Google Summer of Code 2009 Fedora Integration

Title: DSpace2 storage-fedora module implementation (Initially: Fedora DAO implementation for DSpace, beta release)

Student: Andrius Blažinskas

Mentor: Richard Rodgers

About

Project DSpace2 storage-fedora module implementation is a storage module allowing DSpace store its data to Fedora repository. Targeted versions are DSpace 2.x and Fedora 3.x (during development Fedora 3.2.1 was used).

After discussion with community members, it was decided to abandon GSOC2008 work on DSpace 1.x (DSpace & Fedora Integration) and continue this work on DSpace 2.x. The data model in DSpace 2.x is different so mapping part was remade. The same way code heavily reorganized to reflect changes and to prepare it as DSpace 2 module.

Development plan/progress

- In-depth analysis of DSpace 2 data model and the possibilities of mapping it with Fedora 3 model. (Done)
- DSpace & Fedora model mapping design: basic mapping. (Done, but mapping will evolve)
- Mapping implementation (Done, however some minor fixes are needed).
    - StorageVersionable implementation for Fedora3 (on TODO list)
- Creation of tests (Done, however some extensions are being created)
- Creation of documentation (Done)

DSpace 2 data model

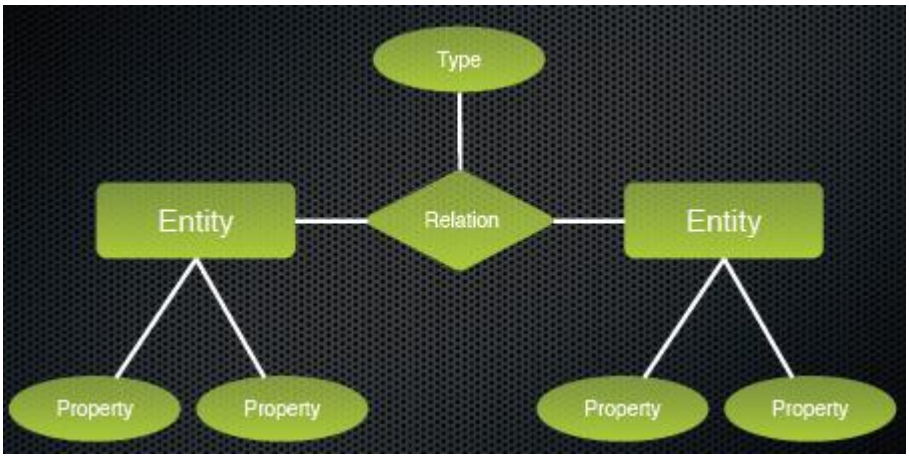

Figure 1: General DSpace 2 data model (http://smartech.gatech.edu/dspace/bitstream/1853/28078/5/214-578-1-PB.pdf)

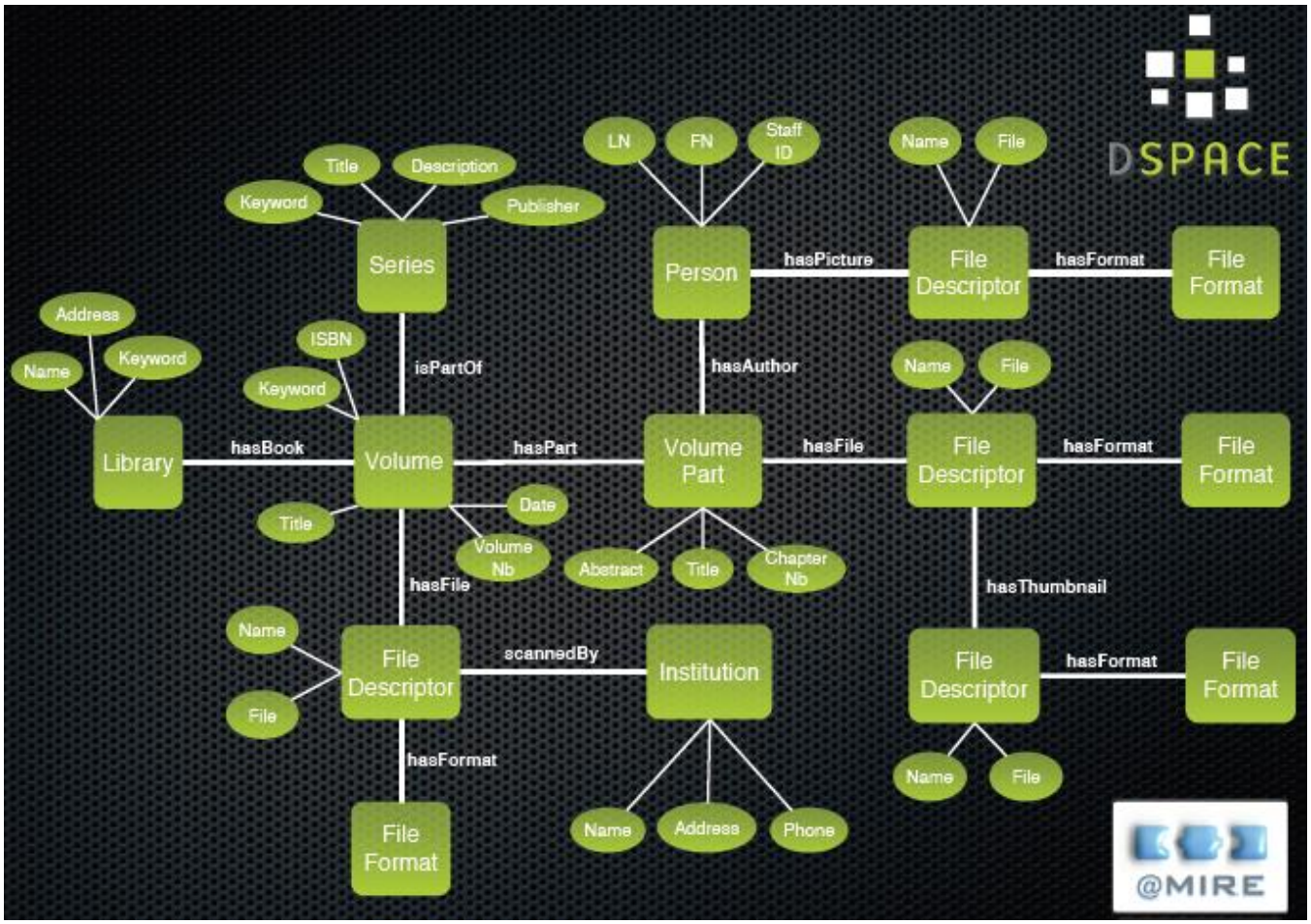

Figure 2: Example DSpace 2 data model implementation (http://smartech.gatech.edu/dspace/bitstream/1853/28078/5/214-578-1-PB.pdf)

Model mapping

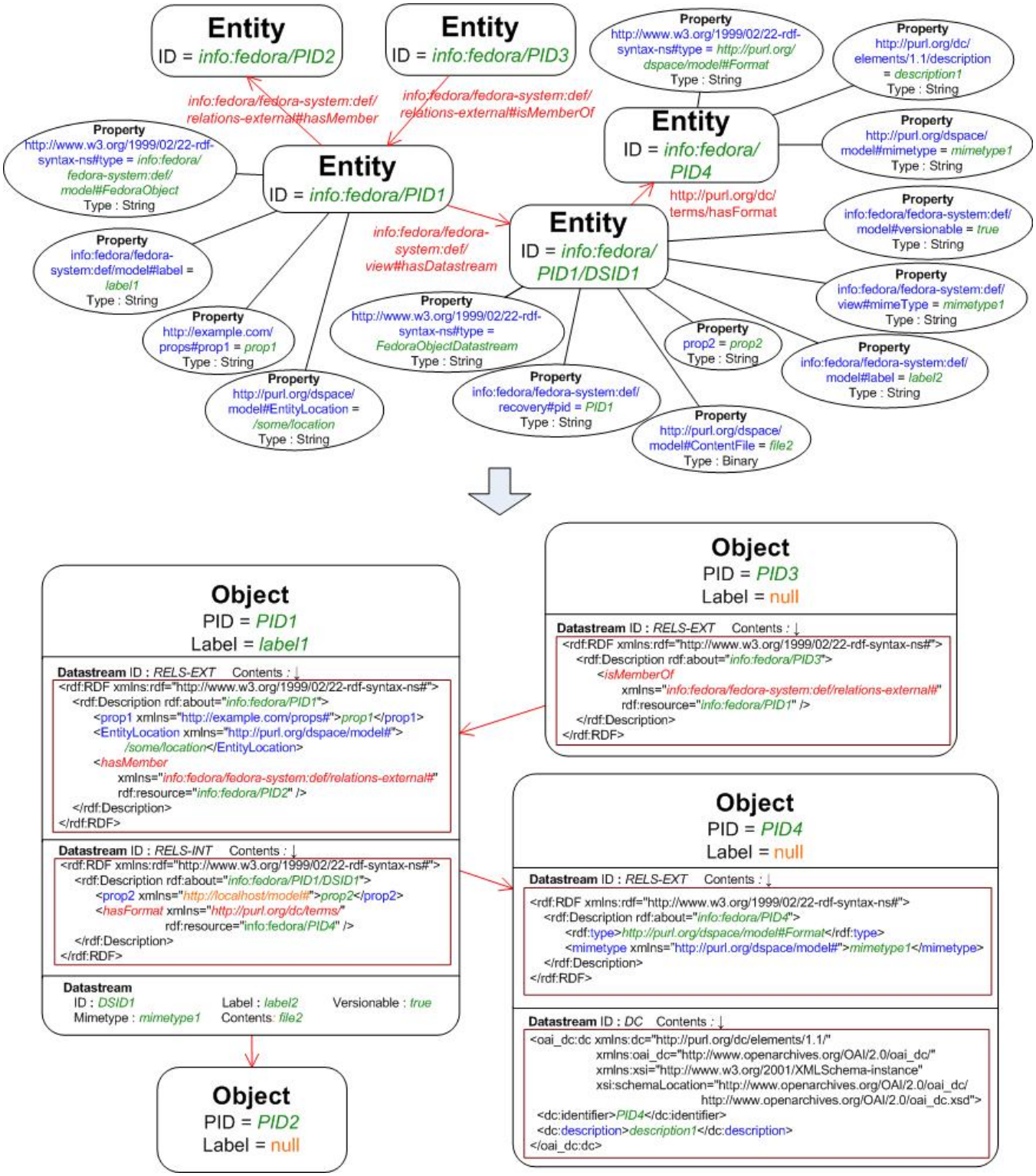

Figure 3: Proposed model mapping

Mapping notes:

- Entity type is identified using general predicate http://www.w3.org/1999/02/22-rdf-syntax-ns#type. For now, literal FedoraObjectDatastream used to indicate mapping to datastream.
- Any binary (file) properties are unmapped, unless they are located in FedoraObjectDatastream entity and has name http://purl.org/dspace/model#ContentFile. Only one such property allowed per FedoraObjectDatastream entity.
- In diagram, relations between objects indicated using info:fedora/fedora-system:def/relations-external#hasMember/isMemberOf predicates, however other custom predicates also possible and will be literally transferred if provided.
- Datastream dependence to particular Fedora object must be indicated using info:fedora/fedora-system:def/view#hasDatastream predicate. Such relations between FedoraObjectDatastream entities are not allowed.
- String properties provided without namespace are assigned default http://local/properties# namespace.

- Any property starting with http://purl.org/dc/elements will end up in DC datastream.
- Datastream info:fedora/fedora-system:def/view#mimeType and Format entity http://purl.org/dspace/model#mimetype are managed separately, however they should be the same.
- Fedora object label indicated using info:fedora/fedora-system:def/model#label and datastream label (for now) - http://www.w3.org/2000/01/rdf-schema#label.
- Easy notable in DSpace2 code, however no direct alternative in Fedora having entity location, will be put in RELS-EXT as separate http://purl.org/dspace/model#EntityLocation (yet "invented") metadata field.

Other potentially useful Fedora predicates to be implemented:

- info:fedora/fedora-system:def/view#lastModifiedDate - to retrieve object modification date
- info:fedora/fedora-system:def/view#version - to retrieve datastream version, as versioning to be enabled
- info:fedora/fedora-system:def/view#disseminates and #disseminationType - to define more advanced dissemination services?
- info:fedora/fedora-system:def/model#ownerId - set/get owner
- info:fedora/fedora-system:def/model#altIds - set/get alternate ids
- info:fedora/fedora-system:def/model#digest and #digestType - set?/get digest
- info:fedora/fedora-system:def/model#state - manage state (info:fedora/fedora-system:def/model#Active / #Inactive / #Deleted)
- info:fedora/fedora-system:def/model#createdDate - to retrieve creation date
- info:fedora/fedora-system:def/model#contentModel - defining more specific content model?
- info:fedora/fedora-system:def/model#length - length?
- Others?...

Entities

DSpace 2 data model entities "marked" with property http://www.w3.org/1999/02/22-rdf-syntax-ns#type = info:fedora/fedora-system:def/model#FedoraObject are mapped to Fedora objects. Entities having property http://www.w3.org/1999/02/22-rdf-syntax-ns#type = FedoraObjectDatastream are indirectly mapped (binary property has direct datastream mapping) to Fedora objects datastreams. Entities having no #type property, by default are mapped to Fedora objects. Datastream dependence to object is indicated using info:fedora/fedora-system:def/recovery#pid property.
All necessary administrative Fedora object and datastream properties are taken from corresponding entity properties. If multiple properties with same name exist and only one is needed - first one is taken.

Properties

Properties of DSpace 2 entities are mapped to Fedora RELS-EXT, RELS-INT, DC datastream entries and separate datastreams. If property has name http://purl.org/dspace/model#ContentFile, is binary type (InputStream java class) and is located in FedoraObjectDatastream entity, then it will directly result as a datastream. Only one http://purl.org/dspace/model#ContentFile property is allowed per FedoraObjectDatastream entity. Any string property starting with http://purl.org/dc/elements or http://www.openarchives.org/OAI/2.0/oai_dc/ will end up in DC datastream. Any other non DC and non administrative (administravite starts with info:fedora) string property will go into RELS-EXT for FedoraObject entities and RELS-INT for FedoraObjectDatastream entities. String properties can be freely defined by user which may not provide namespace, so in such cases "local" namespace http://localhost/model# will be forced.

Relations

Relations between DSpace2 FedoraObject entities are directly mapped to Fedora relations between objects, which in turn are put in RELS-EXT datastream. Relations pointing from datastreams are defined in RELS-INT. In diagram, relation info:fedora/fedora-system:def/relations-external#hasDatastream has no direct mapping and currently does not participate in any way. Using current mapping, DSpace2 relations in Fedora generally can result in any combination: object-to-object, object-to-other-object-datastream (in RELS-EXT); datastream-to-datastream, datastream-to-object (in RELS-INT), etc. While relations between datastreams in different objects may not be very correct, it is left up for the user to choose the resulting model implementation specifics including relation types.

Where are a lot of relations types defined out there, but in storage-fedora module they can also be freely defined by user. If namespace is not provided for particular relation type, local namespace http://localhost/model# will be forced.

Example of child objects RELS-EXT content fragments:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="info:fedora/dspace:Book~1">
    <locatedIn xmlns="http://localhost/model#" rdf:resource="info:fedora/dspace:Library~1"/>
 </rdf:Description>
</rdf:RDF>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="info:fedora/dspace:Book~2">
    <locatedIn xmlns="http://localhost/model#" rdf:resource="info:fedora/dspace:Library~1"/>
  </rdf:Description>
</rdf:RDF>
```

Example ITQL query for fast child selection (Fedora resource index must be turned on):

```
select $subject from <#ri>
where  $subject <http://localhost/model#locatedIn>
   <info:fedora/dspace:Library~1>
```

Example CSV response to it:

```
"subject"
info:fedora/dspace:Book~1
info:fedora/dspace:Book~2
```

When designing DSpace2 model implementation, designer (user) should also keep in mind, that entities relations pointing from parent to child can be inefficient, since parent entities usually tend to have a lot of child entities (consider the example of parent Library and child Book above). If parent references all of its children, parent Fedora object will possibly have large rapidly changing and growing number of RELS-EXT entries. This problem does not arise in child to parent referencing.

Identifiers

It is very likely, that organizations using Fedora, may prefer using their custom Fedora objects PIDs and DSIDs (datastream IDs), so implemented storage-fedora module does allow this functionality. User himself must ensure uniqueness of custom identifiers. DSpace entity identifier must have form of **info:fedora/PID** for objects and **info:fedora/PID/DSID** for datastreams, so that it can be interpreted correctly by storage-fedora module. Incorrect entity identifier (incompatible with Fedora resource URI) will result in error. If Fedora object or datastream identifier in not provided - one will be generated automatically.

Fedora PID namespace, used for automatic PID generation, is configurable and predefined in storage-fedora module configuration file.

Concerned about having pids contain any semantic meaning, discussions to date concerned having pids always be opaque to the application, the best example to support this would be the usage of uuids or fedora ids out of the box. please be cautious about the proposed usage above. Use of other properties will be more appropriate to determine the object type from (rdf:type or dc:type for instance). --Mark Diggory 22:37, 12 July 2009 (EDT) |

Identifiers having form <namespace>:<Entity name>~<UUID> and <namespace>:<UUID> were decided not to be used, thus removed from wiki. Though UUIDs are quite attractive and possibly will have more attention in future. --Andrius Blažinskas 00:46, 30 July 2009 (GMT+2) |

Versioning

Datastream versioning is important feature in Fedora what DSpace 2 could take advantage of. Fedora can version all datastreams, so basically both - binary files and RELS-EXT & RELS-INT (DSpace metadata and relations) can be versioned. The problem here is that a lot of time scattered changes in one datastream will result in lot of its copies, because Fedora simply keeps every changed version. This can be complicated when datastreams are relatively big and change rapidly.

Work on versioning for storage-fedora currently is in progress.

Where if REL-EXT supports versioning, then the majority of encoded DSpace metadata and relationships would be versioned as a unit for each DSpace Object. --Mark Diggory 22:41, 12 July 2009 (EDT) |

Implementation details

storage-fedora module is implemented in similar way storage-jackrabbit is. Currently module implements org.dspace.providers.StorageProvider, org.dspace.services.mixins. StorageWriteable/StorageVersionable and org.dspace.kernel.mixins.ShutdownService.
Most recent code of storage-fedora will be available at http://scm.dspace.org/svn/repo/modules/storage-fedora/.

Comments

DSpace+2.0 Developer Recommendations

We propose using RELS-EXT to store the majority of DSpace Properties and Relations for a DSpace+2.0 Entity. The Goal we are hope to see attained is to have DSpace 2.0 act as a Management Toll on exisitng Fedora Repository Content that may have not come from DSpace in the first place, this means

1. No DSpace centric metadata formats stored in separate bitstreams
2. Use of RELS-EXT for all relations in DSpace+2.0
3. Use of dc metadata datastream for any Dublic Core Elements

4. Use of RELS-EXT for any other metadata properties
5. Use of RELS-INT to identify relationships that are data files

Consider that there are efforts to map Fedora to JCR and we should consider these in the approriate mappings to DSPace 2.0 / JCR and Fedora (I will try to add more detail on this shortly) --Mark Diggory 16:16, 12 July 2009 (EDT)

"Caution against the use of the following expressed namespace "http://purl.org/dspace2/model/relations/local" the relations already have their own namespace appropriate (FoaF, ORE, DCMI, etc). The only place that a "dspace" specific namespace will probably be employed in DSpace+2.0 is to capture cases where legacy DSpace data model cannot be mapped explicitly to an already existing ontology from one of the various communities. --Mark Diggory 22:35, 12 July 2009 (EDT)

References

DSpace2 model and demo by Ben Bosman: http://smartech.gatech.edu/dspace/handle/1853/28078, http://presentations.dlpe.gatech.edu/or09/or09_052009_3/index.html

DSpace2 RDF: http://wiki.dspace.org/index.php/DSpace+2.0/Expressing_DSpace_Domain_Model_In_RDF

JCR for Fedora mappings: http://jcr-connect.at.northwestern.edu/en/JCR_for_Fedora_-_Discussion

Project code is available at: http://scm.dspace.org/svn/repo/modules/storage-fedora