The Repository OSID and SRW

Leveraging standards to provide broad markets for content

Summary

The Open Knowledge Initiative (O.K.I.™) Repository OSID includes interfaces for integration among applications (consumers) and repository content (providers). Applications use the OSID to gain access to content in a manner that hides the technical detail by which that content is provided. This allows the application to mange integration simply with a wide range of content providers without including the complexity inherent in supporting heterogeneous means of communication and data exchange. The application is also insulated from technology changes made by providers which leads to a longer useful lifetime for an application and thereby a greater return on investment.

MIT developed a Repository OSID that provides access to content from sites supporting the OCLC Search and Retrieve Web (SRW) standard. This has allowed application such as Sakai, Tufts University's Visual Understanding Environment (VUE), HarvestRoad Hive Explorer, Giunti Interactive Labs Learn eXact Package, and the Mac Learning Environment's SearchParty to access SRW-served content for no additional development effort beyond their initial adoption of O.K.I. SRW-served content is available to a wider range of tools with no effort required of the provider.

Applications gain access to more content; content providers gain a wider market, all at a low marginal cost. This case study will illustrate:

- A Repository OSID implementation covering a web service.
- Content available to applications after application development ended.
- Applications using content providers they were unaware of during application development.
- Content providers gaining additional market for no additional effort.
- The same OSID implementation suitable for most SRW-supporting sites – only configuration required.
- The cumulative benefit from using Standards.

Background

The Open Knowledge Initiative (O.K.I.™) is a MIT-led, community effort to improve interoperability among applications and the enterprise system services on which they depend. The initial focus has been on higher education and eLearning, but the initiative's service-oriented architecture is applicable outside these domains. O.K.I. provides Open Services Interface Definitions (OSIDs): contracts between service consumers and providers. The OSIDs are well-defined integration boundaries that leave flexibility in the hands of developers. The OSIDs are neutral with regard to programming language[1] and implementation detail. There are OSIDs for common services such as authentication, authorization, hierarchy, scheduling, and workflow and eLearning services such as repository, assessment, grading, and course management. The OSID currently witnessing the greatest level of interest and adoption is the Repository OSID. Repository defines objects such as a Repository Manager, Repositories, Assets, and their metadata as well as methods for managing object lifecycle, data maintenance, and searching.

Search and Retrieve Web (SRW) grew out of collaboration among developers in the library community who were trying to leverage Z39.50 experience into the web world. SRW specifies using the Common Query Language (CQL) Repository content providers use diverse implementation approaches. There is local and remote content, content in databases accessible through a driver, web services, proprietary communications, etc. There are also numerous data formats, authentication schemes, and other details. In addition, every time a content provider makes a technology change, applications that are bound to that means of integration must change as well. All this makes integration hard to deliver and expensive and fragile to maintain.

For tool developers, O.K.I. offered a standards-based repository content integration strategy. Supporting a technology-neutral OSID saved the tool developer from the complexity of understanding and tracking many disparate technologies. By implementing OSID consumer support, each tool developer gained immediate access to any content for which OSID provider implementation was available, subject to the content vendor's licensing requirements.

SRW-serving sites have been available for some time and there is an established community of users and tools for this content. None of the tool developers that were early O.K.I. adopters yet had made the investment in supporting access to content via SRW. Users of these tools wanted access to that content. MIT stepped in to offer a solution.

MIT undertook development of an OSID Repository (Java) provider implementation that understands SRW. Queries passed to the implementation via the methods in the OSID are converted into the appropriate SRW calls. Results from the SRW service are presented in the asset and metadata constructs appropriate for the Repository OSID. Immediately, tools could access the content from SRW in the same manner they would *any* OSID Repository provider. The fact that SRW was being used by the provider implementation was unknown to the tool. The scope of content now accessible was broadened further to include DSpace sites that use the SRW / DSpace support. Tools that had

[1] The programming language-neutral form of an OSID is called an XOSID. There are bindings of the XOSIDs for Java, Objective-C, and PHP. Other languages bindings are in development.

no knowledge of SRW when they were written were now able to work with that content – increasing the return on the original investment in the tool. Since SRW is a standard and regular approach, the Repository OSID implementation for SRW can use a straightforward configuration mechanism to interact with one or more SRW sites.

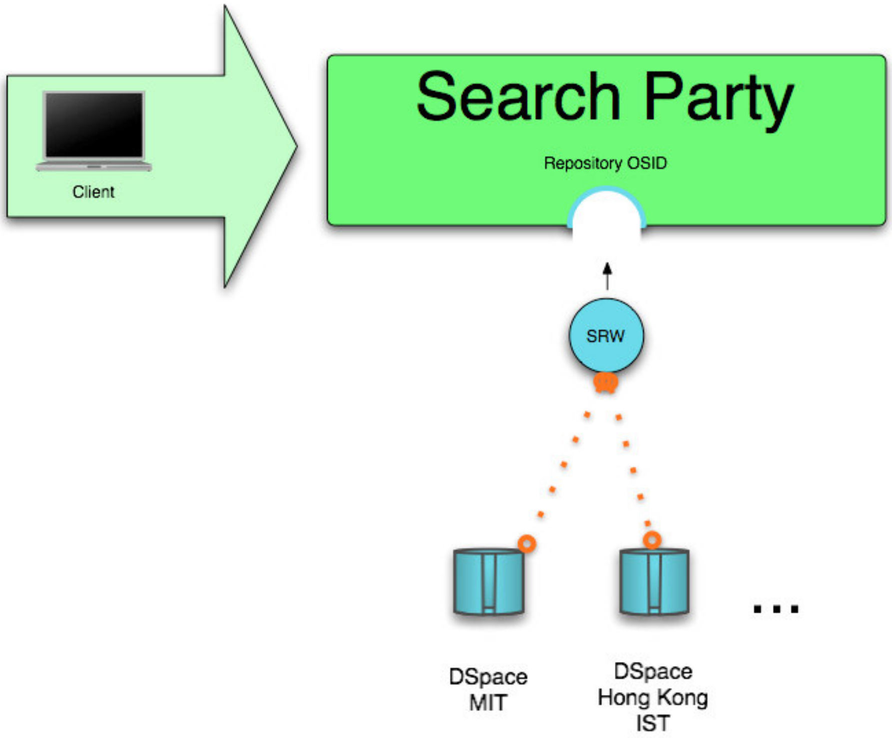

Figure 1: Mac OS X Search Party Application Accessing DSpace Content via OSID Plug-in atop SRW

Here we see the cumulative benefits that accrue from using standards. As the number and variety of content made available through SRW sites grows, so does the value of tools that work with SRW.

Similarly, OSIDs offer a "network" benefit that grows as adoption grows. OSIDs offer a further advantage because they are implementation-neutral technology. That technology can be SRW or something else and the consuming tool is no wiser and, more importantly, needs no additional SRW-specific support code.

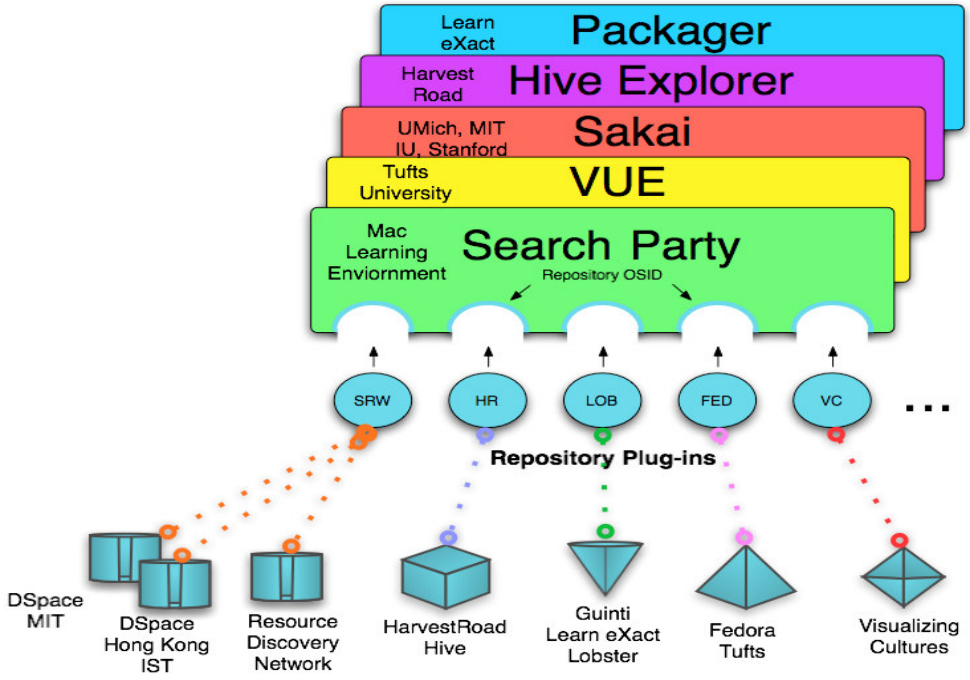

Figure 2: Multiple Applications Accessing Content via OSID Plug-ins

Special thanks to Ralph LeVan with OCLC for technical assistance with SRW.

<u>Websites</u>

O.K.I.	http:// www.okiproject.org
CQL	http://www.loc.gov/z3950/agency/zing/cql
DSpace	http://www.dspace.org
Giunti Interactive Labs	http://www.giuntilabs.com
HarvestRoad	http://www.harvestroad.com
Mac Learning Environments	http://www.maclearningenvironments.org
OCLC	http://www.oclc.org
Sakai	http://www.sakaiproject.org
SRW	http://www.oclc.org/research/software/srw http://www.loc.gov/z3950/agency/zing/srw
VUE	http://vue.tccs.tufts.edu