Sakai Use Cases Temporary

A Learning Management System (or Collaborative Learning Environment, or Framework for and Suite of Web Applications Used by Teachers, Students, and Researchers in an Institution of Higher Education) typically needs to deal with far more complex authorization requirements than a hosted web application, a corporate site, or a government site. In higher education, a single individual may play an instructor in a classroom, a student in a seminar, a leader in a discussion group, a grader in a lab section, and an editor on a research project. Some of these roles and contexts (but not all) are typically determined outside the LMS. These context-dependent roles imply different application roles and permissions. The set of possible roles and their implied permissions vary from institution to institution, and even within an institution. The applications (and their own ideas of user functions) are developed independently and deployed over time.

What follows is an initial culling from known requirements for groups, roles, and privileges. Group and role management use cases take the most space, but I also mention less visible authorization integration requirements that have caused QA and maintenance pain over the years. The names in parentheses are experts whose accounts I've drawn from and who I hope will correct any mistakes in my summary. I hope the terminology I use will be familiar to Sakai 2 veterans, but bear in mind that new standards and solutions will likely bring new vocabulary (e.g., "function" instead of "role" or "permission").

A. Mostly group management

1. Basic community features
2. Memberships from external systems
3. Managing subgroups of external members
4. Federating membership sources

B. Mostly roles and privileges

5. Integrating applications and services
6. Supporting local installations and institutions
7. Outside the ivory firewalls

1. Basic community features

To make more room for the Hard Problems, I won't bother to detail the many use cases which fit now well-established collaborative paradigms. Here are a few sample activities:

- Create a personal workspace with a digital repository, some public documents, and an blog on which any registered user can comment.
- Create a publicly accessible interest group. Joining the group gives shared access to a workspace including a discussion board, notifications, announcements, a calendar, and a wiki. The group can be dropped at any time.
- Create a by-invitation-only research project site including archived messages, chat room events, and video clips.
- Form personal contact lists from known users.
- Send a message to people who've indicated on their calendar that they're attending a certain event.
- See which public groups a person belongs to.

2. Memberships from external systems

Hierarchical memberships and contextual roles derived from a course management interface

American universities typically derive information about classes, enrollment statuses, instructor assignments, and so forth from an SIS department. Other externally managed "enterprise" data might include course descriptions, department organizations, cohort memberships, tutorial groups, and lab or discussion sections and leaders. Currently in Sakai, the Course Management API is the most generic interface to such externally managed data. The upcoming IMS LIS standard targets the same functionality, and should find wider use among clients (e.g., Moodle and Blackboard as well as Sakai) and implementations (e.g., out-of-the-box integrations from PeopleSoft and through LDAP attributes).

The canonical LMS/CLE use case is that the instructors, students, and other people officially related to a course want a multi-purpose online workspace that automatically includes all the course members that SIS knows about and maps official roles to online application permissions in a consistent and natural way. The workspace may also include (or be restricted to) officially maintained subgroups of the course (such as lecture sections, scheduled labs, or discussion groups). It should be possible to address the subgroups separately and to give them special access to particular areas or resources. (Oliver Heyer)

Memberships and roles derived from Shibboleth attributes

On the discretion of the authenticating system, Shibboleth can pass attributes about a logged-in user along to the client system. The client system can then translate these attributes into a local group membership or role. As with LDAP queries, to maximize usefulness, the attributes will probably be as standardized as possible (e.g., based on EduPerson).

For example, QMShibb can be configured to map a Shibboleth-provided list of groups to group membership in an online assessment system. For another example, a Drupalinstallation can map a Shibboleth attribute to a list of Drupal roles. (E.g., "If the attribute HTTP_SHIB_EP_AFFILIATION matches 'student@mit.edu', give the user the roles 'authenticated user, student user'.")

In this case, an external source of memberships can be specified without the membership list being readable. For example, someone at the University of Leeds might set up a website which will automatically be accessible by anyone certified by Shibboleth as a member of the "bioinformatics" group at the University of Manchester. But that doesn't give anyone at Leeds a way to view the full set of Manchester students who might fall into that category. (Andrew G. Booth, Paul B. Hill, Steve Swinsburg)

Hierarchical memberships derived from LDAP queries

Oxford University uses LDAP as a central source of membership lists, such as "all first year students studing Medieval English" and "all staff in Exeter College" or even the intersection between the queries. The source groups are pure membership lists (with no intrinsic role mappings, for example). These query-based groups are used as sources of installation-wide group memberships or workspace memberships. Additional members can be added to these linked but internally-managed groups, and named sub-groups can draw from them. A person should no longer appear in internally-managed groups or sub-groups when the person disappears from their source LDAP-query.

There is a natural hierarchy to many of these lists, such that each college within the university might have a top-level workspace (accessible to all members of the college) and each tutorial group within the college might have a sub-area, while each academic department might have their own top-level workspace with lectures and seminars as sub-areas. An individual workspace might exist in multiple hierarchical positions (e.g., a Latin tutorial workspace might be found either from a college or from a department). Memberships and roles within a workspace might be inherited from a "parent site" regardless of hierarchical positions - in effect, treating an existing workspace within the LMS as an "external" authorization source. (Matthew Buckett, Adam Marshall)

Maintaining links to sources

Sources of memberships and roles generally have more to them than a bare list of user identifiers. It may be important to preserve access to them even aside from their status as feeds.

For example, during much of a term, the most important aspect of a student's enrollment in a class might just be that the same person shows up in the course's online workspace as a "student". When it comes time to submit final grades, however, the integration may need to check on enrollment properties such as pass/fail selection or the elected amount of credit.

As another example, a tutorial section's leader and meeting times may need to be maintained along with memberships.

Knowledge of the original source can also be important when negotiating conflicts between external systems, as described below. (Oliver Heyer)

Changes to available source groups

In August, an instructor creates a workspace for Fall 2008 Chem 1a, integrating its memberships and roles to memberships and roles associated with the class's two lecture sections, five lab sections, and two discussion sections. In September, the chemistry department realizes that there are only enough Chem 1a enrollments to fill four lab sections, so they cancel the fifth. "Chem 1a Fall 2008 Lab 005" then disappears from the course management system. At the end of September, the instructor goes to check section memberships in the course workspace. This should not fail because of the "missing" lab. Either the integration should gracefully handle the concept of "all existing sections, dynamically determined" or the instructor should be notified that the source section no longer exists and be given the chance to remove that link. (Oliver Heyer)

Modifying derived memberships

The link between online memberships and externally-defined memberships is not *equivalence*. We may want to give online memberships to people who do not officially (or do not yet officially) play a role according to SIS. That means reconciling multiple sources of membership changes. For example, an instructor wants to let officially enrolled students automatically gain access to her class workspace, but doesn't want to punish students who are still working their way through SIS official channels, and also doesn't want to have to monitor the status of every "early" student separately. Workspace membership therefore needs to be based on a combination of external integration and internal intervention. (In a local integration, we automatically converted "manual" memberships to "externally provided" memberships when they first showed up in the feed. In Sakai 2.* course site management, "manual" memberships stay manual. In Sakai 2.* section management, section memberships are either all-external or all-manual, but can be switched from one type to the other.) (Oliver Heyer)

We will also want to be able to create and manage new subsets of a membership list, while maintaining a dependent link from the source memberships to external systems. (That is, if a student drops the course, that student no longer appears as an active member of our inside-the-workspace "Third Week Research Project" team.) A few special examples follow.

3. Managing subgroups of external members

Space-limited student sign-up

Each year, about 100 medical students sign up for elective projects (with associated online workspaces) with a supervisor. Each supervisor can only work with 4 students at most. Some supervisors and topics are much more in demand than others, and so competition can be fierce.

A lecture course with a typical enrollment of more than a thousand students is associated with fifty tutorial groups, each with their own meeting times, leaders, and mailing list. Again, it's important for enrolled students to be able to compete fairly for spots in their most desired group. (Stephen Marquard)

Randomized grading groups

The Instructor of a 100-student lecture class has 3 TAs who grade all assignments. Each student must complete all of the same assignments. The Instructor divides the class into 3 groups for each assignment, each group comprising different, randomly chosen members. (Motives might include breaking up TA-student connections, encouraging a demographic balance, or simple convenience.) Each group is assigned a TA who is responsible for grading the associated assignment. (Oliver Heyer, David Horwitz)

The remaining pedagogical use cases are all slavishly copied from Clay Fenlason's list of Georgia Tech Group and Course Requirements*.*

- **"Capstone" experiences**extend across 1 or 2 semesters with groups of varying size, but typically around 4-5. These students will produce project deliverables of varying types (written and oral reports, software solutions, peer ratings, etc) as they move through different stages of the design and often work with customers external to the university.
- **Problem-based learning**: Several problems during a term are introduced to students who then may work in teams of 3-5 and need to pull together information from a variety of resources in order to fully address the problem. Student presentations of their work on the problems may be included.

- **Group notetakers**: Rotating student groups of up to 10 or so students take notes for a single course and produce editable public versions of those notes for the rest of the class.
- **Video research projects**: Teams of students from a single course perform a technical research project and document solutions via the production of a video.
- **Distributed teaming for engineering design**: Students from multiple campuses collaborate on teams for the purpose of creating design solutions to complex problems. Teams typically confined within one term but may be enrolled in multiple courses.
- **Paired problem solution presentations**: Students choose or are assigned a partner to work with on daily or weekly problem sets. One of these pairs is then randomly selected to present their attempt at a solution for one of the problems from the recently assigned set. The class discusses the problem.
- **Vertically integrated project teams**: These support large scale, complex, multi-year projects in engineering with perhaps a dozen students from sophomore through graduate level and from a variety of majors. A single student may participate in the project for 1 or more years but students enter and leave the team each term. External stakeholders may have significant roles to play.
- **"Jigsaw"**: Students work in small groups to develop knowledge/expertise about a given topic and to determine effective ways of teaching this knowledge to others. These "expert groups" then break up and move to new "jigsaw groups." (Each jigsaw group now consists of students who have developed expertise in different subtopics.) Each "expert" in the jigsaw group then assumes the role of teacher and instructs the other members of the jigsaw group about the subtopic.
- **"Test-Taking Teams"**: Students work in teams to prepare for exams and then take the exam first individually and next as a group.
  This collaborative learning process involves three steps: (1) the group studies for the exam together; (2) individuals take the exam; and (3) the group takes the exam (students rejoin their groups to reach a consensus on the answers and submit a group response to the test). Rationale: By working together to prepare for the exam, students help each other understand the content. Because each student first takes the test independently, this process emphasizes individual accountability. By retaking the test as a team, individual students benefit from the collective knowledge of the group. This technique is used for both short quizzes and tests for covering larger amounts of material--and since the group score is usually better than the individual scores, the technique is often used to demonstrate the value of collaborative learning. Individual grades are determined by using both individual test grades and group test grades. Scores are weighted (for example, two-thirds for individual plus one-third for group).
- **"Feedback Teams"**(Peer-Review, Peer Editing): Students critically review peer work (either in pairs or small groups) and provide feedback on each other's essays, reports, research papers. or projects. This works best when students are given guidance on what to look for (rubrics) and guidance on how to make constructive suggestions. Once feedback is exchanged, students take feedback into consideration and revise/prepare their final product. Typically students include their peer review forms when they submit their completed product to the instructor for evaluation.

4. Federating membership sources

Merging external memberships

We frequently need to create project-oriented, departmental, or cross-listed workspaces whose memberships are drawn from multiple courses. Naturally, however, we can't accidentally let a teaching assistent in a lecture class grade herself in a seminar. For this and other reasons, it's important to preserve backward links to the original information sources, and to define clear ways to reconcile role-mapping conflicts. (Oliver Heyer)

Merging from multiple external authorities

A CLE may draw both on SIS-controlled course management data and on personnel or other data.

For example, along with the usual course sites and project sites, I may want to grant special access to departmental administrators or to set up an alumni workspace. These external systems are frequently administered by different external bodies and use different role and group vocabularies. Mapping needs to be configured as part of any integration.

Also, in a large university, sources of membership for a single person often overlap. (E.g., a university museum curator might also teach a course, or a staff member in the multimedia department might be a graduate student.) Again, reconciliation methods are needed.

Finally, when searching for contacts, I might want to combine criteria from different integration sources ("undergraduates" vs. "staff"; "English tutors" vs. "Chemistry tutors"). (Kirk Alexander. Oliver Heyer)

Cohort spaces with course-specific areas

Boston University hosts student Teams which span courses and sites. Each Team coordinates on a long-term project and attends courses as a Team. Group grades are factored into individual grades. In MBA and some other programs, a similar long-lived student grouping is called a Cohort, and the Cohorts are enrolled in course offerings as a unit.

Ideally in such an arrangement, the following site hierarchy would be supported: A primary site would operate as the Team or Cohort collaboration space, and administered by a special coordinator not necessarily represented by enterprise data. Sub-sites would be course sites governed by individual instructors and registrar data. The course sites tend to be peripheral to the students' collaborative work. (Clay Fenlason)

Supervising students across courses

At Cambridge University, supervisors are graduate students who have been assigned a group of students. In some cases, the supervisor may supervise the students across quite a few different courses; in other cases, the supervisor sticks to just a course or two. The balance varies by discipline and by supervisor. Inside a course context, a supervisor sees the enrolled students who are assigned to her. (Laura James)

Shared educational resources

An instructor who is going to teach the next semester of Calculus 101 automatically has access to the workspace and resources associated with the "canonical course" of Calculus 101 across all semesters. Within the workspace associated with that semester, the instructor can either link directly to the cross-semester resources or make local revisions. (Stephen Marquard)

Access to previous work

A student who took Freshman Comp three years ago refers back to that class's associated workspace, including their own contributions. (Stephen Marquard)

Alternatively, students create and manage their own project sites and portfolios, linking to (and storing) course-related or project-related material across multiple courses, working groups, and applications, with easy access to and from the original context. (Adam Marshall, Stephen Marquard)

Program goals and course activities

Program goals (viewed at a discipline-level context that may cut across multiple departments) link into course activities (mostly defined and met in course-specific contexts). A program coordinator can view the goals and ratings published by course instructors but can't necessarily change them. A course instructor can link course activities to program goals and ratings, but doesn't necessarily get to see goals and ratings published from other courses. (Sean Keesler)

5. Integrating applications and services

Application-determined privileges and roles

Multi-user applications and services often need to treat some users differently than others. A high-volume highly configurable discussion board tailors access to activities such as "attach files", "use a signature", "disable word censors", "upload an avatar", and "approve posts". A chat room gives "Moderators", "Speakers", and "Members" slightly different interfaces and capabilities.

Although integrated authentication is fairly common, most collaborative applications and services do not bother to expose their internal notions of roles and privileges to external management. However, plugging into a large collaborative framework sometimes forces such exposure. We can't ask instructors, researchers, and students (or even long-suffering administrators) to separately manage groups, roles, and privileges of dozens of applications and services across thousands of instances. To reduce the load, Sakai adds a level of indirection, bundling application-level mappings into larger and more familiar cross-application community roles. Instead of forcing administrators of five thousand course workspaces to assign hundreds of functions like "Course grader", "Section grader", "Grade receiver", "Quiz editor", and "Quiz taker", we let installation-wide roles like "Instructor", "Section leader", and "Student" decide them by default when possible.

Ideally, this decouples plug-in development from administration of local community roles. However, as with any public interface, careful design is needed. For example, if the application developer naively opens file-system-like CRUD access permissions to each object type in its internal data model, it becomes almost certain that integrators will accidentally create rights-combinations that wreck key assumptions of the UX or API design. Some other issues follow.

Adding a new plug-in to an existing installation

When a new application or service is added, existing users and workspaces need reasonable access to it. Among other things, this generally means adding the new plug-in's privileges to mappings from existing roles.

Updating a plug-in in an existing installation

Since roles and privileges are so central to application and service functionality, it frequently happens that a new feature or a fixed bug leads to a new plug-in privilege. An existing permission may even be given a new interpretation. (This is a good example of how not to maintain a public interface, but it has been known to happen.) Such software upgrades must include installation-wide changes of role-to-privilege mappings to prevent existing sites from behaving in unexpected ways.

6. Supporting local installations and institutions

Support existing institutional roles

Reflecting a wide variety of institutional organizations around the world, contextual roles in higher education differ widely. A "GSI" or "Head GSI" at UC Berkeley may have little in common with a "Tutor" or a "Coordinator" elsewhere. Most schools maintain (even if optionally) at least some basic split between "teacher" and "learner", but expectations (and therefore permission-mappings) differ even there. In some environments, for example, a lecturer might not be a grader, and a person who grades assignments might not have the right to view or change overall course grades.

Workspace types and other role-mapping hierarchies

Reflecting a wide variety of pedagogical approaches, community support levels, and research orientations, more than one set of roles or permission-mappings are needed within a single CLE installation.

The fact that someone grades a group of students in a lecture course does not dictate whether they have edit privileges in an independent research project's workspace. To meet this need, Sakai 2's realm-template approach lets installations define a set of workspace types, each with their own standard roles and mappings. The two types most commonly seen are "project sites" (with two default roles, "access" and "maintain") and "course sites" (with three default roles, "Instructor", "Student", and "TA"). (Unenforced expectations about these two types have resulted in a number of hard-to-catch Sakai 2 bugs over the years.)

The fact that someone is enrolled in a seminar doesn't always determine their function in a wiki or a discussion forum in that seminar's workspace. Default mappings are essential to reduce the cognitive cost of plug-ins, but those defaults frequently have to be overridden. (Which is another reason to treat exposed plug-in privileges as a public interface: they need to be comprehensible to non-developers if they're expected to be adjusted by non-developers.)

Even within a site type, expectations of associated roles may change from place to place within an institutional hierarchy. Medical, law, and business schools frequently support very different activity flows and responsibilities from undergraduate courses in the same institution. Virtually all the thirty-plus organizations at Cambridge University maintain slightly modified ideas of role functions.

In effect, we need a flexible hierarchy of role-to-privilege defaults which support easy deployment and maintenance but can be customized without unbearable pain.

New institutional/installation roles

After defining sets of roles and mappings, they're unlikely to stay static. Occasionally, institutional changes or new features lead us to define a new role with a new set of permissions: "Head TA", "Concurrent Student", or "Visiting Staff", for example. When users start showing up with the new role in existing contexts, its application-privilege mappings need to be in place.

Modified privilege mappings

A role's permissions might need to be changed due to a change in security policy (for example, "SIS is going to pull the plug unless we immediately block students from seeing other students' official roster photos"), or due to having misunderstood the meaning of a permission in the first place. When this real-life problem comes up, it needs to be solved in a hurry for all affected sites.

(This requirement might occasionally conflict with the desire for easily customized role-to-privilege mappings.)

7. Outside the ivory firewalls

This extensive list has stayed within the confines of a large university (perhaps with some cooperating partners via Shibboleth). However, the combination of financial pressures, user expectations, cloud computing services, and increased cooperation among social networking hosts moves the problem of federated authorization beyond the LMS/CLE border. Solutions to these new challenges should play a part in solving many of the old ones as well.

- Open access to selected LMS/CLE resources via OAuth.
- Directly reference social or professional contact lists from the LMS/CLE.
- Securely host all or part of an LMS/CLE on Google Apps.
- And so on...