Roster Synchronization

Title:	*Roster Synchronization*
Jira Number:	SAK-16693
Related Jira Numbers:	*TBD*
Component(s):	*Matrices, Evaluations, Wizards, Groups, Roster*
Author:	*Lynn Ward, Indiana University*
Date:	*July 1, 2009*
Demo Status and Date (s):	Status: *Pending* Date: *Demoed most recently on February 15, 2010*

Part 1: Functional Description

Roster Synchronization (1)

Summary (1)

Roster synchronization is a new option under "Manage Site Associations" in the Matrices tool that functions as a toggle. When another Sakai course or project site is associated with the current site with roster synching turned on, the students (or members) of the associated site are automatically added to the current portfolio site and a new editable group with the members of the associated site is created. As members are added to or deleted from the associated site, the roster and associated group membership in the portfolio site is dynamically updated. The feature reduces the administrative overhead of manually updating he portfolio site roster when assignment linking is used in a large number of course sites.

Rationale (1)

At IU, the rosters for all course sites are autopopulated from SIS data, but portfolio rosters must be maintained by hand. Many of our portfolio projects involve collecting examples of student work from multiple courses over multiple semester.   To maintain the rosters for these projects, the coordinator must import the roster for each course participating in the project at the beginning of the semester, and add individual students that enroll in the course following the import. This is a labor intensive process and late enrollees are inevitably locked out of the portfolio site until the site coordinator is notified. This feature eliminates the need to manually populate and maintain the portfolio site participant roster in cases where the roster is made up largely of course sections.

Origin (1)

The need for this enhancement arose at Indiana University in connection with two early portfolio projects, the English Capstone portfolio and the Transition to Teaching Portfolio. In both projects, students enrolled in specific courses are required to complete a portfolio matrix or wizard published in a portfolio site. The same site and scaffolding is used each semester to facilitate longitudinal reporting. In both cases, after one semester of maintaining the roster manualy, the site coordinator requested an automated approach. The need has become even more pressing as we've begun to mount large-scale programmatic assessment projects in which multiple course sites are associated with the Matrices tool each semester.

User Stories (1)

The User Stories should paint a picture of what it is like for a user to make use of the enhancement. The actors should be based on real users with definable tasks and goals. Include as many stories as necessary to demonstrate how the enhancement would be used by different types of users.

Actors and Stakeholders

- Portfolio site coordinator - this feature is intended to benefit the site coordinator by automating a process that is otherwise hadnled manually. Instead of manually importing rosters from associated sites, the coordinator enables roster syncing
- Instructor - without this feature, instructors teaching associated courses must notify the portfolio site coordinator when students add or drop the class after the initial roster import.
- student/participants - the impact on students is negligible, since they are not typically aware of how they are added to or removed from portfolio sites.

Coordinator Associates Sites with Roster Syncing Enabled

1. Coordinator establishes a portfolio site for programmatic assessment
2. Coordinator creates and publishes a matrix that represents the department's essential learning goals or outcomes
3. Coordinator associates multiple courses taught by the department with the Matrices tool and syncs the rosters.
4. Students in each associated course sites are added to the portfolio site and a separate group is created for each associated site.
5. As students add or drop one of the associated, the portfolio site roster and group membership is dynamically updated.

Coordinator Disables Roster Syncing for a Course

1. Coordinator learns that the enrollment in one or more courses taught by the department is primarily non-majors who should not be tracked in the matrix. Only enrolled students who are majors should be added to the portfolio site
2. Coordinator turns roster syncing off for the designated courses.
3. The students in the associated courses for which syncing was disabled are removed from the portfolio site roster and group unless they are in another associated course.
4. The coordinator manually adds the majors from the courses in which syncing was disabled to the portfolio site and corresponding course group.
5. The course sites in which syncing was disabled continue to be associated with the portfolio site, so the instructor can continue to link assignments to matrix cells in the portfolio site.
6. When students submit linked assignments, those that were manually added to the portfolio site will see their linked assignments in the matrix.

Functional Details (may be added after community demo) (1)

See OSP Roster Automation Design Document.

Interaction and Implications (1)

At the current time, roster synching only affects users in the following roles:

- Course sites: student
- Project sites: student, member
- Portfolio sites: participants

When roster syncing is enabled, users in the above roles are added to the portfolio site as "participants"

Roster syncing affects other Sakai tools in the following ways:

- Site Setup - users added to the portfolio site appear in the roster as participants.
- Site Setup > Manage Groups: a separate group is automatically created for each associated site with roster syncing enabled. The group can be manually edited. If roster syncing is turned off and the only group members were those in the associated site, the group is deleted automatically. If the group contains members that are not in the associated course, the group remain intact.
- Roster - participants provided via roster syncing are listed in the site roster and identified as members of the groups created when the site was associated.

Diagrams and Mockups (3)

See OSP Roster Automation Design Document.

Community Acceptance (4)

This feature was described in two presentations at the 2009 Sakai Conference in Boston. In addition...

Part 2 of the Proposal for Enhancement Template: The Specification
The specification should be filled out once the feature is clearly defined.

Specification Template (5)

Behavior

See OSP Roster Automation Design Document.

Quality Metrics

Describe any non-functional requirements of the feature, such as usability, performance, or design. Provide objective and, where possible, quantitative measures against which actual implementations can be evaluated.

Assumptions

Provide any assumptions about implementation path, availability of other required features, schedule concerns or otherwise.

Outstanding Issues

The Outstanding Issues section is a placeholder for the evolution of this specific feature. It should mention any explicit design or implementation decisions that are pending. There must be no outstanding decisions as of the confirmation of the feature as a requirement.

Configuration Information

For those folks interested in customizing the configuration, here are some examples and a few options for implementing those customizations.
The default configuration might look something like this:

```
<!--
Configuration object which can safely be overridden by a deployment's
sakai-configuration.xml file.
-->
<util:map id="org.sakaiproject.coursemanagement.GroupProviderConfiguration">
        <entry key="siteRoleResolutionOrder">
                <list>
                        <value>Instructor</value>
                        <value>Teaching Assistant</value>
                        <value>Student</value>
                </list>
        </entry>
        <entry key="officialInstructorToSiteRole" value="Instructor"/>
        <entry key="enrollmentStatusToSiteRole">
                <map>
                        <entry key="enrolled" value="Student"/>
                        <entry key="wait value="Student"/>
                </map>
        </entry>
        <entry key="sectionRoleToSiteRole">
                <map>
                        <entry key="I" value="Instructor"/>
                        <entry key="GSI" value="Teaching Assistant"/>
                        <entry key="S" value="Student"/>
                </map>
        </entry>
        <entry key="courseOfferingRoleToSiteRole">
                <map>
                        <entry key="CourseAdmin" value="Instructor"/>
                        <entry key="I" value="Instructor"/>
                </map>
        </entry>
        <entry key="courseSetRoleToSiteRole">
                <map>
                        <entry key="DeptAdmin" value="Instructor"/>
                </map>
        </entry>
</util:map>
```

At IU, our config looks like this:

```
<util:map id="org.sakaiproject.coursemanagement.GroupProviderConfiguration">
        <entry key="siteRoleResolutionOrder">
                <list>
                        <value>instructor</value>
                        <value>teaching assistant</value>
                        <value>student</value>
                        <value>participant</value>
                </list>
        </entry>
        <entry key="officialInstructorToSiteRole" value="instructor"/>
        <entry key="enrollmentStatusToSiteRole">
                <map>
                </map>
        </entry>
        <entry key="sectionRoleToSiteRole">
                <map>
                        <entry key="student" value="participant"/>
                        <entry key="access" value="participant"/>
                        <entry key="participant" value="participant"/>
                        <entry key="member" value="participant"/>
                </map>
        </entry>
        <entry key="courseOfferingRoleToSiteRole">
                <map>
                </map>
        </entry>
        <entry key="courseSetRoleToSiteRole">
                <map>
                </map>
        </entry>
</util:map>
```

We're not using any of the "real" Course Management stuff, so if you are, YMMV here! For our implementation, we only used the sectionRoleToSiteRole portion and had all of the "access" level roles map over to the participant role (of a portfolio site). Our roles are customized here, so yours again may be different. There's not much to it, really.

As far as where to do the configuration, you can either change the file directly (/providers/component/src/webapp/WEB-INF/components.xml), or if you have a sakai-configuration.xml file in your tomcat/sakai folder, you can override it there.