ArchReviewNotesWeds

DSpace Architectural Review

Notes from Wednesday, 25 Oct 2006 (JSE)

I. Review of Agenda

1. Workflow
2. ID management and ePeople
2. Authorization & Policy Implementation
3. Other?

II. Workflow

See diagram at []

1. Current Ingest "Workflow"

- Submission
- "workflow" (post-submission)
- install

2. Event Mechanism (Larry Stone, MS, RR)

- a general purpose notification system
- policy driven
- customizable

3. History System

- creates an audit trail
- follows an ABC ontology
- writes to triple-store

4. Preservation

5. Versioning

6. Issues
what*are the first-class items we're worried about for long-term preservation?

- establishing precedent for life-cycle management; not much experience in the field
    - lifecycle management currently is done poorly in all systems
- little things (improving current workflow) vs big things (event mechanism)
- future robust content integrty service
    - where do the artifacts

7. Providing hooks and innovations tthat enable experimentation in this area

RECOMMENDATION: Move towards improving some sort of improved history system

- which includes rigorously improved event system

8. Discussion of relationship between versioning, events and history system

- Q: what do we get from a

9. Workflow: There have proposals to RIP OUT DSpace workflow "engine"

- and replace with third-party system
- (rob) brief history of dspace workflow system
- (rj) definitely need more flexible workflow capability
- (sp) degree to which Manakin helps
- (rob) BUT aspects are baked into data model

10. Workflow: (rob) Opportunity to use flexible worklow system for implementation e.g. **preservation** workflows, etc.

- (ms) Also: a generic workflow system would help untangle administration system

RECOMMENDATION (ms) Keep "lightweight" system whilst enabling access by other systems via LNI??

- Or: re-implement "Workflow" module in DS based on third-party, open-source worflow engine & language
    - e.g. OpenWFE

- - - e.g. [Enhydra Shark](#)
    - e.g. [Lenya](#)
    - etc...
  - Q: Are there any plugin mechanisms that work esp well with any workflow systems, etc?
    - (md) See [Open Symphony](#)
      *Do as much as we can to improve workflow with Manakin
      *Investigate and recommend a third-party open-source workflow engine

11. Identify Management (hj)

- (hj) trouble with changing eperson records
  - changing email addr, etc
- (jse) what is req'd? what is 'identity" used for?
- permission control (persons and groups)
- "role" management
  - permissions and responsibility
- auditing (events/history)
  - eperson record is the source of the data
  - who did what (name, email)
- authority control
- persistent query
  - notification services
  - creator metadata
- every item has the submitting ePerson

- (jse) how is "role" specified?
- policy table
  - eperson|group, action, object
- problems occur with administration

- relationship between DSpace ePerson and e.g. LDAP
- protected data in record

- (rob) Three basic ways that identity manifest
  - There is the "stuff" to do with roles and permissions
    - getting authoritative assertions from third-party services
  - Records in the metadata
    - different set of issues
  - Notifications
    - e.g. email address
    - (rj) could abstract how notifications are done

RECOMMENDATION: It would be useful to have persistent IDs for ePeople

- that are valid URIs
  - format that the URIs could take
- aggregating metadata associated with ePerson
- should they be actionable
- they could be handles
- they could be managed by some other system
- Reminder: "Out-of-the-box" is in the manifesto
- Application-specifiable
  - format
  - some way of minting them
- Ways of importing epeople?
- (ms) making people equivalent to items

- (ms) What about the Info URI system the Rob proposed years ago
  - See ["info" URI scheme](#)

12. Authorization

- (ms) today we have a home-grown but okay for "version 1" solution
- do we re-factor for "glamorous"?
- do we fix specific problems?

- (rob) what do permissions really mean?
  - e.g. what are the semantics of a particular permision
  - bigger problem is managing permissions, ui, etc
  - there are certain inconsistencies in management
  - set of behaviors that are undocumented; e.g. changing permissions on collection, impact on other
  - whole load of unconfigurable, invisible baked-in logic
- **roles and permissions are*conflated**, which makes making a UI hard
  - (rj) can roles be aggregations of permissions, to which people are assigned?
- set of actions that are distinguished
  - roles and actions are currently mixed up, need to be clarified
  - these defined roles, these defined permissions
- (rj) do we need a way to define roles?
- are e.g. WfS1, WfS2 states or actions or????

RECOMMENDATION: Current conflation isn't working

-
  - do we incrementally change vs refactor and adopt
  - Clean up/carify specification of model
  - re-implement (or tweak) based on cleaned model

- Rob: strawman model
  - role, permissions, objects, actions
  - eperson, group

RECOMMENDATION: For workflows, rely on the AuthZ engines of an adopted Workflow Engine

- Conversely, make Workflow AuthZ a criteria/requirement of Workflow Engine selection
- (rj) presumably such an AuthZ is specific

13. Topics for Thursday and Friday

- in the perfect world, setting up a Community is a workflow step
- also, extending
- Abstract data model
  - communities/collections
  - bitstream relationships
- concrete data model & storage
- history, provenance, audit
  - admin, curatorial -> workflow
- Friday:
  - requirements