

2013 Apereo TCC Meeting Notes

Sunday, June 2nd Agenda: Who we are & how we work

- Introductions & Ground Rules
- PMC Discussion (AW/ST)
- Clean up Communication Lists (AW)
- Collapsing of the majority of indies (SW)
- Review SVN global commit changes & Kernel commit process discussion (AZ)
- Moving from svn to GIT (??)
- Documentation /Confluence revamp - BK /NC
- Tools Survey Summary (NC)
- Best practices for committing minor enhancements

Summary:

The group set forth to work on a number of proposals throughout the week

- PMC Proposal
- Project organization & release simplifications
- Documentation
- ACL's - kernel; global

Additionally, the following were discussed:

- An attempt to consolidate Sakai email lists will be undertaken in collaboration with groups using those lists
- Will not pursue moving the repo to GIT at this time. This will be something that is revisited once simplification in svn occurs

Thursday, June 6th Agenda Theme: What we're going to do

- [Introductions & Ground Rules](#)

Sunday, June 2nd Notes: Who we are & how we work

Introductions & Ground Rules

Attendees: Megan May, Mark Norton, Chuck Sev, Alan Berg, Seth T, Beth K, Aaron Z, Sam O. Chris H., Bob Long, Neal Caidin, Jean-François Lévêque, Richard Weber, Makoto, Matt Jones, Anthony Whyte

Ground rules: No decisions are made without being circulated on list so that our colleagues not in attendance can weigh in.

PMC Discussion

- Topic Leader: Anthony Whyte, Seth Theriult
- Problem Statement/Brief Summary: The Sakai TCC was created in July 2010 to work with existing community groups and processes to provide technical direction, advice, and coordination that both nurtures and enhances the Sakai CLE ecosystem. Inspired directly by the Apache Foundation's Project Management Committee (PMC), the TCC established for itself a light-weight governance model based on the meritocratic principle of member selection and featuring consensus-based decision-making practiced in an open and transparent manner. [1] Since its formation the TCC has operated as the CLE project's de facto PMC in all respects save formal recognition by the Sakai Board.

Following the merger of the Sakai and Jasig Foundations and the establishment of the Apereo Foundation in early 2013, a new model of community governance has been established for the "software communities" comprising the new Foundation. In line with this change we propose that the TCC seek formal recognition as the Sakai CLE software community's "strategic governance body" as described in Article IX, section 4 of the Apereo Foundation Bylaws. If so recognized by the Apereo Board, the TCC will assume responsibility for the "strategic, financial and operational oversight of the community, including adherence to overall Foundation licensing and intellectual property policies" as outlined in Article IX, section 3. [2]

In framing this proposal we recommend that the TCC rename itself the Sakai CLE Project Management Committee in order to both acknowledge the influence of the ASF in our thinking about governance as well as to allay fears that technical issues will constitute our only concerns as a governing body.

[1] <http://www.apache.org/foundation/how-it-works.html>, <http://www.apache.org/dev/pmc.html>

[2] <http://www.apereo.org/content/bylaws>

Discussion Notes

- TCC is very technical at this time. Would membership change? Not necessarily. Hope that people evaluate their role.
- Concern about assume responsibility for the "strategic, financial and operational oversight of the community, including adherence to overall Foundation licensing and intellectual property policies" as outlined in Article IX, section 3.
- What does this mean?

Action: Anthony will flesh out proposal more and circulate on list.

Email list review

- Topic Leader: Anthony Whyte
- Problem Statement/Brief Summary: I recommend that we slim down the email lists we support. Currently, I count 69 "Sakai" or CLE-related lists, a minimum 40 of which I recommend that we retire before the end of summer; see [sakai-mailman-lists.xlsx](#)
- Deadend? forward? Add autoresponder.

ACTION: Anthony will contact list owners & execute

Project organization and release process simplification

- Project Work
 - iRubric
- Followup on Actions from Sunday Meeting
 - PMC Proposal
 - Project organization & release simplification
 - Documentation
 - ACL's - kernel; global
- Future Release Cycles
- Deprecations/ Tool Promotions
- Project Work
 - Spring/Hibernate upgrade (NB/AZ)
 - JMS service [SAK-11021](#) (AZ)
 - Replace Commons Logging+Log4j with SLF4J+Logback (ST)
 - Usability issues (CH)
 - Back porting new trunk LessonBuilder features to 2.9.x for 2.9.3 (CH)
 - Proposal: JSR-168 Portlet as Sample Code / Preferred Tool Pattern for New Tools (CS)
 - IMS Learning Tools Interoperability 2.0 - CS
 - On the radar: Replacing the help system
 - Learning Analytics enhancements and raising their awareness (AZ/AB)
 - Support for browser back button? (BK)
 - sakai properties (undocumented ones, files, defaults, samples) (??)
 - Mooc profile for 2.10 (Alan)
 - Unfunded Mandates List
- Topic Leader: Steve Swinsburg, Anthony Whyte
- Problem Statement/Brief Summary: [cle-release-process-proposal.pdf](#)
- addendum: full-source check out (default); an emergency module release can be tagged with a fourth identifier (e.g., kernel-1.3.2.1)
- consider reorganizing core Sakai as a single project (for example, see: <https://github.com/arwhyte/octocat-cle>); our finely-diced module structure is largely anachronistic and complicates unnecessarily our workflows, issue tracking (e.g., Jira), communications, committer model, etc.
- Versions are confusing; JIRA is
- There are too many proposals
 - Consolidation (version #'s would change)
 - shifting code (ie edu services) around.
- Need to coordinate and ensure that contrib is not broken
- Anyone opposed to collapsing Indies in? No
- Anyone opposed to standardize: No
- Incremental process: Start working in trunk (not for 2.9); Work toward code organization. Phase I
 - Re-Consolidate JIRA Projects: only open & new
 - Re-Consolidate Indies
 - Re-synch up versions.

ACTION: Anthony, Matt & Aaron will discuss and think about some of the implications (ie will need 2.9 support)

Review SVN global commit changes

Topic Leader: Beth Kirschner, Anthony Whyte

- Problem Statement/Brief Summary: Our current SVN access control list (ACL) is overly complex, bloated, difficult to maintain and designed for a different era when it was assumed that largely independent clusterings of schools and committers would work on different parts of the CLE. It contains out-of-date groups and committers no longer active in the project. The new reality is very different from the old and we recommend that we review our ACL with an eye towards both rationalization and simplification.

A few thoughts regarding our current svn-acl-policy doc (arwhyte):
 svn-admin: this group at one time was limited to those administering the SVN repo, Chris, myself (in the past), Beth, etc. as well as a few other trusted souls who could step in during an emergency when an svn admin admin was unavailable. Over time, however, other names (and other groups) were added to the svn-admin group in order to shorten the time required to administer an ever lengthening and cumbersome ACL doc.

A counter-intuitive recommendation: don't slim down the svn-admin group, upsize it to include all senior members of the CLE team (admins, devs and branch managers). In other words, besides svn administrators I would include all individuals who can be trusted to range across the code base, either in the role of developer or as a designated branch manager. This would eliminate a number of module boundaries that occasionally trip up senior devs. All others would revert to contributor status and be removed. I would then eliminate the maint-team and branch-managers group as redundant.

Module level permissions (a more radical suggestion): I'd eliminate the standard module level permissions for all projects under maintenance (e.g., @gradebook = rw). In practical terms, I'd expect the number of patches to rise as a result and, alongside, the review and commit workload of the members of the svn-admin group to increase. This might be considered unacceptable but I think it preferable to the current state of affairs. Again, perhaps counter-intuitively, I think the CLE would be better off with fewer committers and more contributors. Reducing the number of committers might also make it a bit easier to manage trunk with an aim of getting it into a near-releasable state and effecting a mind shift to the future and away from the present/past (e.g., the .x branches).

We might exempt modules like the Kernel, Samigo, Lesson Builder and perhaps a couple of others where it makes sense to retain finer grain permissions—but only after careful deliberation since in line with the argument I'm not all that convinced any more that our internal fencing is all that necessary. If those in the svn-admin group were willing to handle help file and language bundle commits I'd eliminate those permissions as well. If not, retain them. Individual branch permissions would probably need to stay. See below for the variety of permissions currently in play for gradebook (see attachment). Maintaining svn permissions for all CLE modules while not difficult is, at the same time, non-trivial, especially given the bloat in svn-acl-policy.

I regard simplifying svn-acl-policy as a "small win," part of a broader strategy aimed at both rationalizing, simplifying (where possible), right-sizing the infrastructure supporting this project and the workflows under which we operate.

[svn-acl-gradebook.txt](#)

Discussion

- right now each branch (SAK-###) anyone with access to one branch should have access branches,
- There are 60 active people. 2/3 of them probably shouldn't touch kernel.
- Everyone makes mistakes. At some point lines to have to be drawn - this is OK.
- What code really needs to be walled off? Kernel, Access
- Set goal: simplify & increase involvement

Action:

- Anthony will adjust his proposal

Followup:

Proposal from Beth: <http://collab.sakaiproject.org/pipermail/sakai2-tcc/2013-June/003503.html>

Kernel commit process

- Topic Leader: Aaron Z
- Problem Statement/Brief Summary: Problem is that we have things going into kernel without any review (sometimes causing the build to break or causing regression bugs). Some people are careful to always get someone to review their code but those are not the majority. We need to discuss this for the sake of stability of the kernel and the overall product. (use of a formal code review/holding tool? gentleman's agreement? other?)

Discussion

- right now each branch (SAK-###) anyone with access to one branch should have access branches,
- There are 60 active people. 2/3 of them probably shouldn't touch kernel.
- Everyone makes mistakes. At some point lines to have to be drawn - this is OK.
- What code really needs to be walled off? Kernel, Access
- Set goal: simplify & increase involvement
- Again, with kernel commit everyone makes mistakes. Need a more formal process of review around the areas of code
- What is the roadblock with crucible?
 - Atlassian thinks the Sakai project is too big. Did get it in but
 - Wush to host is another?
 - GIT Hub
- Gentleperson's commit
 - Attach a patch to the JIRA.
 - Don't commit

ACTION:

- Aaron will adjust his kernel proposal (ie all the critical paths like access, ect)

Investigate moving from SVN to Git and Github

- Topic Leader: Anthony Whyte

- Problem Statement/Brief Summary: Suggest setting this as a goal to which we aspire.

Discussion Notes

- Not ready now
- The poll requests is a GITHUB service. Is the underlying request rather than changing software?
- What are the tools we need to promote the goals we have.
- A fun re-imagining of <https://github.com/arwhyte/octocat-cle>
- Solves process issues (commit list, kernel commit)
- ACL: only a small number of people have commit access
- Why is GIT desirable?
 - distributed system
 - When a checkout occurs, you get everything with history.
 - W/ Github there are poll requests

ACTION: None - revisit in the future

Documentation/Confluence revamp

- Topic Leader: Neal, Beth
- Problem Statement/Brief Summary: Users need an easy way to find critical information related to installing, using, and contributing to Sakai CLE. They need to have confidence that the information they are reviewing is up-to-date and understand how it is maintained. We also need a space for other types of community needs. Questions: What criteria applies to content in a new, contained documentation space? Who keeps it up to date (process & timeline for review)?

- [Documentation/Confluence revamp](#)

Discussion Notes

- Take official release documentation
- Good way to get non-technical people involved.
- Too much content and not.
- Drupal and Moodle use source control
- Need good process & structure
- Use some sort of source repository
- Look for process that exists to publish documentation from svn or github
- Are we over engineering a solution?
- What is the core content that should be
- suggestion that we crowd source this.

ACTION:

- All review the proposal and think about how we might.
- Seth will be the point (Aaron, Seth, Mark N, Beth, Neal- let Seth know)
- Revisit this on Thursday

Tools Survey Summary

- Topic Leader: Neal Caidin
- Problem Statement/Brief Summary: Data is for sharing

Discussion Notes:

- Should post some responses to the comments.
- Signup tool has a large - should include in distro?
- OSP is at the bottom of the list - should we remove?

ACTION: Use data in to aid in community decisions.

Best practices for committing minor enhancements.

- Topic Leaders: Alan Berg
- Problem Statement/Brief Summary: This is a question from our hard working Spanish colleges. Can the TCC confirm the following as best practices for committing minor enhancements.
 - Create a Jira with the feature request and ask on the lists, try to get as much information and comments about the issue as possible before starting to code. It will save time on the long term.

A clear explanation of the problem the patch is attempting to solve

- Develop the feature against latest version of trunk.

Clean patches; no formatting changes, whitespace changes

- Test the patch in a trunk server

<http://sakai-trunk.atica.um.es/portal> - Patches are listed in the portal.

- Document well the changes in jira and provide ways to test it.
- Attend the CLE Team call to answer questions or explain the need.
- Answer questions presented in the JIRA (the quicker the responses, the better)

Help Testing once it is committed into trunk.

Discussion Notes:

- Institutions need take responsibility for moving their code in (ie lead this process)
- The CLE CC should be tapped as a resource to help move this forward when there is confusion
- Don't want to

ACTION: Write up this documentation and include it in managed content.

Thursday, June 6th Notes: What we're going to do

Introductions & Ground Rules

Attendees: Megan M, Matt J, Athnoy Whyte, Mark Norton, Alan Berg, Charles Hedrick, Bob Long, Cris Holdorff, Ursala, Mary Francisi, JF, Beth, Seth, Richard Webber, Makoto, Lydia, Sam, Chuck S. [], Ian D, David H., Aaron Z., Neal, John Bush, []

Ground rules: No decisions are made without being circulated on list so that our colleagues not in attendance can weigh in.

PROJECT WORK: iRubric integration

- Topic Leader: Beth K & John Bush
- Problem Statement/Brief Summary: Currently we have perhaps a dozen institutions who have to fork their Sakai code in order to integrate iRubric functionality into the Assignments & Gradebook tools using a poorly designed patch that will likely never be integrated into the Sakai code for several reasons. A better solution would be to create an abstraction layer (e.g. ScoringService) that iRubric could implement, allowing a more maintainable solution to be checked into core Sakai. I would like to discuss distributing the effort for creating this abstraction layer in Assignments, Assignments2, Gradebook, and Gradebook 2. Asahi is interested in implementing this for Gradebook 2. We would need work in Gradebook and Assignments -- UM could probably step up to Assignments.

Discussion Notes:

- There are a couple options. Take code as is. Not abstracted well.
- John Bush is working on an API. Almost done with GB2
- Licensing is not an issue
- UI will change slightly - likely a generic icon.
- UMich will handle A1. ANI already on GB2. Indiana can put resources on this for GB and A2 (will adapt GB for IU's fork)

ACTION: UMich, ANI & IU will work on this!

Followup on Actions from Sunday Meeting

- PMC Proposal (Anthony & Seth)
- Project organization & release simplification (Anthony, Matt & Aaron)
- Documentation (Seth, Mark N., Beth)
- ACL's - kernel; global (Anthony & Aaron)

Discussion Notes:

PMC Proposal (Anthony & Seth)

- Has been focusing on conversations with individuals about proposal (plz see the top of the page)
- Anthony & Seth will be writing up final proposal (1 page max)
- On Sunday there was general consensus with moving forward with the proposal
- Should be Sakai CLE PMC or Sakai PMC
- Need to communicate that the name change will be following by changes in the way we operate.
- Do you think you have the correct skillset on the team or will morph over time?
 - Discussion about how that won't change right away but we are recognizing that changes need to be made and we're ready to tackle.
 - Some uncertainty about what responsibilities need to be taken
 - If we declare that we are the PMC, we need to be prepared to step up our game.
- One thing mentioned - delegating the nuts and bolts (release mgmt) to the cle team and focusing more on strategy.
- Strawpoll of : many +1's, no -1, 0 abstain

Project organization & release simplification (Anthony, Matt & Aaron)

- Haven't been able to get together. Will take **and keep working on**.

Documentation (Seth)

- The group from Sunday sketched out a bare-bones top-level outline of the basic info:
<http://etherpad.ctools.org/sakai-docs>

A parallel goal is to make this information presentable for inclusion on an updated Sakai Web site.

At this point, we are looking to crib "good" information from other sources and get that content under revision control. In time, we can expand this (and Mark had good suggestions), but for now, we want the basics.

A Github repo has been started for source control, and some stuff is already there:

<https://github.com/sakaiproject/cle-documentation>

Please fork, add, and edit HTML content. Again, let's keep it simple and straightforward.

- Mark Norton - tech doc is in javadoc and in confluence. Pull it out and put into authoritative source. There is an outline.
- Why not use Confluence?
 - It's a mess. Hard to know where to start/
 - lots of discussion about the tool. It was about who would do it, not how.
- Let's focus on the content and not the tool. This effort is about content.
- Believe that sakaiproject.org will likely be TCC responsibility. This exercise is a walk toward that.
- What are the expectations of the CLE CC.

ACL's

- Beth is ready to check things in
- Once changes in, Anthony wants to see us go further.

<add beth's email>

ACTIONS:

- Float final version of PMC proposal to list (Anthony)
- Work on changes to project organization & release simplification
- Documentation group will keep moving
- ACL's: Beth will check it in.

Future Release Cycles

- Topic Leader: All
- Problem Statement/Brief Summary: Where do we want to go with future releases? Anthony would also like to remove requirement to officially vote on releases. Lazy consensus.

Discussion Notes:

- If release simplification process goes through, the release process will be much easier.
- Goal so far, release a quarter. only plan for 3 maint. releases and if we need to do a 4th.
 - Plan for a light 2.9.3
- Should one release be fast (not a lot of QA resources), the next does.
- Need to make them smaller. Is this really better?
- 2.10
 - Feature freeze traditionally sept. This is not a good time.
 - Nov release of 2.9 worked because institutions were running it ahead of time so we could push out 2.9.2 for many schools to run for fall.
 - What are we doing that can lead us to better things in the future? What changes is there confidence around including right now
 - dashboard
 - keiti
 - LTI 2.0
 - analytics (Tincan API)
 - calculated questions in samigo
 - elastic search
 - Lessons CC export. inline questions and rubrics (tentative)
 - Freeze Oct 1 & make beta by Nov. Aim to release in March
 - Don't want blame to be dished out to release team if it's late.

- String freeze (don't change user interface strings)
 - In past we've given three months
 - Doesn't mean that bugs can't be fixed - notification needs to be made

ACTION:

- Float formal voting changes to the list.
- Consensus on releases going forward. Ask Neal to write it up and communicate

Promotions/Deprecations

- Topic Leader: All
- Problem Statement/Brief Summary: See list below.
 - Proposed deprecations: Hybrid (arwhyte: checked with Nico, it can go), OSP, rwiki, Profile
 - Proposed promotions: Sign up tool, Dashboard

Discussion Notes:

- Promotions - things to look at but no commitment to a particular release
 - Help?
 - Roster2?
 - Signup
 - JSF tool?
- Deprecations (Sleath with removal from release externals)
 - profile - where are the dependencies?
 - Hybrid - arwhyte: checked with Nico, it can go
 - Link tool - start by sleath
- Decision to not make changes
 - OSP - no
 - rwiki - often used in TWSIA

ACTION:

PROJECT WORK

Spring/Hibernate upgrade

- Topic Leader: AZ/NB
- Problem Statement/Brief Summary

The work has been completed to upgrade the CLE's usage of Spring and Hibernate to versions 3.2.3 and 3.6.1, respectively. These are significant upgrades with respect to functionality and fixes, yet they retain very good backward compatibility. Because they are broad changes, passive testing by developers and general QA will help identify whether any regressions have resulted. More information about the upgrades can be found on these pages:

<https://jira.sakaiproject.org/browse/KNL-517>
<http://static.springsource.org/spring/docs/upgrade/spring3/html/>
<https://jira.sakaiproject.org/browse/KNL-515>
<https://community.jboss.org/wiki/Hibernate3MigrationGuides>

Discussion Notes:

- If we were to go to 4, everything in contrib would break. Will need to do at some point
- Not done but Aaron would like to pursue: auto injection. slight startup delay. will look into later.

ACTION: It's done. Just be aware

JMS service

 **SAK-11021** - JMS service needed for passing around events in a more standard way
 CLOSED

- Topic Leader: John B
- Problem Statement/Brief Summary: We need some type of messaging system.

Discussion Notes:

- agreement that there is value in this
- Needs to work with current events service
- Should it be embedded? if we do that, it needs to be rock solid
- This is an awesome thing to do but this is an unfunded mandate.
- Should we tap rest of the Apero community?

ACTION: Further analysis needed

Replace Commons Logging+Log4j with SLF4J+Logback

- Topic Leader: Seth Theriault
- Problem Statement/Brief Summary: Replace the existing JUL+JCL+Log4J with SLF4J+Logback. This would move Sakai to a more modern logging framework and improve overall stability and performance. See <http://collab.sakaiproject.org/pipermail/production/2013-May/002115.html> for the ongoing discussion.

Discussion Notes:

- target of 2.10
- broad intrusive shallow change.
- is it backwards compatible. There is some bridging
- csev skeptical about putting resources into this one
- logback used by most other projects in Apero

ACTION: Look at Seth's proposal and give him feedback

Usability issues

- Topic Leader: All
- Problem Statement/Brief Summary

From Chuck Hedrick: "I'd like to see us start looking at compatibility across tools. Our Camden branch is having serious support issues because faculty get confused by things it would be trivial to fix. Their biggest concern is with connecting tools to the Gradebook. Their users tend to start by creating all assignments in the grade book. If you look across assignments, assignment 2, samigo, forums, and forum, the UI for connecting to the grade book is different for each. Most seriously, Samigo won't let you use an existing item. But even those that will have different UI's for it. One asks you. One has two radio buttons. Another has a choice widget and a link to create a new item. These are all reasonable choices, but we need to standardize. This is something the TCC probably has to do. Individual developers won't do it on their own. My pet peeve is inconsistencies in notification. See SAM-1607 and ASN-760, both of which are still open. Developers aren't going to do this on their own without some push from the TCC. I also suggest a search-and-destroy mission for tools that use up arrow and down arrow to rearrange items. Please let's all use the Fluid reorder tool (or some other reorder tool, I don't care). We're now at the point where we've got enough functionality in place that usability issues really should get priority."

Discussion Notes:

- Would like to address UI consistency. Perhaps tap user groups to give us a list and we hit the top 10
- What is low hanging fruit? In UI
- How to do this?
 - Ask institutions to share where they have changed things?
 - Something about a contest

ACTION: Chuck H will put together a survey and come up with the top list of issues. Will tackle first couple low hanging fruit items.

Back porting new trunk LessonBuilder features to 2.9.x for 2.9.3

- Topic Leader: Charles Hedrick.
- Problem Statement/Brief Summary: at a minimum 2.9.3 should include CC export. inline questions and rubrics should probably wait a bit longer, but not forever. I'd like to avoid having the copy of Lessons in 2.9 become an orphan.

Discussion Notes:

- With the discussion about the 2.10 release, will hold off
- Lots of discussion of keiti and putting in branch. Consensus to put out it in maint. branch.
 - General comments to be sure to review security

ACTION:

- With the discussion about the 2.10 release, will hold off
-

IMS Learning Tools Interoperability 2.0

- Topic Leader: Charles Sev.
- Problem Statement/Brief Summary - see below

Discussion Notes:

- key & secret done by web services
- csev done by mid july
- UMich dev team will work on interface
- new services should be 1-2 weeks in this new environment. will want to talk about where we want this to go. Chuck wants to write all of these!
- Leap ahead of IMS standards. 2.0 ratification isn't that far off - need real implementations. (2 LMS and 2 tools)
- Likely candidate for next major release (2.10 or whatever) as long as that is before june of next year. Otherwise will start looking to move in earlier.

ACTION: See above

Proposal: JSR-168 Portlet as Sample Code / Preferred Tool Pattern for New Tools

- Topic Leader: Chuck Sev.
- Problem Statement/Brief Summary

Discussion Notes:

Skipped topic

ACTION:

On the radar: Replacing the help system

- Topic Leaders: Alan Berg, Neal Caidin, Jaeques Koeman
- Problem Statement/Brief Summary: The current help system has a number of limitations. You cannot update content easily, share with other institutes and has no social aspect such as the ability to like, comment or dislike. There is an unsatisfied demand for these extra capabilities and more, which might be translated into wider institutional resources. EDIA's Knowledge Base tool is an example of a possible long term replacement. However, what is not clear is what are the conditions that need to be passed before a newly minted help system is accepted by the TCC to be included in a version of Sakai CLE.

Discussion Notes:

- Requirements are fleshed out quite a bit
- There needs to be a transition strategy. There are technical changes that will need to be made across tools.
- Elizabeth is going to be sending out an update on where this is. Need to get this plugged into technical community.
- Neal, Elizabeth and Sam will get documentation up to date for 2.9.3 (Perl script)
- Edia will need some assistance in this.
- Does not need to go through "Incubation" - this would just be a tool promotion

ACTION:

Learning Analytics enhancements and raising their awareness

- Topic Leaders: Alan Berg, Aaron Zeckoski
- Problem Statement/Brief Summary: Enhancing LMS's with Learning Analytic is a differentiator and competitive advantage in the market place. Using standards avoids vendor lock in and allows for the maximum amount of compatibility. TinCan API is a protocol that allows LMS's (Sakai, BlackBoard, Moodle are looking at this) to pass student activity streams to a secure central repository ready for further analysis. As a partnership between UvA and Unicon the feature is already implemented and can be enabled in trunk through additions to sakai.properties and compiling a provider. We should consider the actions needed to be taken for 2.10 to increase awareness and ease of implementation. Further, the piece of the jigsaw puzzle that is still missing is a BasicLTI implementation of an LA dashboard that pulls data from the Learning Record Store. We should look towards supporting the work of the Apereo LA community and opening a synchronizing dialogue.

Discussion Notes:

- This is in trunk and a part of 2.10

ACTION:

Support for browser back button?

- Topic Leader: Beth
- Problem Statement/Brief Summary: This seems to be mostly a problem with velocity based tools that have a lot of server side state and don't send any tracking information to the client.
 - JSP, springMVC, RSF, and Wicket work well (Wicket supports versioned pages so back button works OK in those tools)
 - It seems to also work pretty well in JSF tools when reading, if you create content and press back in most tools there is an NPE (Error getting property)
 - In velocity, the pattern is based on action/refresh so the url never changes. So you have something like panel=Main&sakai_action=doNewannouncement going back to the same page. If there was actual a /doNewannouncement URL that was static, it seems like it the browser would be able to navigate back to it easier. As it is now, they'll often get stuck until a tool reset

Discussion Notes:

- Beth said that we should **fold this one into identifying and addressing usability issues.**

ACTION: N/A on this one.

sakai properties (undocumented ones, files, defaults, samples)?

- Topic Leader:
- Problem Statement/Brief Summary

Discussion Notes:

- AZ has been working on this one for past couple years.
- What is left? list of ?? that needs documentation. Does not require technical skills - must be great at reading Aarons mind digging through email.

ACTION:

Mooc profile for 2.10 (Alan)

- Proposal will be written up for later share
-

Review of [Important Unfunded Mandates](#) & Other lists

- Topic Leader: All
- Problem Statement/Brief Summary: Would like to see the TCC come up with 2-3 goals and seek resources for these outside group that is currently working on CLE team.

Discussion Notes:

- Tabled this item so we can focus on UI area.
- Suggest that people look at this and push to CLE team

ACTION: