Proposal For Metadata Enhancement

Primary Objective

The primary objective of this proposal is that the DSpace metadata registry be "naturally" extended to support a richer and more expressive "Metadata Schema". Technical Objectives of the proposal are to provide the following features:

1. Capability to Define Metadata "Profiles" that may be assigned to individual DSpace Objects
2. Capability to Define "subPropertyOf" relationships in place of the legacy ns.element.qualifier approach for more expressive inheritance and mapping to OAI_DC
3. Capability to have "immutable" DC, DCTERMS and other "well established" namespaces to treat as sources for "subPropertyOf" assignments.
4. Capability to to Restrict and Validate Existing DSpace Object Metadata based on the assigned "Profile".
5. Capability to Apply these profiles similarly to any DSO; Communities, Collections, Items, Bundles and Bitstreams, even Groups and EPeople.

Expectations for Backward Compatibility

This proposal is based on the premis that changes to DSpace metadata characteristics must be backward comparable and retain the same functionality as previously existed to ease transition for all existing users of the platform. So many different functional areas of DSpace are reliant on existing metadata functionality that it is criticial that any changes in functionality also have well defined and scripted updates across releases. Thus another very critical feature of this proposal is that this new Schema model should support the above features without significant need to transform existing DSpace Item metadata nor the registry itself.

Conceptual Definition of "Schema"

The DSpace MetadataSchema registry was designed based on an outdated concept of "Application Profiles" and "Qualified Dublin Core" that predated the current DCMI Abstract Model. Due to this, there are number of significant shortcomings to the current implementation.

1. Namespaces are not really "Schema"
2. Schema may be validated, however, there are no actual rules in DSpace "MetadataSchema" or "MetadataField" data models.
3. Qualification does not effectively meet needs for use of alternative namespaces nor support any ability for programatic mapping to DC for exposing metadata in other namespaces within in OAI_DC.
4. The Schema and Fields defined are insufficient to support attributes and rules for validation of DSpace metadata fields in relation to Item Submission or other methods of Deposit.

The current "DSpace Schema" does not meet the requirements that a Schema is traditionally used for. Schema are traditionally used to define a scaffolding or framework of rules which actual content can be validated against. While the current MetadataSchema/Field does restrict what can be assigned to any item in DSpace, it does not provide any support for validation of these assignments, nor allow us to further define the encoding of the metadata values nor if they are required or not. At this time, much if of the validation, rules and encoding is poorly assigned instead, at the UI/Presentation level in the DSpace Submission input-forms.xml file and only enforced in the Describe Step of the Submission workflow.

This proposal seeks to extend the definition of the DSpace Metadata Schema to include support of these features previously found only in the Submission input-forms.xml. Formaizing a strategy for metadata validation in DSpace that is a new core feature.

Content Models For DSpace (Extending on MetadataSchema and MetadataField to provide "Metadata Profiles")

Rather than the current MetadataSchema applying to the namespace of the metadata fields that are allowed by the entire repository. It is instead recommended that this table be repurposed and expanded to support creation of "Named Profiles" that can be easily assigned to DSpaceObjects as Metadata or or Content Models. In this case, typing would initially be based on:

- DSpace Object Types (Site, Community, Collection, Item, Bundle, Bitstream)
- DCMI or Other Classes (Collection , Dataset , Event , Image , InteractiveResource , MovingImage , PhysicalObject , Service , Software , Sound , StillImage , Text)
- Custom Local Types, of which the existing Qualified Dublin Core schema will be initial considered one of.

These above types will be expressed through the addition of properties to the MetadataSchemaRegistry and MetadataFieldRegistry tables to provide the facility to expand on and add additional Schema. Some Hypothetical examples of such schema would be:

Community or Collection Profiles	**Description**
Document Collection Profile	Metadata Fields Appropriate to Describe Details of a Generic Document Collection for the purposes of a Finding Aid
Journal Issue Profile	Metadata Fields Appropriate to Describe Details of a Journal or Journal Issue for the purposes of a Finding Aid
Image Gallery Profile	Metadata Fields Appropriate to Describe Details of an Image or Multimedia Collection, allowing UI Hooks that are beneficial for Multimedia (Slide Decks, Light tables, Viewers, etc)
Etc,	...

Item Profiles	Description
Scholarly Item	Metadata Fields Appropriate to Describe a Scholarly Research Article
Website Item	Metadata Fields Appropriate to Describe a number of individual Bitstream files that constitute a website.
Thesis Item	Metadata Fields Appropriate to Describe a Dissertation of Thesis Item based on conventional ETDMS terminology
Technical Report Item	
Journal Article	
Learning Object Item	
Etc,	...

Bitstream Profiles	Description
Streaming Video Profile	Metadata Fields Appropriate to Describe a moving picture or video
Image Profile	Metadata Fields Appropriate to Describe an individual image
Document	Metadata Fields Appropriate to Describe an individual document
Spreadsheet	Metadata Fields Appropriate to Describe a Data file
Etc,	...

Likewise, the above profiles could be applied heterogeniously though metadata attached to any level of the DSpace object hierarchy.

Metadata Field Inheritance

Individual Metadata Fields, like DCMI metadata properties will support subTyping or inheritance. For example, from the DCMI Website, we have the following:

http://dublincore.org/documents/dcmi-terms/#terms-title

Term Name: title	
URI:	http://purl.org/dc/terms/title
Label:	Title
Definition:	A name given to the resource.
Type of Term:	Property
Refines:	http://purl.org/dc/elements/1.1/title
Version:	http://dublincore.org/usage/terms/history/#titleT-002
Has Range:	http://www.w3.org/2000/01/rdf-schema#Literal

Supporting a similar level of refinement for DSpace Metadata can be supported through the addition of new MetadataFieldRegistry properties that are capable of storing this relationship.

Data Model Changes to Support This Proposal

To support this prpoposal, only additional fields and relational tables will be required to be added to the existing DSpace schema.

MetadataProfile:

Profile will be used to identify a set of MetadataFieldProfile that define the fields allowed on a DSpace Object.

MetadataFieldProfile

Individul field profile used to identify the basic rules allowed for a field assigned to a DSpace Object.

Profile2dso:

This table will be utilized to directly map any specified schema as a validation target for any existing DSpace Item. One, or more than one Schema assignment will be allow, creating a situation where an Item may be polymorphic and support more than one type.

Profile2container:

This table will support the identification of which profile should be applied to new Items being created in any Collection within DSpace. This will be extended when support for metadata at all levels of DSpace is introduced, allowing assignment of Collection and Community "Types" to Community containers and likewise, support for Specific Bitstream types to be allowed in Item Containers.

A tentative list of new fields and tables is exemplified in the class diagram below.

Gliffy Macro Error

An error occurred while rendering this diagram. Please contact your administrator.

- **Name:** New Metadata Class Diagram

The above solution can be easily encoded into the database schema, while the existing MetadataSchema, MetadataField and MetadataValue objects should be easy extendable to support new methods and business logic.

A Example Use-Case

Metadata Schema Registry

In the following example an additional "dcterm" schema has been created to house the proper dcterms predicates while the "dc" schema continues to hold the existing qualified dc for legacy purposes.

	ID	Namespace	Name
	1	http://purl.org/dc/elements/1.1/	dc
	2	http://purl.org/dc/terms	dcterms

Metadata Schema: "dcterms"

where "dcterms:xxx" refinements point to a new Schema in the repository that contains the fields required for the typical dcterms namespace. In the current case, with the "item" and "item2" schema, this schema is not applied directly to Items, but inherited into defined "item" fields through "refinement".

ID	Field	refines	encoding	default	required	Scope Note
15	dcterms.date	rdf:Property	W3CDTF	${now}	true	Date of publication or distribution.
25	dcterms.identifier	rdf:Property	URI		true	Uniform Resource Identifier
37	dcterms.language	rdf:Property	RFC5646	en		Catch-all for non-ISO forms of the language of the item, accommodating harvested values.
40	dcterms.relation	rdf:Property	URI			Catch-all for references to other related items.
57	dcterms.subject	rdf:Property	Literal			Uncontrolled index term.
64	dcterms.title	rdf:Property	Literal		true	Title statement/title proper.
66	dcterms.type	rdf:Property	Class			Nature or genre of content.
...	...	...	...	...	...	...

Metadata Profile Registry

The profile registry defines fields that may be attached to a DSpace Item.

- A new "Profile" has been defined with its own namespace to be allowed on Collections A and B.
- Each custom "Profile" can be applied to a specific DSO type (in this case, Item) via an "Applies To" mapping to objects that are of its type (in the diagram above, this is the profile2dso mapping).
- Each custom "Profile" can enabled in a specific Container (Community, Collection, Item) via an "Allowed In" mapping (in the diagram above, this is the profile2container mapping).

	ID	Namespace	Name	Applies To	Allowed In
	1	http://mydspace/schema/item	Generic Item	Item	All Collections
	2	http://mydspace/schema/item2	Simple Item	Item	Collection A, Collection B

Item Metadata Profile "Generic Item"

The following exemplifies how a Profile for generic items that may have many optional fields attached to them.

element	refines	encoding	default	required	Scope Note
issued	dcterms:issued	W3CDTF	${now}	true	Date of publication or distribution.
date	dcterms:date	W3CDTF	${now}		Use qualified form if possible.
uri	dcterms:identifier	URI		true	Uniform Resource Identifier
identifier	dcterms:identifier	Literal			Catch-all for unambiguous identifiers not defined by qualified form; use identifier.other for a known identifier common to a local collection instead of unqualified form.
iso	dcterms:language	RFC5646	en		Current ISO standard for language of intellectual content, including country codes (e.g. "en_US").
language	dcterms:language	RFC5646	en		Catch-all for non-ISO forms of the language of the item, accommodating harvested values.
haspart	dcterms:relation	URI			References physically or logically contained item.
relation	dcterms:relation	URI			Catch-all for references to other related items.
mesh	dcterms:subject	URI			MEdical Subject Headings
other	dcterms:subject	Literal			Local controlled vocabulary; global vocabularies will receive specific qualifier.
subject	dcterms:subject	Literal			Uncontrolled index term.
alternative	dcterms:title	Literal			Varying (or substitute) form of title proper appearing in item, e.g. abbreviation or translation
title	dcterms:title	Literal		true	Title statement/title proper.
type	dcterms:type	Class			Nature or genre of content.
...	...	...	...	...	...

Item Metadata Profile "Simple Item"

The second Item profile exemplifies a simple item with a smaller set of fields allowed, but with stricter requirements for populating those fields.

Field	refines	encoding	default	required	Scope Note
issued	dcterms:date	W3CDTF	${now}	true	Date of publication or distribution.
uri	dcterms:identifier	URI		true	Uniform Resource Identifier
language	dcterms:language	RFC5646	en	true	Catch-all for non-ISO forms of the language of the item, accommodating harvested values.
mesh	dcterms:subject	URI		true	MEdical Subject Headings
title	dcterms:title	Literal		true	Title statement/title proper.
type	dcterms:type	Class		true	Nature or genre of content.

Steps To getting There

- New fields and tables are added to database.
- New attributes for existing DC schema and addition of the DCTerms Schema should be added to Registry after it has been extended.
- Creation of several "Item Profiles" that can exemplify different types of Items in DSpace. each should utilize the new DCTERMS Schema where-ever possible.
- Update DSpace build process to populate any necessary fields in new MetadataField and Profile tables.

- Improve User interface and DSO data model to include returning details pertaining to Profile types for informing the User interface
- Creation of new Describe Step and ItemEdit interfaces that enforce validation requirements expressed in the Metadata Profile
- Creation of MetadataProfile Administrative Interfaces for managing Profiles.

Summary

The above proposal clarifies that new capabilities may emerge for "Typing" , "Restriction" and "Validation" of DSpace objects through extension of the existing data model. The proposed strategy will support stronger typing of not only DSpaceObejcts, but also the values of metadata fields through validation rules such as syntax or vocabulary encodings, requiredness, Dublin Core or other metadata schema types. DSpace should be able to utilize the new MetadataProfileRegistry as a means to replace large portions of the functionally found in the input-forms.xml file in future DSpace versions.