e-Literate
What Michael Feldstein is Learning About Online Learning..Online

19 Nov 2005

» LMOS Services and Service Brokers, Part II

In a (relatively) recent post, I started to outline how a service broker mechanism could greatly increase the pace of innovation in LMS design. The basic idea was that individual applications in the system could provide services that other applications could automagically pick up on, without requiring developers to wire up integration individually every time. In this post, I'm going to illustrate how such a beast might work, using the example of adding a blog to a Learning Management Operating System.

Let's imagine that we're taking a relatively generic weblog application and adding it to the LMS environment. Each student would get her own personal blog--not within each course, but one personal weblog per student that stays with that student throughout her college career. Naturally, the weblog would have an RSS feed.

Now, it turns out that RSS feeds carry quite a bit of information that could be useful in a learning context. Here's a list of just a small subset of the information available from my RSS feed, for example:

- The URL of my blog's home page
- The ID of the blog's author (in my case, it gives my email address as my ID)
- The software application that generated the posts
- The URL, title, description, contents, time stamp and category labels for each post

Notice two things about this data. First, it's very generic and could be useful information about just about any content online. Second, the post-specific information (in the last bullet point) is pretty much exactly what you need to know about any assignment that a student submits for a class.
In order to start making use of this data in the context of an LMOS, the blog developer need only make a few relatively minor technical enhancements in order to plug into the service broker:

1. Write an adapter that enables the RSS feed to talk to the broker. (Since RSS is a very common format, chances are good that such an adapter would already exist.)
2. Tie the blog into the single sign-on mechanism, so that the LMOS knows that the person that owns a particular blog is also, say, a student in the Psychology 101 class.

3. Extend the blog to be able to subscribe to category labels that are related to the groups to which the student belongs (e.g., the Psych 101 student should see a "Psych 101" category tag show up in her post category list).

Now we're ready for some service broker automagic. Let's say the student decides to write a blog post on a topic related to her Psych 101 class. As she writes her post, she looks over to the category list. Because the system knows that she is a registered student in Psych 101, it automatically adds "Psych 101" to her category list. She selects the appropriate category heading(s), writes her post, and publishes. The service broker, seeing that the content is labeled "Psych 101", announces to all the applications within the Psych 101 course environment that it has some student-created content. "Can any of you applications do anything with this student-created content?" it asks.
It gets the following responses:

- The class RSS aggregator responds, "Yeah, I can do something with it." It takes the student's post and publishes it along with those of the other students in the class.
- The course activity tracker says, "Me too. Gimme some of that." It notes the student ID, the time stamp, and the title and URL of the post. Using this information it adds an entry for the student's class activity on that particular date.
- The grade gook says, "I can also use that." Noting that the content is generated by the weblog application, it pulls the post text and URL into the student's row in the gradebook under the "weblog entries" heading. The instructor can now assign a grade and comment to it.

Notice that the weblog developer didn't have to write separate integration code for course activity tracker and grade book. The service broker was able to integrate the new application on-the-fly because the blog publishes the basic required knowledge in a standard format. All the blog developer had to do was write a connector that picks up the categories from the system and works with its single sign-on mechanism. The broker does the rest. It would be the same for any other application, too. You could, for example, use more or less the same mechanism to integrate your discussion board with the grade book and the course activity tracker.

But wait. There's more.

Suppose we make one more minor enhancement. Suppose that individual applications *within* the course environment could publish categories to share with each other. Suppose, for example, that the grade book could publish a category corresponding to a particular assignment. Our student could select that particular assignment category for her post and the instructor would automagically have it show up in the appropriate grade book column, with the appropriate point scale and weighting, and so on. Let's imagine, too, that you could set your discussion board to generate a forum topic for particular category (such as the assignment heading in the grade book) and generate a new thread for each post that comes in labeled with that category. Students could continue to post to their personal blogs that travel with them beyond the class, but the instructor could also create class-internal discussions based on those posts. This is all done using fairly generic mechanisms, so developers creating new applications won't need to do anything special to integrate their new wiki, or simulation, or flux capacitor, or whatever with individual applications already in the course environment.

But wait. There's ***still*** more.

Suppose that, in addition to having students publish information **into** the course, the service broker also let the course publish information **out** to the student's personal data store (read "portfolio"). Imagine that for every content item that the student creates and owns in her personal area--blog posts, assignment drafts in her online file storage, etc.--there is also a data store to which courses could publish metadata. For example, the grade book, having recorded a grade and a comment about the student's blog post, could push that information (along with the post's URL as an identifier) back out to the student's data store. Now the student has her professor's grade and comment (in read-only format, of course), traveling with her long after the system administrator closed an archived the Psych 101 course. She can publish that information to her public e-portfolio, or not, as she pleases.

So a service broker can greatly increase the pace of innovation in LMS design by greatly increasing the ease with which new applications can be deeply integrated with the rest of the learning environment. If you add in a portal for unified display capabilities, you have your Learning Management Operating System.