~~THE NSDL~~ A METADATA REGISTRY

9/25/2008

DC2008 May 2008

What is it?

A (fairly) complete multi-user environment for the managed development of value vocabularies, metadata schemas, and application profiles.

Who is it?

- An NSF-funded project of the National Science Digital Library
- Started in 2005 with a team of 2 PIs (Stuart Sutton & Diane Hillmann) and 2 developers (Jon Phipps & Ryan Laundry)
- Project extended in 2007 with a smaller team – Jon (PI) & Diane

What's under the hood?

- PHP 5
- MySQL 5
- Symfony PHP framework (RoR-like)
- ARC RDF library
- Selenium/PHPunit for functional tests

One of the principal goals was to make it easy to deploy

Do I need a 'registry'?

If you …

- Have a permanent domain for namespace management
- Can setup and maintain content negotiation (or don't care)
- Have only one contributor, or can manage user access
- Can track changes & versions over time (or don't care)
- Have a simple way to move from development to published
- Don't need to manage html, rdf, and xml encodings
- Have a way to notify dependent systems of changes
- Don't care about participating in a broader community

… then you don't!

So, what does it do?

- Provides a web of trust by managing **access and editing rights** for groups of vocabulary maintainers maintaining individual vocabularies
- **namespace management and maintenance** services providing permanent URIs
- **content negotiation for retrieval of registered vocabularies** in various formats
  currently: RDF/XML (rdf), XHTML (html), and XML Schema (xsd)
- controlled **concept editing and maintenance using SKOS**
- controlled **mapping of relationships between concepts in different vocabularies.**
- maintains property-level **change history**

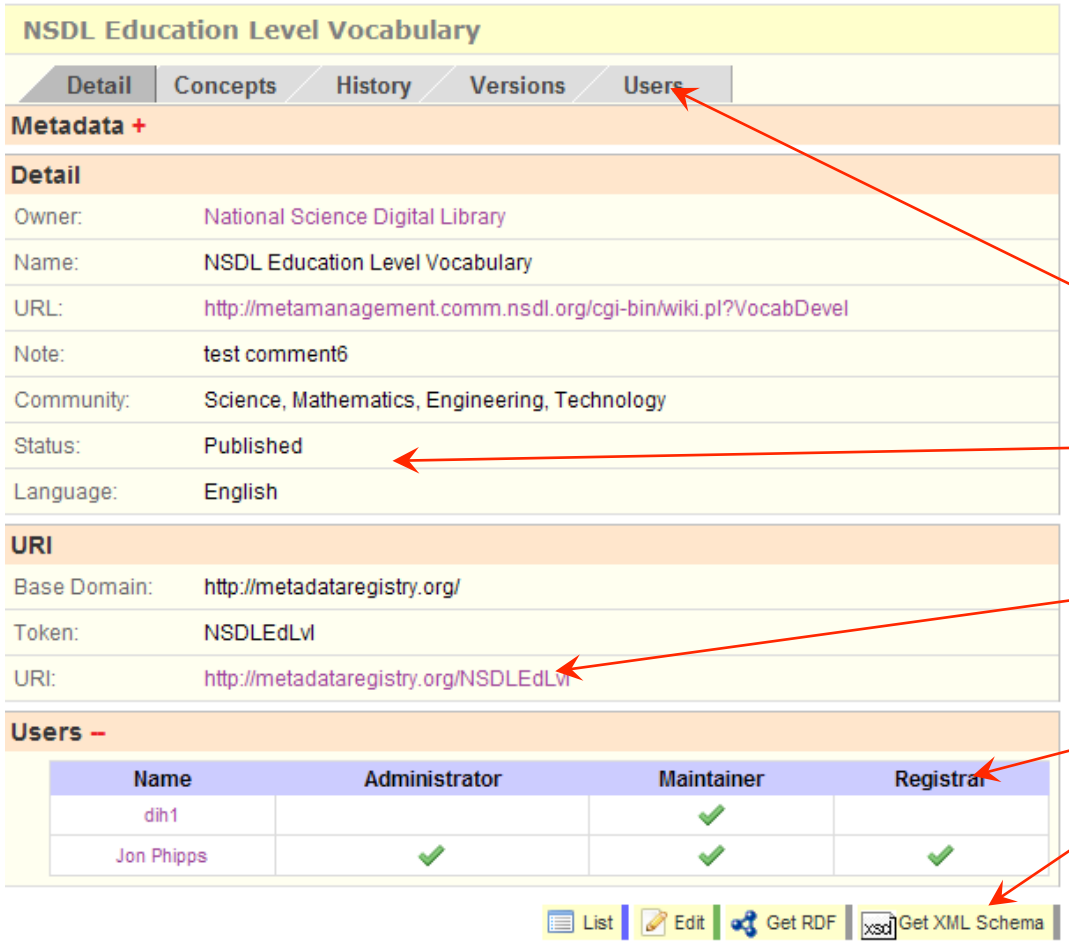
But wait! There's more!

- **search and browse** for concepts by label
- **multilingual vocabulary maintenance**
- vocabulary- and concept-level **version management** (in alpha)
- **import and management of existing vocabularies,** with and without existing URIs (RSN)
- registered users can receive **notifications of changes** to vocabularies to which they have subscribed (RSN)
- **content negotiation** and resolution services for registered vocabularies **with URIs in non-registry namespaces (RSN)**
- metadata schema editing and maintenance
- Dublin Core application profile editing and maintenance (built-in DC Abstract Model compliance)

Web of trust

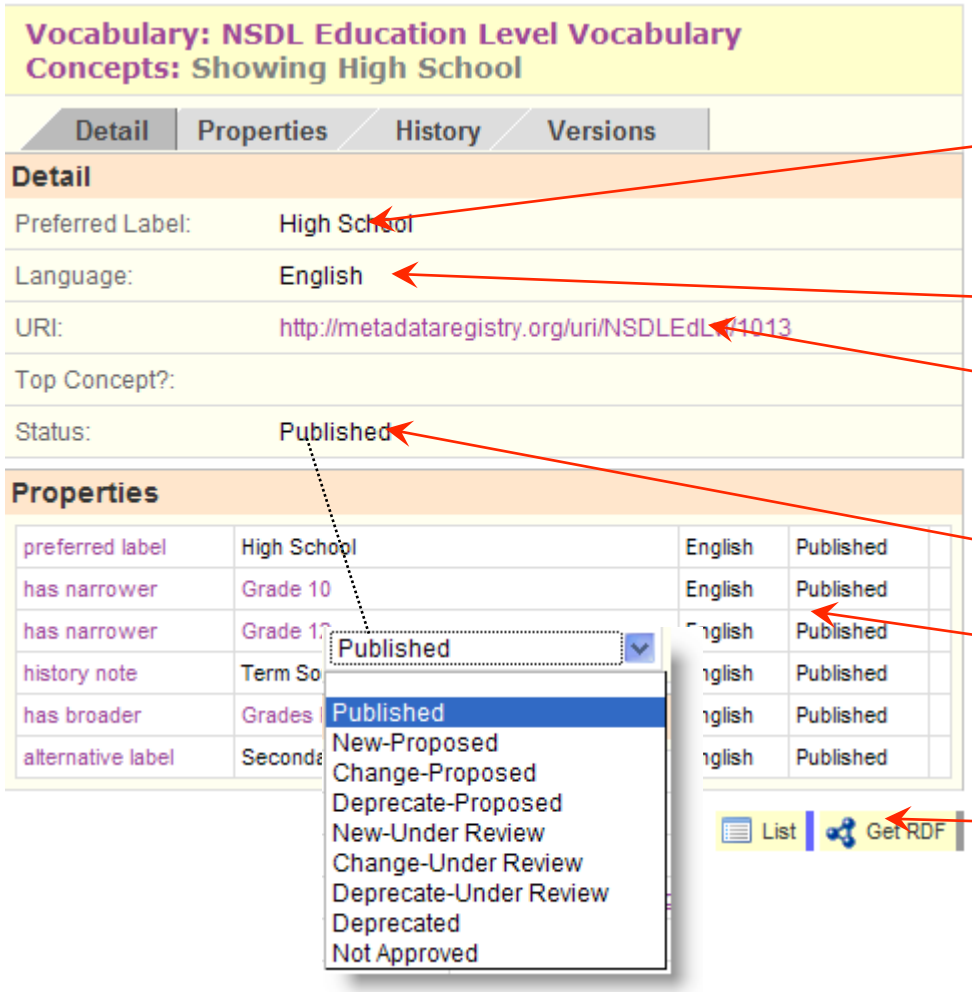
- The Registry is not a completely open system
- Users must register themselves in order to add/edit entities (Owners, Vocabularies, Schemas, APs)
- Once registered, a user must register an Owner entity that will 'own' the other entities
- When an Owner entity has been registered, only then can the other entities it owns be registered
- The user registering an Owner can add other users to that Owner's community, giving them the right to add/edit other entities
- The user registering an entity can add other 'maintainers' to that entity

Concept Scheme registration



- Description of entire vocabulary as a collection of concepts
- Links to Concepts, History, Versions
- Default status and Language for Concepts
- URI for the vocabulary itself (includes token)
- Links to Users
- Links to XML schema and RDF encoding

Concept registration



- *Required* Preferred Label
- Multi-language support
- Permanent URI for the Concept
- Support for status
- Links to other Concept properties
- Link to RDF fragment

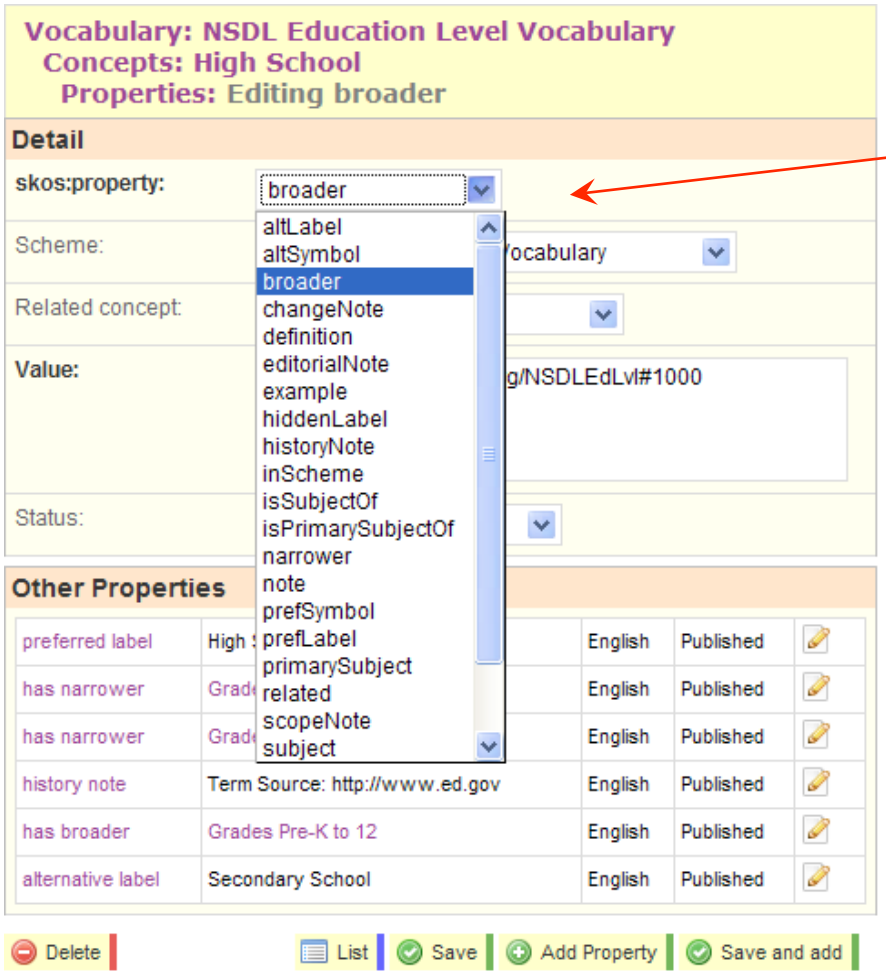
Wait, what happened to Terms?

- The Registry uses SKOS (Simple Knowledge Organization System) to describe vocabularies in RDF
- In SKOS, terms become labels for the more abstract Concepts that they represent
- The *term* 'scale' for example represents several *concepts*:
  - Type of skin (fish, snake)
  - Something that measures weight (bathroom, truck)
  - A parasite that eats my orchids
  - The ability of a system to handle increasing load
- Each of these is a *very* different concept, but each uses the same term as a label

Concepts and Concept Schemes

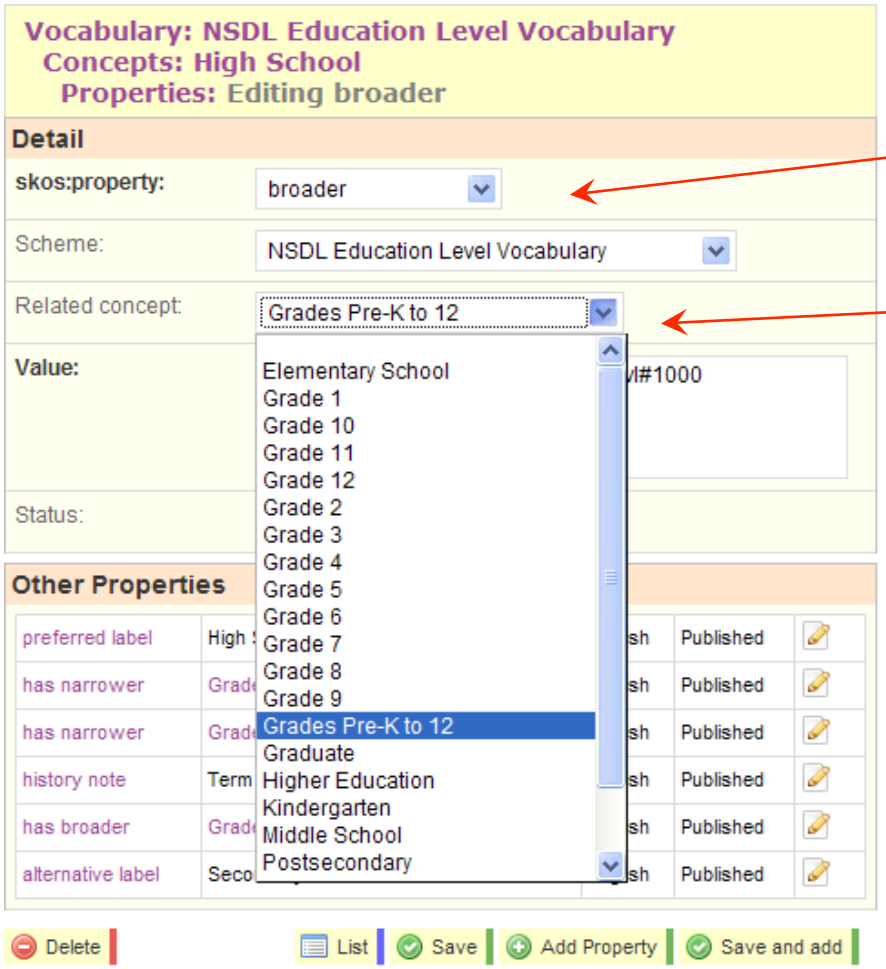
- The Registry assigns unique identifiers (URI) to Concepts (not terms)
- Terms are assigned to Concepts as either preferred labels (skos:prefLabel) or alternate labels (skos:altLabel) or 'hidden' labels (skos:hiddenLabel – usually used for misspellings)
- A Concept can have only one preferred label per language, but many alternates
- Concepts can be organized into Concept Schemes, roughly corresponding to a controlled vocabulary, term list, or taxonomy

Concept Property editing



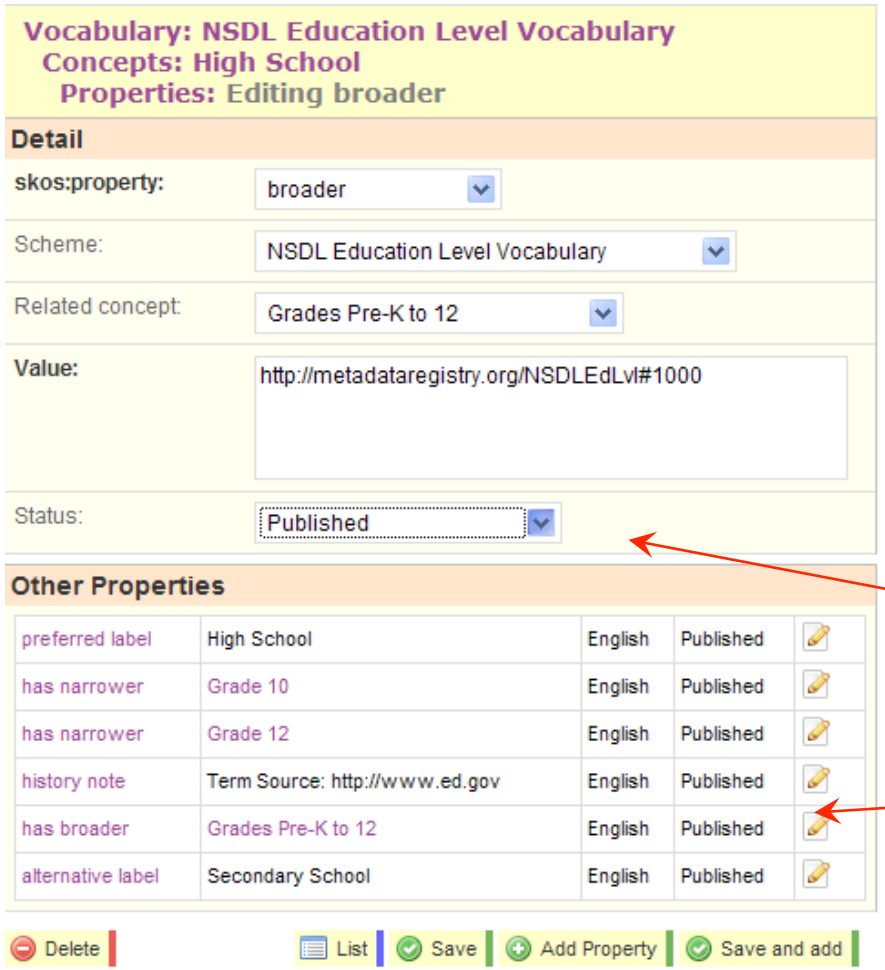
- Guided selection of Concept properties

Concept Property editing



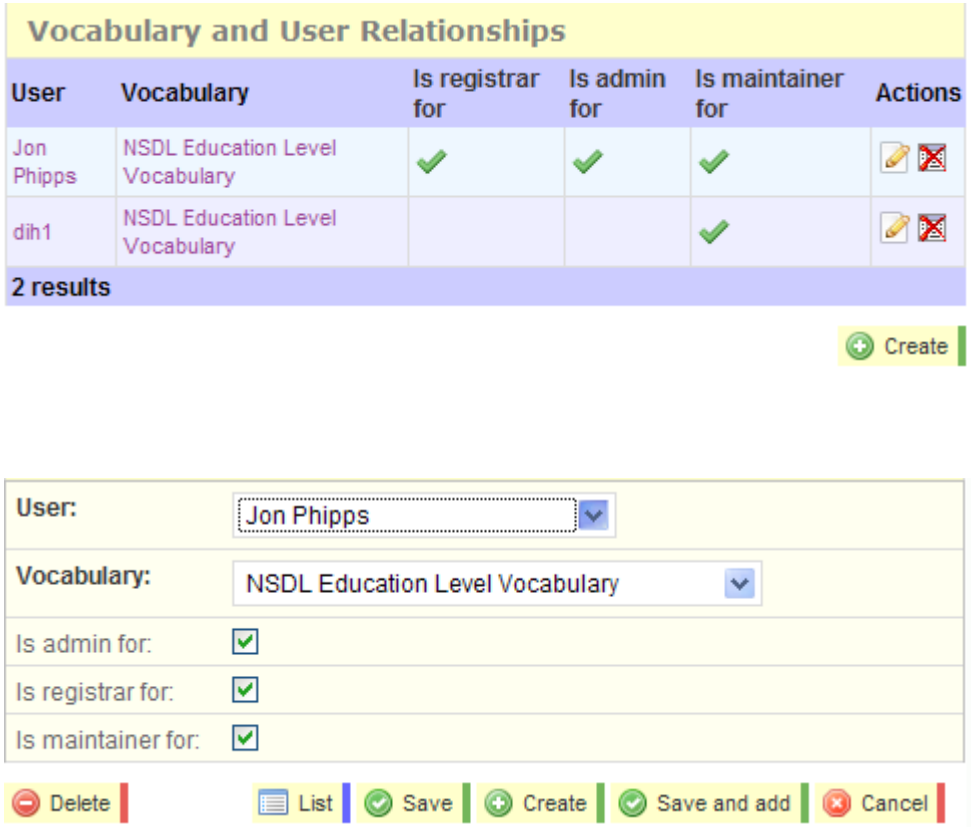
- Guided selection of Concept properties
- Guided selection of related concept when using a “relationship” property

Concept Property editing



- Guided selection of Concept properties
- Guided selection of related concept when using a “relationship” property
- Support for property-level status
- Links to other Concept properties

Vocabulary Maintainers



- Users and rights can be associated with specific Vocabularies
- Vocabulary admins can assign Users and editorial rights
- Any user who creates a new Vocabulary is automatically the admin for it

HTTP Publishing

- Implements http content negotiation (W3C 'Cookbook') based on W3C TAG httpRange-14 for document retrieval
  - See the Recipes http://www.w3.org/TR/swbp-vocab-pub/ and http://norman.walsh.name/2005/06/19/httpRange-14
- We use 'slash' (/) URIs instead of 'hash' (#)
  - Hash URIs must return the entire document when any fragment (identified by #) is requested
  - We think this is impractical for information resources
- You can still use hash URIs if you need to

HTTP Publishing

Also implements 'extension'-based document retrieval…

- A request for a document of the type 'application/rdf\+xml' located at… http://metadataregistry.org/uri/NSDLEdLvl
- redirects to… http://metadataregistry.org/uri/NSDLEdLvl.rdf
- Which returns…

```
- <rdf:RDF>
    <!-- Scheme: NSDL Education Level Vocabulary -->
  - <skos:ConceptScheme rdf:about="http://metadataregistry.org/uri/NSDLEdLvl">
      <dc:title>NSDL Education Level Vocabulary</dc:title>
      <skos:hasTopConcept rdf:resource="http://metadataregistry.org/uri/NSDLEdLvl/1000"/>
      <skos:hasTopConcept rdf:resource="http://metadataregistry.org/uri/NSDLEdLvl/1018"/>
    </skos:ConceptScheme>
    <!-- Concept: Grades Pre-K to 12  -->
  - <skos:Concept rdf:about="http://metadataregistry.org/uri/NSDLEdLvl/1000">
      <skos:inScheme rdf:resource="http://metadataregistry.org/uri/NSDLEdLvl"/>
      <skos:prefLabel>Grades Pre-K to 12</skos:prefLabel>
      <skos:narrower rdf:resource="http://metadataregistry.org/uri/NSDLEdLvl/1001"/>
      <skos:narrower rdf:resource="http://metadataregistry.org/uri/NSDLEdLvl/1013"/>
      <skos:narrower rdf:resource="URI"/>
    </skos:Concept>
```

Time-slices

- Time-slice versioning -- saves the state of the Scheme whenever a Concept property is added or changed
- RESTful interface -- scheme or Concept state can be retrieved by appending an encoded UTC Timestamp to URI…
  - scheme current trunk
    http://metadataregistry.org/NSDLEdLvl
  - scheme as of "2007-01-26T01:15:38.000Z"
    http://metadataregistry.org/NSDLEdLvl/ts/20070126011538000
- Only Concept properties that have 'Published' state
- Provides a permanent URI for Scheme/Concept at any given point in time -- helps minimize 'version churn'

Versions

- Named versions -- identifies an 'official' version by naming a time-slice
- RESTful interface -- Scheme or Concept state can be retrieved by appending a version to URI…
  - scheme as of "2007-01-26T01:15:38.000Z" http://metadataregistry.org/NSDLEdLvl/ts/20070126T011538000
  - permanently named "Release Version 1.2" http://metadataregistry.org/NSDLEdLvl/v/Release+Version+1.2
- Currently no support for editable branches

Metadata Schemas

- Schemas are the foundation for Application Profiles
- Schemas can import schemas from multiple namespaces (the following links are just demos)
- Editing a Schema: http://metadataregistry.org/schema.html
- Adding a namespace: http://metadataregistry.org/schema_namespace.html
- Editing a Schema property: http://metadataregistry.org/schema_property.html

(DC) Application Profiles

- Metadata Schemas are 'immutable' and need to be adapted in order to be 'applied' to a specific use.
- APs in the Registry are compliant with the DC Abstract Model (the following links are just demos)
- Editing an Application Profile: http://metadataregistry.org/profile.html
- Editing an Application Profile property: http://metadataregistry.org/profile_property.html

Where are we headed?

- Current development is expected to be pretty much completed by June 2009
- Integration with and support for the RDA effort
- Long-term (10+ years) support for Registry URI retrieval
- 80 vocabularies in the Registry ‘sandbox
- 17 official vocabularies in the Registry

SKOS Features

- Identifying concepts
  - URIs for unambiguous global identity and reference
- Labelling concepts
  - e.g. lexical labels, symbolic labels, preferred, alternative, hidden …
- Describing & documenting concepts
  - e.g. definition, example, scope note, change note, editorial note …
- Relating concepts
  - e.g. broader, narrower, related …
- Grouping concepts
  - e.g. concept schemes, support for 'node labels' …
- Subject Indexing
  - e.g. subject of a document, primary subject …
- Multilingual
- Extensible

Links

SKOS Homepage

http://www.w3.org/2004/02/skos/

SKOS Primer (2008 draft)

http://www.w3.org/TR/2008/WD-skos-primer-20080221/

SKOS Vocabulary Specification (draft)

http://www.w3.org/TR/2008/WD-skos-reference-20080125/

Dublin Core Abstract Model

http://dublincore.org/documents/abstract-model/

NSDL Metadata Registry
http://metadataregistry.org

Please play in our sandbox!
http://sandbox.metadataregistry.org

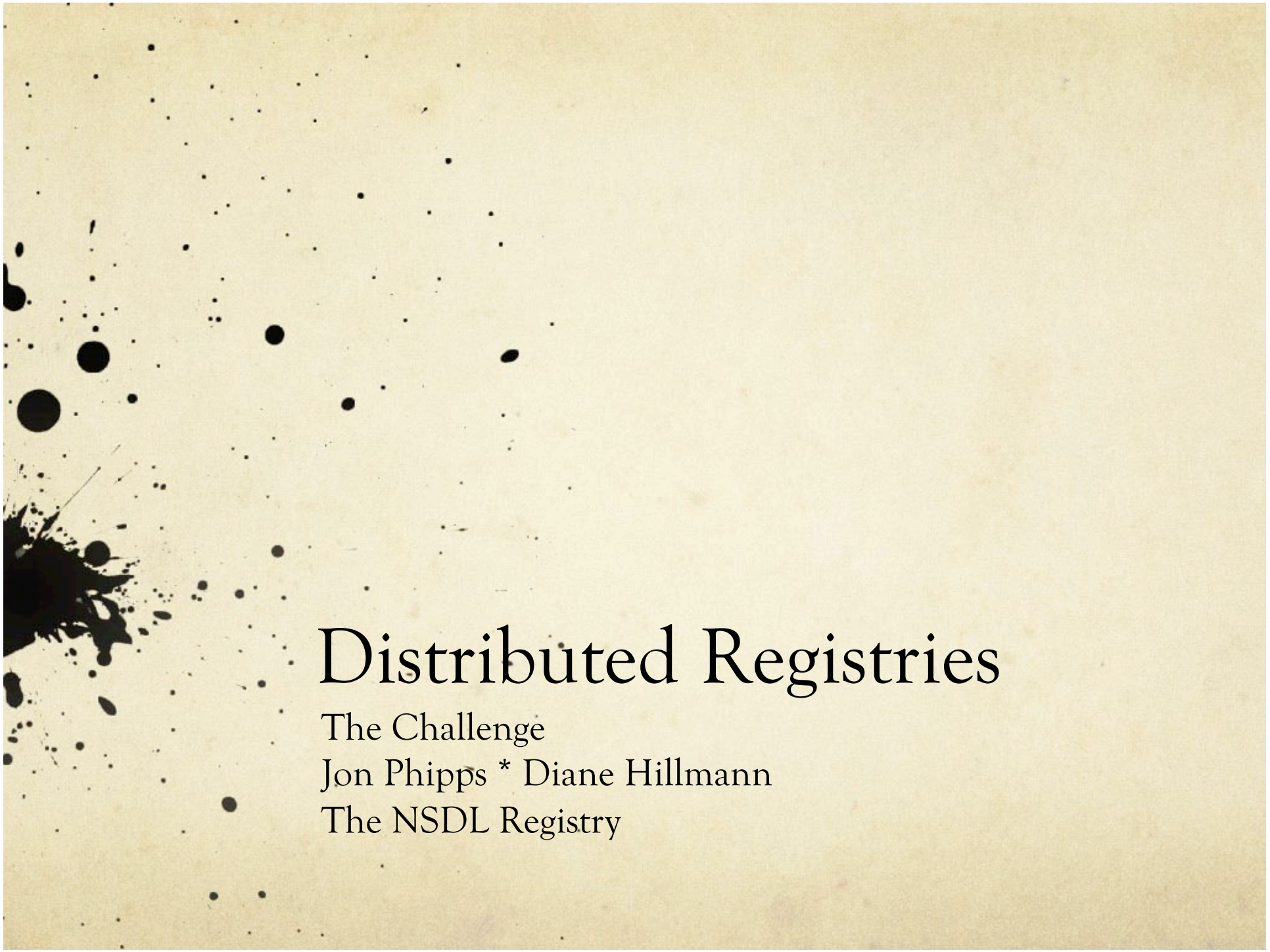
The Registry blog (gotta have one)
http://metadataregistry.org/blog

TRAC issue tracker (ask4account)
http://trac.metadataregistry.org/

Subversion repository
http://svn.metadataregistry.org/registry/

Jon Phipps
mailto:jphipps@madcreek.com

Thanks for listening ☺



Distributed Registries

The Challenge

Jon Phipps * Diane Hillmann

The NSDL Registry

The Current Environment

- Registry development taking off … in multiple directions
- In order to ensure success for all, registries need to define their users broadly—to include those who don't yet know they exist
  - They need to be willing to direct users who approach the task of discovery through their portal to some other registry if that's more appropriate

The Challenge is ...

- Discovery
  - Finding available vocabularies/metadata schemas/APs regardless of starting point
- Use
  - Directing users easily from one registry to another
  - Reliable m2m interactions between registries to ensure timely and up-to-date information
  - Providing a base level of services to users, no matter their starting point

Strawman #1

- Registries enable automated caching of other registry content
- Each registry can then enable searching of available content without the difficulties of federated search
- Requires agreements on mapping between vocabulary standards (SKOS, Zthes, etc.)
- May require “registry of registries” or other mechanism to locate content
- May require agreement on a common search API (SRU, OpenSearch, Z39.50)

Strawman #1 Questions

- Is this politically feasible, given the wide variety of funding regimes represented? Would "branding" be useful/ necessary?
- Can such a solution accommodate both open and semi-open registries? Differing assumptions about licensing and access?
- What other registry discovery mechanisms are available?
- Will registries support discovery based on term/concept label search as well as registry-level descriptive metadata?

The Politics of Re-Use

- What can we enable using distributed methodologies, and do we want to enable them? Are there IP issues? If so, how are they expressed?
  - Extending existing vocabularies
  - Adopting existing work (terms or whole vocabularies) and changing the domain or focus in a reused vocabulary
  - What about abandoned vocabularies?

Other Potential Issues

- How much information can or should be shared?
- How do we accommodate differences in approach that might affect download/caching/presentation of one vocabulary in another registry?
- At what point should users be directed from one registry to another, instead of browsing the vocabulary from the starting point registry?
- What would we need to accomplish to begin trying this out?