Machine Actionable METS Profiles

DLF Spring Forum 2005

Corey Keith

ckeith@loc.gov

Goals

- What we have?
  - Ideas for expressing METS profiles in machine actionable ways
  - Simple prototype tools which are profile aware
- Goal
  - Take profiles to the next step
  - Fuel the discussion. Lots of people thinking about this problem.
  - Build consensus. Make progress. Best practices.
  - Standardize expression
  - Share development effort in tool building

Roadmap

- Current Situation
- Making METS Profiles machine readable
- Sample validation tool
- Thoughts & Questions

METS Profiles

Current Situation

- Only in prose form
- Standardized container for documentation
- Requires interpretation by human
- No way to validate
  - Conformance still open to interpretation

METS Profiles

- Profiles are Good
  - Actually getting some guidance on METS usage.
  - Basis for institutional exchange of digital objects
    - Contract
- Fix METS Weaknesses
  - Flexibility double edge sword
  - Can do anything in METS
  - Encode same object many different ways
  - Little guidance and standardized practice
  - Descriptive metadata in structMap

Needs

- Take prose and make the computer understand
  - Do not have natural language processing yet!
- Machine actionable expression of the profile
- Subject matter experts write METS profiles in prose
- Developers/technologists express the prose in a machine actionable way

But why?

- Why create yet another typing language?
  - Specific needs not covered
  - Similar to XSD, RelaxNG, etc.
    - Want to create tools that are aware
  - End result will draw the best features from other data typing systems
- Why not schematron?
  - Flexible assertion based language
  - Only validation
  - Difficult to reinterpret for other uses

Holy grail

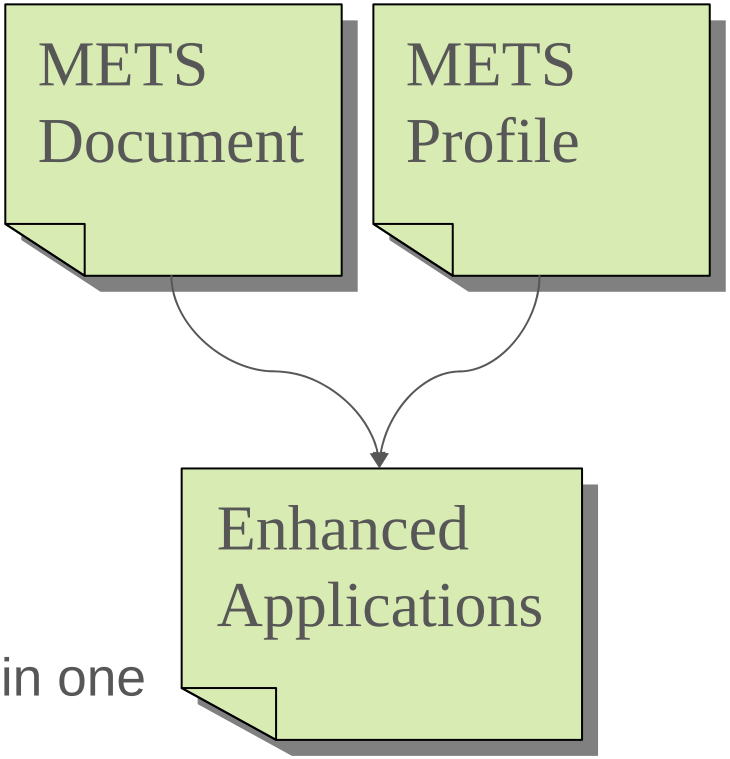
- XML Schema based editor
  - XML Schema is tough to implement
- Still does not solve our problems due to flexibility inherent in METS
- METS Editor not efficient for production
- XML Schema does not handle attribute oriented context of structMap

What’s possible?

- Validation
  - Production – Q/A
  - Interoperability
- Creation tools
- Object editors
- Input forms
- Repository data models
- Dissemination tools

Profile Aware

- Applications which support all kinds of objects
  - Validators
  - Digital Object Viewers
  - Repository Ingest
- Generic implementations
  - Added intelligence from use of profiles
  - DIV typing provides structural clues
- Intelligent pageturning
  - Not limited to display all of the images in one object
  - Get all “pages”
  - Grouping by section
- Reuse of type standardized vocabularies within structMap pays off



METS Document

METS Profile

Enhanced Applications

Profile Specific

- Applications which support specific profiles
  - lc:sheetMusic
  - lc:compactDisc
- Highly customized for particular profile
  - Not going to try to display a website object with a sheet music viewer
- Repository to Repository communication for institutional projects

div TYPE Vocabularies

- Namespaces and Qualified Names
  - xmlns:cd="http://www.loc.gov/mets/profiles/compactDisc"
  - <mets:div TYPE="cd:disc"/>
  - <mets:div TYPE="cd:track"/>
- Mixing different vocabularies in profiles
  - cd:track, loc:image
- Avoid name collisions
  - bk:page vs. sm:page
  - loc:image vs. cdl:image
- Namespaces enable versioning of type vocabularies
- Opportunity for standardization
  - METS Editorial Board endorsed vocabularies
- We don't have a format for expressing them outside of the profiles yet…

Qualified Names

- Encourage the using the same prefix used in the profile
  - xmlns:cd="http://www.loc.gov/mets/profiles/compactDisc
    - <mets:div type=”cd:track”>
  - xmlns:c123="http://www.loc.gov/mets/profiles/compactDisc”
    - <mets:div type=”c123:track”>
- String compare TYPE attributes easily with XSLT and other tools without resolving namespace prefix
- Schema aware XSLT 2.0 processors may support functions to expand qualified names
  - Change METS schema DIV type to QNAME

What we have

- Straw-man xml scheme for expressing METS profile requirements
  - Structmap/Div typing
- Prototype validation framework
  - Feasibility test for profile work

Current Features

- Expressed in XML
  - Transformable
- div typing
- ID/IDREF linking
- Structmap oriented
- Planned features
  - Metadata requirements beyond XSD
    - Required elements
    - Controlled vocabularies

ID/IDREF linking

- Confusion where to link from structMap
  - <mets:div TYPE=“cd:compactDiscObject” DMDID=“x123”/>
  - Where do you link?
    - mets:dmdSec
    - mets:mdWrap
    - mods:mods
    - mods:relatedItem

Metadata Requirements

- Extending/restricting xml schema is difficult
- Especially per class of objects (per profile)
- Lots of extension schema
- Why not put these requirements in the METS Profile?
- soundRecording profile
  - Restrict value of mods:typeOfResource
    - sound recording-musical
    - sound recording-nonmusical

Mechanics

a step towards making profiles actionable

- XML document with div TYPE’s from METS as elements
- <mets:div TYPE=“sp:simplePhotoObject”>
      <mets:div TYPE=“sp:simplePhoto/>
  </mets:div>

  becomes

- <sp:simplePhotoObject>
      <sp:simplePhoto>
  </sp:simplePhotoObject>

Simple Structmap Example

- PROFILE

```
<sp:simplePhotoObject
   pr:name="lc:simplePhoto"
   pr:minOccurs="1”
   pr:maxOccurs="1">
  <sp:simplePhoto
   pr:minOccurs="1"
   pr:maxOccurs="1">
     <sp:photo pr:minOccurs="1">
        <sp:side pr:minOccurs="0"
         pr:maxOccurs="2"/>
     </sp:photo>
  </sp:simplePhoto>
</sp:simplePhotoObject>
```

- METS Document

```
<mets:structMap>
   <mets:div
   TYPE="sp:simplePhotoObject"
         DMDID="MODS1">
      <mets:div TYPE="sp:simplePhoto">
         <mets:div TYPE="sp:photo">
            <mets:fptr FILEID="FN10027"/>
            <mets:fptr FILEID="FN1005F"/>
         </mets:div>
      </mets:div>
   </mets:div>
</mets:structMap>
```

Linked Descriptive Metadata Example

```
<sp:simplePhotoObject pr:name="lc:simplePhoto" pr:minOccurs="1” pr:maxOccurs="1">
  <pr:metadata-ref type="descriptive" required="true">
    <pr:metadata name="mods:mods" pr:minOccurs="1" pr:maxOccurs="1">
      <pr:metadata name="mods:titleInfo" pr:minOccurs="1" pr:maxOccurs="1">
        <pr:metadata name="mods:title" pr:minOccurs="1" pr:maxOccurs="1">
          <pr:text required="true"/>
        </pr:metadata>
      </pr:metadata>
    </pr:metadata>
  </pr:metadata-ref>
  <sp:simplePhoto pr:minOccurs="1" pr:maxOccurs="1">
    <sp:photo pr:minOccurs="1">
      <sp:side pr:minOccurs="0" pr:maxOccurs="2"/>
    </sp:photo>
  </sp:simplePhoto>
</sp:simplePhotoObject>
```

Linked Descriptive Metadata Example

```
<mets:structMap>
    <mets:div TYPE="sp:simplePhotoObject"
           DMDID="MODS1">
        <mets:div TYPE="sp:simplePhoto">
            <mets:div TYPE="sp:photo">
               <mets:fptr FILEID="FN10027"/>
               <mets:fptr FILEID="FN1005F"/>
           </mets:div>
        </mets:div>
    </mets:div>
</mets:structMap>
```

```
<mods:mods ID="MODS1"
    version="3.0">
    <mods:titleInfo>
           <mods:title>[Gerry Mulligan
                 and Mel Torme - 1978]
           </mods:title>
    </mods:titleInfo>
    …
</mods:mods>
```

relatedItem Example

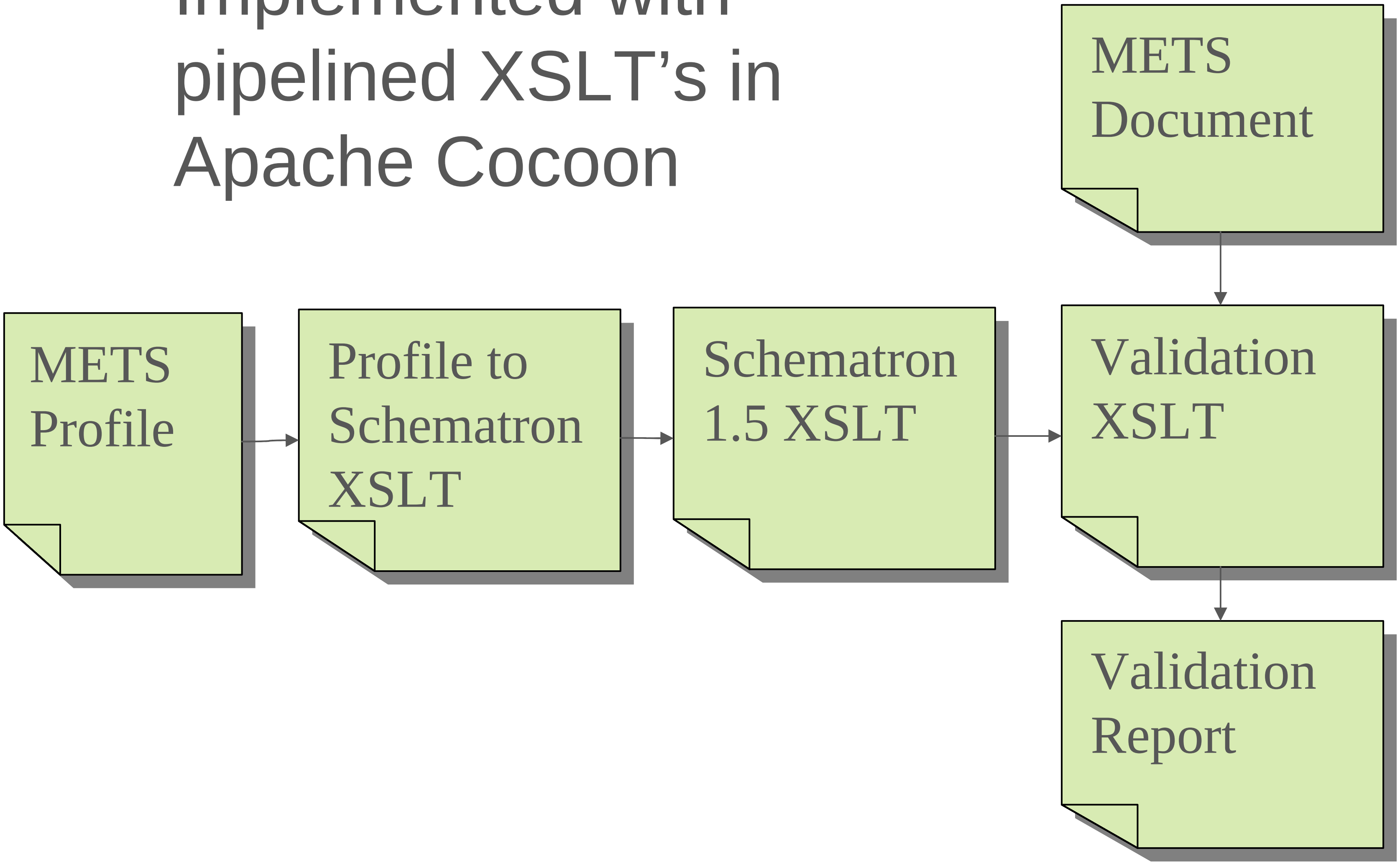
```
<cd:compactDiscObject pr:minOccurs="1" pr:maxOccurs="1" >
    <pr:metadata-ref type="descriptive" required="true" direct="false">
        <pr:metadata name="mods:titleInfo" pr:minOccurs="1">
            <pr:metadata name="mods:title" pr:minOccurs="1"/>
        </pr:metadata>
    </pr:metadata-ref>
      <cd:text pr:minOccurs="0"/>
      <cd:images pr:minOccurs="0"/>
      <cd:disc pr:minOccurs="1">
                 <cd:text pr:minOccurs="0"/>
                 <cd:images pr:minOccurs="0"/>
                 <cd:track pr:minOccurs="1">
                     <pr:metadata-ref type="descriptive" required="true">
                         <pr:metadata name="mods:relatedItem" pr:minOccurs="1"
                                        pr:maxOccurs="1">
                             <pr:metadata name="mods:titleInfo" pr:minOccurs="1"/>
                         </pr:metadata>
                     </pr:metadata-ref>

                     <cd:audio pr:minOccurs="0"/>
                     <cd:text pr:minOccurs="0"/>
                 </cd:track>
      </cd:disc>
</cd:compactDiscObject>
```

relatedItem Example

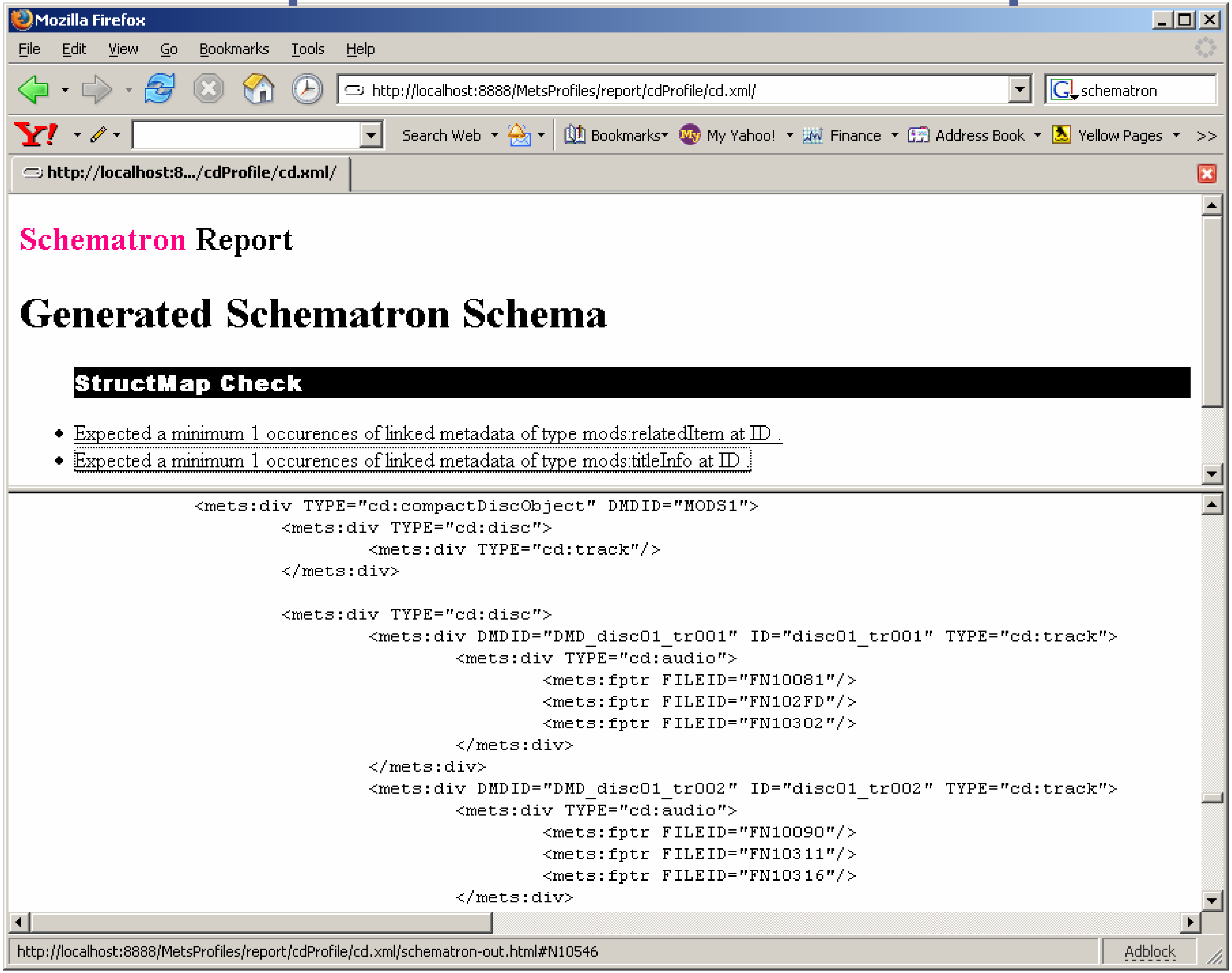
```
<mets:div TYPE=“cd:disc” DMDID=“DMD_disc01_tr001”>
…
</mets:div>


<mods:mods ID="MODS" version="3.0" >
...
    <mods:relatedItem type="constituent" ID="DMD_disc01_tr001">
            <mods:titleInfo>
                        <mods:title>Allegro Maestoso</mods:title>
            </mods:titleInfo>
            <mods:physicalDescription>
                        <mods:extent>15:51</mods:extent>
            </mods:physicalDescription>
    </mods:relatedItem>
</mods:mods>
```

Profile Aware Validation Tool

- Implemented with pipelined XSLT’s in Apache Cocoon



Sample Validation Report



Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:8888/MetsProfiles/report/cdProfile/cd.xml/
schematron
Search Web
Bookmarks
My Yahoo!
Finance
Address Book
Yellow Pages
http://localhost:8.../cdProfile/cd.xml/
http://localhost:8888/MetsProfiles/report/cdProfile/cd.xml/schematron-out.html#N10546
Adblock

Schematron Report

Generated Schematron Schema

StructMap Check

- Expected a minimum 1 occurences of linked metadata of type mods:relatedItem at ID .
- Expected a minimum 1 occurences of linked metadata of type mods:titleInfo at ID .

```
        <mets:div TYPE="cd:compactDiscObject" DMDID="MODS1">
                <mets:div TYPE="cd:disc">
                        <mets:div TYPE="cd:track"/>
                </mets:div>

                <mets:div TYPE="cd:disc">
                        <mets:div DMDID="DMD_disc01_tr001" ID="disc01_tr001" TYPE="cd:track">
                                <mets:div TYPE="cd:audio">
                                        <mets:fptr FILEID="FN10081"/>
                                        <mets:fptr FILEID="FN102FD"/>
                                        <mets:fptr FILEID="FN10302"/>
                                </mets:div>
                        </mets:div>
                        <mets:div DMDID="DMD_disc01_tr002" ID="disc01_tr002" TYPE="cd:track">
                                <mets:div TYPE="cd:audio">
                                        <mets:fptr FILEID="FN10090"/>
                                        <mets:fptr FILEID="FN10311"/>
                                        <mets:fptr FILEID="FN10316"/>
                                </mets:div>
```

Profile Aware Display

- Common vocabulary of div TYPE’s
- Reuse UI “components”
  - loc:page

Repository Data Models

- div TYPE’s correlate with classes in an object oriented repository architecture
  - FEDORA
- Ingestion
  - Atomization into AIP’s based off of div TYPE’s from SIP’s
  - <mets:div TYPE=“loc:image”/> becomes a separate object

National Digital Newspaper Project

- Goal is to have partner institutions submit METS based SIP’s
  - Newspapers
  - Issues
  - Reels
- METS Profiles serve as contracts and enable validation of SIP’s before ingestion
- div TYPE facilitate atomization during the ingestion process to a FEDORA repository
  - Model Objects: Newspapers, Reels, Issues, Pages
  - Content Objects: Image, Text

Future Thoughts

- Syntax
  - RelaxNG like syntax for cardinality
- Documenting type vocabularies which can be used across profiles
- METS Specific
  - Is there a generic way to solve this across a broader set of digital object implementations
- Best way to get others involved?
  - Wiki