assigned for evaluation responsibilities, and he/she uses a customizable Evaluation Form to conduct the Evaluation.

**Portfolio Guest** – A portfolio guest is a person who does not have a normal user account in the system, but has been granted access to a Portfolio which the owner has published. The Portfolio Owner may publish a Portfolio for any person using an email address. If the person does not already have an account in the system, the OSP will create a guest account and provide instructions on how to access the published Portfolio. Guests will often provide the Portfolio Owner with rich text comments on the Portfolio.

**Portfolio Owner** – The Portfolio Owner is the center of the OSP universe. Each user of OSP may be a Portfolio Owner who creates and maintains a collection of Items; uses Wizards to create new information, reflect, submit work for evaluation, and receive feedback; and designs and publishes Portfolios to showcase and share their work.

**Portfolio Reviewer** – A portfolio reviewer is a person who provides commentary and guidance to a Portfolio Owner. The guidance provided by the portfolio reviewer is generally considered informal or formative in nature.

**Program Administrator** - A program is a person who provides guidance and resources to CIG coordinators and evaluators. The program administrator creates and maintains Forms, Reports, A Global Glossary, and Wizards.

**Publish** – The process of sharing a Portfolio available with one or more audiences.

**Rationale** – An explanation of why a given task is to be performed, the logical basis of a procedure.

**Reflection** – An Item created by a Portfolio Owner by completing a Reflection Form. A Reflection Form is used to capture the Portfolio Owner’s reflective thinking about a particular Item, or the relationships between Items.

**Report** – A set of results from a query arranged with a particular layout and style.

**Repository** – See Resources.

**Resources** – A tool for storage of any Item authored by a portfolio owner, as well as a means of organizing the items into a hierarchical system of folders and sub-folders. Items stored in Resources may be associated with any number of other constructs such as Forms, Wizards, and Portfolios.

**Reviewer** – See Portfolio Reviewer.

**Sakai** – “A community source software development effort to design, build and deploy a new Collaboration and Learning Environment (CLE) for higher education.” From <http://www.sakaiproject.org/>.

**Style** - Specifies colors, typography, and other characteristics for use in visually rendering items in Forms, Wizards, and Portfolios.

**System Administrator** - A system administrator is responsible for the maintenance and integrity of the OSP implementation at the institution. This includes assisting each of the portfolio roles when determining appropriate application settings for general use of the portfolio application, performing required backup and maintenance of the portfolio components.

**Tool** - A unit of functionality that can be added to a worksite by its owner.

**View** – See Portfolio.

**Wizard** – An abstract organizational construct that gives a Program Administrator or CIG Coordinator the ability to design a framework for the creation and organization of information.

**Wizard Page** – A Wizard Page is a single screen presented to a user, generally the Portfolio Owner, which may contain Guidance, Forms, Feedback, Evaluation, and Items which the Portfolio Owner has associated (Files and completed Forms).



## Appendix B – OSP Forms and XML Schemas

The following documentation is a work in progress. The most up-to-date information is located at: [bugs.sakaiproject.org/confluence](http://bugs.sakaiproject.org/confluence) and will become part of the OSP 2.1 documentation when development of the 2.1 release is complete.

**Text input element:** A form designer may want to elicit from users short strings representing people's names, document titles, telephone numbers, email addresses, etc. The preferred input widget is a simple input element of type "text". It generally accepts various printable characters and spaces, but tabs, newlines and other "control" characters are not generally accepted. The preferred way to generate a text input element in a form is to specify an element with type of "xs:normalizedString". This can be specialized by adding restrictions, including "xs:length", "xs:maxLength", "xs:minLength" and "xs:pattern". (See additional notes under "textarea element" and "select element").

**Textarea element:** A form designer may want to elicit longer strings representing paragraph-length descriptions, comments, notes, etc. The preferred input widget might be a simple textarea element (see also "rich-text editor"). Textareas accept ascii character inputs including newlines and tabs. The preferred way to generate a textarea in a form is to specify an element with type of "xs:string", which by default will be rendered in the form as a small textarea. The "xs:string" element can be specialized by adding restrictions, including "xs:enumeration" (see "select element"), "xs:length", "xs:maxLength", "xs:minLength" and "xs:pattern". If "xs:length" or "xs:maxLength" is specified with a value of less than 100, the element will be rendered as a text input area. If no length/maxLength restriction is included or if a length/maxLength restriction of 100 or more is given, it will be rendered as a textarea. Note that the "xs:string" type is also used with "xs:enumeration" restrictions to specify select elements. (See additional notes under "text input element" and "select element").

**Rich-text editor:** A form designer may want to require/enable users to enter longer strings with mark-up for special fonts and font attributes. The preferred input widget is the WYSIWYG editor. The way to specify use of the WYSIWYG editor for a field in a form is to specify an element in the XSD file as an "ospi:richText".

**File-picker widget:** A form designer may want to specify a form element that allows a user to link to a sakai resource such as a file or URL. The preferred widget for making such a link is the sakai attachments helper, which allows the user to link to an existing resource or create a new one (by uploading a file, for example). The way to specify that a particular field's value will be supplied with an attachment is to use an element of type "xs:anyURI".

**Select element:** A form designer may want to elicit a selection from a small set of known possibilities. The preferred input widget for such selections in an HTML form is a select element. The way to generate a select element in a form is to specify an element in the XSD file as an "xs:simpleType" based on "xs:string" with a set of possible values included as "xs:enumeration" restrictions. Any length or pattern attributes will be ignored for select elements.

**Date widget:** A form designer may want to elicit input of a date. The preferred input widget for a date in an HTML form consists of three select elements (for month, day and year). The preferred way to specify use of a date widget in the schema is as an element of type "xs:date". The accepted values can be

restricted using "xs:maxExclusive", "xs:maxInclusive", "xs:minExclusive" or "xs:minInclusive". Failure to specify minimum and maximum values may render a date picker that is restricted incorrectly.

(QUESTION: What should be the default range of values if min and max are not specified?) Values for minimum and maximum will be used to validate user inputs. The date widget may accept values outside this range (but between January 1 of the minimum year and December 31 of the maximum year), but the validator will reject those.

**Numeric input element:** A form designer may want to elicit input of numbers – either integer values (so-called "whole numbers") or decimal values (possibly containing fractional parts). Accepted values may be restricted by specifying "xs:totalDigits", "xs:maxExclusive", "xs:maxInclusive", "xs:minExclusive" or "xs:minInclusive".

**Yes/no checkbox element:** A form designer may want to elicit a boolean value (true/yes or false/no) from a user in response to a prompt in a form. The preferred input element is a checkbox, which indicates a value of true if checked and false otherwise. A checkbox will be used for form elements having a type of "xs:boolean".

All fields can be required. This is done with a value of one for minOccurs. If minOccurs is more than one, then the user needs to be able to add multiple values. Required fields will be marked with a RED star next to the field (the usual sakai way), and the validation process will report an error if the value is not present.