

CMIS TC Face-to-face Meeting Notes

Event Date: Jan 26-28,2009

Location: Redmond, WA

Agenda: [<link>](#)

Attendees

- Al Brown (IBM)
- Derek Carr (IBM)
- David Caruana (Alfresco)
- David Choy (EMC)
- Cornelia Davis (EMC)
- Betsy Fanning (AIIM)
- Dustin Friesenhahn (Microsoft)
- Gary Gershon
- Paul Goetz (SAP)
- Florent Guillaume (Nuxeo)
- Ethan Gur-esh (Microsoft)
- Dennis Hamilton
- Martin Hermes (SAP)
- Jens Huebel (Open Text)
- Volker John (Saperion)
- Stephan Klevenz (SAP)
- Tony Lee (Amdocs)
- Ryan McVeigh (Oracle)
- Greg Melahn (IBM)
- Pat Miller (Microsoft)
- Florian Mueller (Open Text)
- John Newton (Alfresco)
- David Neuscheler (Day)
- Conleth O’Connell (Vignette)
- David Pitfield (Oracle)
- Norrie Quinn (EMC)
- Craig Randall (EMC)
- Julian Reschke (Greenbytes)
- Steve Roth (Oracle)
- Patrick Ryan (IBM)
- Spencer Shearer (Exalead)

Day 1 – January 26, 2008

Use Case Review

([Slides from this session can be found here.](#))

Use cases were reviewed/discussed, and questions raised.

A set of potential capabilities to add to the spec were identified, and we agreed to discuss them later during the face-to-face meeting. These capabilities were:

Capability	Notes
Tagging	I.e. Web 2.0-app style tags
Observation	A.k.a. “eventing”
Data Dictionary	
ReST binding improvements	Specifically: <ul style="list-style-type: none">• Facilitating upload via standard web browser mechanisms.• Reducing the verbosity of messages to facilitate mashup development.
Unified Search	A mechanism for a search application to efficiently crawl content stored in a

	CMIS-compliant repository, incl. discovering basic security information & an efficient way to determine incremental updates.
Batch operations	Note: <ul style="list-style-type: none"> If we consider batching as a potential feature addition, we would review the “import” concept included in the Java Content Repository Specification.
Policies vs. ACLs	We agreed that if we can directly incorporate an ACL model into CMIS, we should consider removing the “Policy” object entirely for v1.
Exposing additional base metadata columns	For the following types of information: <ul style="list-style-type: none"> Records Management Digital Asset Management

Questions & Answers from the session

Question	Answer
Did we consider having programming language-specific bindings (e.g. Java, C#, etc.) as part of the CMIS specification?	Not as part of the explicit specification, which is intended as an over-the-wire protocols. But the TC would certainly encourage the development of re-usable language-specific libraries (with leverage the CMIS bindings), to make it easier for developers to work with CMIS-compliant repositories in their applications.

“How did we get here”?

([Slides from this session can be found here.](#))

Questions & Answers from the session

Question	Answer
Same-name siblings: Since CMIS doesn’t explicitly define a notion of path, is it implied that repositories MUST allow same-name siblings to be CMIS compliant?	No. CMIS in no way mandates that a repository must support same-name siblings, although the spec doesn’t prevent it. If a repository doesn’t allow same-name siblings and an application attempts to create one, it should throw a “constraint violation exception”. Per relationship type.
Relationships & version-independence: Is it the intent of the CMIS spec to specify that version-dependence for relationships is scoped to ALL relationships in a repository, per Relationship (Object) Type, or per Relationship instance?	

Test Compatibility Kit (TCK) and Reference Implementations (RI)

After discussion, we agreed to the following parameters/next steps for the TC with regards to TCKs & RIs.

- 1) The official work of the TC should include defining an explicit set of test cases for the various elements of functionality in the spec. These test cases should be sufficiently explicit that anyone

- (incl. a certification authority of some kind) could execute those test cases uniformly against any CMIS implementation and determine which test cases “succeeded” as defined and which “failed”.
- 2) The official work of the TC does NOT include writing/delivering “code” of any kind, just documents.
 - a. Therefore the TC will not directly be delivering any of the following:
 - i. An automated Test Compatibility Kit
 - ii. A Reference Implementation.
 - 3) The official work of the TC does NOT include certifying “code” of any kind as CMIS compliant. This means that the TC will not in any way be certifying any implementation as a “reference” (or as “compliant”), or any TCK as “reference” or “official”.

Show & Tell: Apache JackRabbit & CMIS

[Slides from this presentation can be found here.](#)

Day 2: January 27, 2009

Review REST Issues

[\(Slides from this session can be found here\)](#)

We reviewed several of the issues listed in the slides/JIRA. Notes per issue are listed below.

Issue #	Issue Summary	Discussion Notes	Next steps for...
18	Naming of the REST binding	Rest is a style, not a protocol. We agreed we should update the name in a way that accurately describes it as “REST-ful”, but in a way that more appropriate uses the term. (E.g. “REST-ful AtomPub binding)	David Nuescheler
19	Headers	Let’s drop the CMIS-header mechanisms for passing arguments to methods in favour of only using URL parameters. Headers interfere with bookmark-ability, and are more frequently dropped by intermediate processing agents.	Al Brown
26	Short names for link types	We should make our link type names generic (i.e. remove the CMIS- prefix) need to register our CMIS link types with IANA	Al Brown
28	PUT vs. PATCH for updates	We’ll use PUT for full item updates, PATCH for partial item updates.	Al Brown
29	Location headers	This is just a cut-and-paste error that needs correcting.	Al Brown
31	Methods for creating items (entry vs. stream vs. entry w/encoded stream)	We should conform to AtomPub, with a note in the spec that says repositories MAY not allow use of all 3 methods in a given location because of their own internal constraints (e.g. in a routing-type scenario, we should mention that the repository may reject just POST-ing the media with no properties -- because at POST-time the URL becomes fixed under REST dogma, and we know we'll want to change the URL after properties are received.)	Al Brown

Questions & Answers from the session

Question	Answer
What about a protocol binding to a CMIS specific Microformat rather than AtomPub? That would allow us to greatly reduce the payload size.	For version 1.0, we thought that the benefits of leveraging the existing AtomPub eco-system outweighed the additional complexity it imposed (e.g. payload size bloat). However, for future spec versions, adding a Microformat binding should be considered.

Discussion Topics

Exposing Records Management Metadata

We looked at 3 possible Records Management-related scenarios for CMIS:

- 1) Exposing Policy/Retention information:
 - a. Conclusion: We believe that this would be valuable for existing use cases like e-Discovery, and could be added with minimal burden on existing repositories (e.g. read-only metadata columns that can be defaulted to some form of “null” if the repository does not have native records management capabilities)
- 2) Affecting/applying policies & holds.
 - a. Conclusions:
 - i. We agreed that the potential customer value of a unified hold system would be very large... however we at this time don’t believe that there exists a reasonable way to map the differing hold-related capabilities of various repositories in a uniform way, at least not in time for CMIS 1.0.
 - ii. However, we can revisit this decision if/when we remove “policies” from the CMIS specification (as policies might be a way to apply holds).
- 3) Trying “what if” retention scenarios, as mandated by the DoD 5015.2 standard.
 - a. Conclusion: This is definitely out-of-scope for CMIS v1.0.

Tagging

After an initial review of the scope of Tagging, we don’t believe that not enough repositories support tagging natively or consistently enough to include a tagging concept in CMIS. Here are some examples of questions that are still unclear:

- Would we allow tags to be automatically applied to items based on their location?
- Can tag values be “normalized”? (E.g. can the name/meaning of a tag be changed, and if so how are tagged documents affected)?
- Are tags applied to Documents or Content Streams (or both)?
- Does tagging “update” an object? (E.g. can you tag a read-only object? Does tagging affect the modified date of the object?)

- How would tags be query-able? (E.g. is there Boolean logic queries for tags, like “All documents tagged with <foo> and NOT tagged with <bar>)?
- What metadata is stored about the tag? (E.g. Can you discover who tagged an item? When they tagged it?)
- Would we support social tags (i.e. where anyone can tag an item) vs. “authoritative tags” (where only experts can tag an item)? Both? How would they interrelate?
- How would tag cloud-style aggregations work? What metadata about the tags would need to be stored/available? (Frequency of usage? Recency of usage?)

However, we’ll collect some more data on this via a survey, and we can review accordingly.

Unified Search

We agreed to consider a proposal for supporting Unified Search in CMIS v1.0 (see Next Steps).

Functional requirements for Unified Search:

- 1) A way for the search agent to efficiently determine what has changed in a repository over a period of time (so that “incremental” search index updates can be performed), a.k.a. a “Change Log” or “Transaction Log”
 - a. The log would need to include information about adds/updates/*deletes* to the repository over time. (Note that currently there’s no way in the CMIS spec for a query to identify that an object has been deleted.)
- 2) A way for the search application to determine which users can discover/read individual objects.
 - a. This is required so that the Search application can do its own trimming of results for search queries, rather than having to ping each repository for each potential result for each query.

Questions to be addressed with this proposal:

- Are we worried that not all repositories can implement a change/transaction log (push or pull)?
 - o Should we have a push model? Pull model? Both?
- What rights should you have to have to read the change/transaction log for a repository?
 - o Should consider this if/when we add ACLs to the CMIS spec? Or can we just leave this as repository-specific?
- What if you can’t accurately represent the security state of your repository as ACLs that you expose to the search application?
 - o In that case, you’d have to fall back to run-time ACL checking per query results.
- What would the scope of the transaction log be? One per repository? More?
 - o Proposal: We should start out with one transaction log per repository.
- Related: Do we have a CMIS-defined notion of search scope? Or do we assume that search apps are storing metadata about content somewhere to define a scope?
- Inheritance of events/log events vs. items:

- What about filing/folder containership? Delete a folder with N items in it → What does the trans log look like? What are expecting/mandating that search apps understand about CMIS hierarchy.
- What about security inheritance? I.e. if a single ACL changes that affects N documents, does the search app need to get N update events (which is inefficient) or 1 event (which implies the search app will understand the inheritance model)?
- What if the object type schema changes? Are those in the transaction log?
 - Or can we assume these never happen, because there's no way to change them in CMIS 1.0?
- What about search analytics? Can we expose data like what's the most popular/recently accessed content to the search app?
 - Proposal: We should call that out-of-scope for version 1.0 of the spec.
 - Same is true for auditing.
- What types of objects are we expecting to be included in the transaction log? Docs? Folders? Relationships?
- Note about JCR – the design there is a relatively “openly defined” change log
 - No mandate of which operations must be in there
 - Not a closed set of operations (so the repository can log apps that search may not care about, but other apps might, e.g. for observations)
- Observation – if we don't have support for ACLs (and ACL discovery in the change/transaction log), then using the log for observation becomes a bit more dangerous.

Observation

We don't currently believe that observation could be readily standardized via CMIS – outside of the notion of a Transaction log as described for the Unified Search use case. In particular, a few questions were raised that convinced us of this:

- Wouldn't any push model be language dependent?
- Would we have/need synchronous and/or async events? Should they trigger *before* the operation is performed or after?
- Aren't all mechanisms of registering for an “event” relatively language-specific?
- Do we have use cases that *require* observation?

SOAP Schema Issues

[\(Slides for this session can be found here\)](#)

Show & Tell: Alfresco

Alfresco presented their JUnit test suite for CMIS test cases. Also, some CMIS components (incl. a REST-binding based browser app that was built in 15 lines of JavaScript leveraging the Abderra AtomFeed libraries.)

Day 3: January 28, 2009

Access Control List Proposal

[\(Slides from this session can be found here\)](#)

We reviewed this discussion and agreed that we'd like to see an updated version of this proposal with the following parameters:

- User Discovery will remain out-of-scope.
- We should have a fixed set of fine-grained permissions.
 - This should include an ACL for "manage permissions".
- We should NOT use XACML
- ACLs should be modeled as a new "service" on objects, not as a synthetic property.
 - This is so that they don't show up in queries, need to be updated by updating the object, etc.
- ACLs should be change-able over the lifetime of an object. (I.e. they can be set after an object is created).
- ACL inheritance: There should be a way for a user to discover if a given ACL on an object is inherited or set explicitly on the object (e.g. a boolean on each ACE for "isInherited").
 - Figuring out *where* the ACE inherits from, or affected inherited ACEs would be out-of-scope for CMIS 1.0.

The TC brainstormed the outline of an updated proposal, which is the last slide of the presentation. SAP will use this as the basis of a revised proposal.

Aspect/Mix-in Proposal

[\(Slides from this session can be found here\)](#)

The TC reviewed the slides, and we are not clear on whether we actually *need* mix-ins for CMIS v1.0, or whether the use cases are already covered by concepts in the spec (e.g. relationships & JOIN'd queries).

A sub-group of the TC (see next steps) will work on a proposal regarding:

- How the use cases they currently achieve via mix-ins could/couldn't be accomplished on current CMIS spec.
- What (if any) the gaps are
- Proposed spec changes to address them.

Hierarchical Properties

We agreed that while CMIS 1.0 will not include hierarchical properties, if group members have guidance/spec proposals they'd like to say regarding namespaces that they believe will help "future-proof" the spec for the eventual addition of hierarchical properties in a future version, we will consider them.

Survey of TC membership: What changes **MUST/SHOULD** we get included in CMIS 1.0

We surveyed the group, and the consensus list of what **MUST** be done for CMIS version 1.0 of the spec:

- General bugfix/clean-up based on issues raised.
- Addition of test cases to the spec to ensure conformance/clarity of implementation.
- Updating of text style to ISO normative format.
- Access Control Lists

Things that we **SHOULD** fix (i.e. we want to fix them, will do so if we have time, but not delay release of the spec for):

- Unified Search
- Extra base metadata fields for Records Management and Digital Asset Management.

CMIS Spec Schedule

[\(Schedule discussed can be found here\)](#)

Schedule attached summarizes overall timeline, given the OASIS process. Key upcoming dates for the TC are:

- 2/28/2009: Next spec draft due.
 - Will include: Updates to ISO conformance style, bug fixes discussed earlier, some test cases.
- 3/15: Deadline for all spec proposals to be accepted for v1.0.
 - Any proposal still outstanding at this point will be deferred to v2.
- 4/1: Spec draft due:
 - Will include: All accepted proposals, most test cases.
- 4/29: Final spec draft due
- 5/15: Vote to release spec as Public Review Draft.

Next Steps (All Days)

Note: These actions items do NOT include issues already logged in JIRA.

Assigned To	Action Item	Due Date
Ethan Gur-esh	We should pull non-normative sections of spec (e.g. use cases, etc.) into a new Non-normative document (e.g. an appendix or pre-amble)	2/13/2009
Ethan Gur-esh	Spec should be updated to include chapter numbers, line numbers & page numbers.	2/6/2009
All spec editors	Spec sections should be updated to include placeholder sections for test cases with all methods	2/28/2009
Ethan Gur-esh	Include a few sample test cases for the TC to review	2/13/2009
David Nuescheler / Al Brown	Invite a formal liaison from the AtomPub TC to the CMIS TC.	2/28/2009
Al Brown	Review the latest proposals in the AtomPub TC that could be relevance to CMIS (e.g. Oracle's proposals for hierarchy/folders in feeds)	2/28/2009
Ethan Gur-esh, Greg Melahn	Add JIRA issue tracking spec proposal to add metadata for Records Management & DAM	2/13/2009
Ethan Gur-esh	Create a survey of the TC membership regarding if/how tagging is supported in their products, so we can re-consider inclusion of tagging in the CMIS spec.	2/13/2009.
Ethan Gur-esh (lead), John Newton, Paul Goetz, Greg Melahn	Draft Unified Search spec update proposal.	2/28/2009
Paul Goetz	Draft updated ACL proposal	2/28/2009
OpenText, EMC, IBM, Nuxeo	Draft mix-in report/proposal: <ul style="list-style-type: none">• How the use cases they currently achieve via mix-ins could/couldn't be accomplished on current CMIS spec.• What (if any) the gaps are• Proposed spec changes to address them.	2/28/2009