CORES

Contract N°: IST-2001-35273

A forum on shared metadata vocabularies

D 24

CORES Registry and Registration Tool: evaluation report

Document number:

CORES-UKOLN-WP2-D24-Final-20030630

General Information

Title	CORES Registry and Registration Tool: evaluation report
Creator	Rachel Heery
Creator	Pete Johnston
Subject-Keywords	Metadata schema registry
Description	This report presents an evaluation of the CORES Registry, taking into account the feedback provided at the Schema Creation and Registration Workshop and during subsequent use of the registry software.
Publisher	Cores Project
Contributor	Thomas Baker
Contributor	Andras Micsik
Contributor	Csaba Fulop
Date	30 June 2003
Type	Text Manuscript
Format	application/MSWord 2000
Identifier	
Document Number	CORES-UKOLN-WP2-D24-Final-20030630
Language	English
Rights	European Commission; public distribution on CORES Web site

Dublin Core metadata for this document

```
<?xml version="1.0"?>
<rdf:RDF xml:lang="en"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc=”http://purl.org/dc/elements/1.1/”>
<rdf:Description rdf:about = “”>
<dc:title> Cores Registry and Registration tool: evaluation report/dc:title>
<dc:creator>Heery, Rachel </dc:creator>
<dc:creator>Johnston, Pete</dc:creator>
<dc:subject> Metadata Schema Registry </dc:subject>
<dc:description> This report presents an evaluation of the CORES Registry,
taking into account the feedback provided at the Schema Creation and
Registration Workshop and during subsequent use of the registry software.
</dc:description>
<dc:contributor> Baker, Thomas </dc:contributor>
<dc:contributor> Micsik, Andras </dc:contributor>
<dc:contributor> Fulop, Csaba </dc:contributor>
<dc:publisher> CORES Project </dc:publisher>
<dc:date> 2003-06-30 </dc:date>
<dc:type> Text </dc:type>
<dc:format> application/MSWord </dc:format>
<dc:identifier> CORES-UKOLN-WP2-D24-Final-20030630 </dc:identifier>
<dc:language> en </dc:language>
<dc:rights> European Commission; External circulation on CORES Web site
</dc:rights>
</rdf:Description>
</rdf:RDF>
```

Table of Contents

1. Introduction

This report provides an evaluation of the usefulness of the CORES Metadata Schema Registry and Schema Creation tool. The evaluation is based on the responses received from users of the software, primarily at the "Schema Creation and Registration Workshop" held in Budapest in March 2003[i].

The function of the CORES Schema Registry is to provide a publication environment for the descriptions of metadata vocabularies and their constituent terms. Those descriptions include not only the standard declarations of terms provided by the creators of the vocabularies, but also descriptions of the usages of those terms as they are deployed by implementers in the context of a project or application. Earlier research projects introduced the term "application profile" to describe these implementer "optimisations" of standard metadata vocabularies.[ii]

The CORES Registry indexes machine-readable descriptions of metadata vocabularies provided as RDF data (schemas) and provides interfaces that allow human users and software applications to browse and query that data. The CORES Schema Creation Tool enables the managers of metadata vocabularies to generate such schemas and to submit them to the Registry for indexing.

2. The Workshop

2.1. Purpose

The workshop provided an opportunity for the creators and implementers of metadata schemas to explore the usefulness of the CORES Registry software by creating and registering schemas for metadata vocabularies with which they were familiar. The workshop also provided an opportunity for participants to gain a working understanding of RDF and its application to the description of metadata vocabularies.

The workshop aimed:

- to provide a brief introduction to RDF and specifically to the use of RDF to describe metadata vocabularies
- to introduce the CORES Registry data model: the classes of resources described, their attributes and relationships, and their representation using RDF
- to allow participants to use the CORES Schema Creation Tool to create and register schemas
- to allow participants to navigate schemas via the CORES Registry HTML interface and via some other tools

- to explore the effectiveness of the CORES data model for representing a variety of element sets and application profiles
- to evaluate the functionality of the CORES Registry and Schema Creation Tool

2.2. Target Audience

The workshop was targeted at those with an interest in creating and managing metadata vocabularies. Workshop participants were expected to be familiar with a specific metadata element set that they could use as a basis for input into the schema creation tool. The workshop publicity emphasised that participants were not required to have a detailed knowledge of RDF Schema semantics or RDF/XML syntax.

2.3. Participants

As this was a 'hands on' workshop the limited number of attendees allowed for one-to-one discussion and technical support. Attendees included researchers with existing expertise in metadata schema registries from Japan, USA, Italy and Germany.

There were 12 external participants and 8 partner participants.

External participants by country		External participants by sector	
Japan	1	Higher education and research institutes	7
USA	1	National libraries	2
Germany	1	European and international agencies	2
Hungary	2	Corporate	1
Netherlands	3		
UK	3		
Italy	1		

The workshop was widely publicised but attracted only a small number of participants. Details of the workshop were announced on the CORES Web site tree months before it was held, with follow-up mailings as reminders. In order to explore the reasons for low attendance, the workshop attendees were asked the ideal notice for workshops as part of the workshop evaluation and most responded that 8 weeks was ideal with no respondents asking for more than 12 weeks notice.

The project concludes that the CORES workshop was targetting a relatively small audience of 'active' schema creators with detailed interest in schema creation; this was not a 'general interest' workshop so a large audience could not be expected. In addition, the target audience includes many individuals who prefer to get information over the Web rather than attending meetings.

2.4. Workshop programme

Thursday, 6 March

10.30-11.00	Registration and Coffee
11.00-11.10	Welcome to SZTAKI
11.10-11.30	CORES Project Update Makx Dekkers
11.30-12.15	Schema Registries and Context Rachel Heery, UKOLN
12.15-12.45	Overview of CORES architecture and introductory demonstration Pete Johnston, UKOLN
12.45-13.45	Lunch
13.45-14.00	Introduction to browsing the CORES Registry Pete Johnston, UKOLN
14:00-15.00	Introduction to RDF and RDFS Tom Baker, Fraunhofer
15:00-15.15	Coffee
15:15-16.00	The Registry Data Model and its expression in RDF Pete Johnston
16.00-16.30	Discussion/Questions

Friday, 7 March

09.30-10.00	Creating Schemas for the Registry: A Walk-Through Pete Johnston
10.00-10.45	Practical sessions: Element Sets Rachel Heery
10.45-11.00	Coffee
11.00-11.45	Practical sessions: Application Profiles Tom Baker
11.45-12.45	Some Other Tools for Authoring and Browsing RDF Pete Johnston and Andras Micsik
12.45-13.45	Lunch
13.45-14.15	Review of other registries Rachel Heery
14:15-15.15	Structured Discussion and Feedback

2.5. *Feedback*

Six feedback forms were returned (i.e. 50% of participants)

Most valuable aspects of workshop:

- Hands on exercises (2)
- Tool demonstrations
- RDF intro (4)
- Architecture and context (4)
- Meeting colleagues and networking (2)
- Applying data model to existing application profile

Could be improved:

- Demonstrations
- Balance between presentations and hands on sessions
- Waits for machines to re-boot, handovers between presenters
- Room layout, noise of computers and data projection (4)
- More time to use schema creation tool

3. Analysis of issues arising

This section summarises some of the issues that were raised by users of the registry software either during the Budapest workshop or during subsequent testing of the application against their real metadata vocabularies.

3.1. *Applicability of data model*

The registry data model is influenced by two main sources:

- the Grammatical Principles of the Dublin Core Metadata Element Set, particularly the concepts of "element refinement" and "encoding schemes" [iii]
- the idea that implementers optimise their use of metadata element sets for specific contexts, and that this customisation can be captured in the form of an application profile

The key entities described in the registry data model are:

- **Elements:** the formally defined terms which are used to describe attributes of a resource;

- **Element Sets:** sets of functionally-related Elements which are defined and managed as a unit;
- **Encoding Schemes:** mechanisms that constrain the value space of Elements;
- **Values:** the enumerated Values which form part of a Vocabulary Encoding Scheme;
- **Usages of Elements:** deployments of metadata elements in the context of particular applications;
- **Application Profiles:** sets of functionally-related Element Usages, created and managed as a unit;
- **Agencies:** persons or organisations responsible for the ownership or management of Element Sets, Application Profiles and Encoding Schemes.

The CORES data model is a modified version of that used within the earlier DESIRE and SCHEMAS projects. Several of the workshop participants had some familiarity with this earlier work or had encountered the application profile concept in other contexts (for example, DCMI Working Groups), and most engaged quickly with the model presented here.

However, after applying this model to their own application profiles during the workshop, participants raised a number of points where some clarification of the model was required or where it might be improved.

3.1.1. Clarifying element usage

Fundamental to the model of the application profile is the concept of an "element usage", where the use of a previously defined metadata element is tailored to the context of an application by (one or more of)

- A narrowing of its semantic definition
- Association with specific datatypes or encoding schemes
- Specification of local obligation/occurrence constraints

"Standard" values for these properties may be specified in the original declaration of the metadata element. If no values are specified in the description of the element usage, then it is implied that the initial values apply in the context of the application profile. However, this implied "inheritance" is not always clear to the user of the schema creation tool and the registry Web interface. It may be preferable to change the model (and the tools) so that the creation of a "uses" relation results in the explicit "copying" of these properties from the used metadata element.

The nature of the relationship between an element and an encoding scheme also requires clarification, i.e. there is some ambiguity as to whether specifying an encoding scheme mandates the use of that scheme or whether it presents the encoding scheme as a possible scheme for that element.

There was some discussion of whether the description of a metadata element should include a description of "standard" encoding schemes and obligation/occurrence constraints, or whether those values should always be supplied in an application profile, but it was felt that permitting them only in an application profile would make reuse more difficult and would run counter to the aims of encouraging convergence in the use of metadata vocabularies.

It was noted that the model and the tools permit multiple usages of a single element within an application profile, but discussion suggested that this was appropriate.

Finally, discussion highlighted that it is difficult to establish a clear rule to distinguish between when it is appropriate to "use" an existing metadata element (but with refined semantics) and when it is better to define a new metadata element specific to the application (but described as an element refinement of an existing element).

3.1.2. Element sets and application profiles describing multiple resource types

Discussion of the use of Dublin Core metadata tends to start from the premise that a metadata record is a description of exactly one resource. In real world metadata applications, however, it is common to provide descriptions of several related resources of different types and the relations between them, and many metadata element sets and application profiles are designed on this basis.

The registry data model does not explicitly represent the relation between a metadata element and a type or class of resource to which the element should be applied, and this can only be captured implicitly by creating separate element sets or application profiles, one for each type of resource. So for example, the RSLP Collection Description schema includes "groups" of metadata elements for describing a collection, the location of that collection, and a number of agents related to the collection or location.[iv] Those groups can be represented only as multiple (unrelated) application profiles. The features of the Web Ontology Language (OWL) that describe restrictions on how properties can be used with instances of specific classes may be useful in this context.

3.1.3. Hierarchical models

The registry model is closely aligned with the Dublin Core model, where a metadata record describes a single resource and consists of a simple "flat" set of attribute-value pairs, where the values of metadata elements are either literals or literals qualified by the name of an "encoding scheme". This fits well with the RDF model where a resource has a property whose value is either a literal (plain or typed) or a second resource.

Some of these difficulties are highlighted by attempts to represent the IEEE Learning Object Metadata (LOM) vocabulary within the registry model.[v] The LOM information model is formulated as a hierarchical model, with elements grouped into categories, and LOM instances have typically been represented in an XML tree-structure. Furthermore, closer inspection of the LOM highlights that a LOM instance provides metadata about multiple resources (a learning object, contributions to that object, entities related to those

contributions, and the LOM metadata record itself). Such a model requires a degree of reengineering before it can be represented in RDF, and current work on the development of the LOM RDF binding[vi] is highlighting issues of reconciling the "structural" and "conceptual" approaches.[vii] Further, the proposed LOM RDF binding recommends the use of Dublin Core metadata elements to express some LOM semantics: in the registry model, then, it would be described as an application profile - but some LOM implementers would not consider it as such.

3.1.4. Modelling the Schema as an entity

(This is discussed here as a data model issue although it underlies several of the usability problems discussed in section 3.2.)

The schema creation tool permits the author to describe multiple element sets, application profiles and encoding schemes within a single authoring session and to save this data within one RDF/XML document (schema). A request to submit data to the registry generates a sequence of HTTP POST requests that send RDF/XML documents, one for each resource described, to the server. The complete schema – the RDF/XML document that is saved as a file – is **not** submitted to the registry as a unit. The schema creation tool does not capture either a description of the schema itself or the fact that these individual resources are described in the same schema.

This means that metadata about the schema itself is not available from the registry. It is not possible for the registry to provide a link to a schema published on the Web by the author, and it is not possible to issue a query on the registry database that will select the set of resources described in a specified schema. This underlies several of the problems identified in section 3.2.3 below, and reduces the usefulness of the registry.

To address this, the data model should include “Schema” as an entity type, the schema creation tool should be modified to capture information about the schema being edited, and the registry interfaces should be amended to present that information. The relationship between a schema and instances of the other entities is (potentially at least) many-to-many: i.e., an instance of any of the entity types above may be described by statements in multiple RDF/XML documents, and each RDF/XML document may describe multiple instances of any of those entities.

The revised data model is outlined in Figure 1. (For clarity, this does not include annotations.)

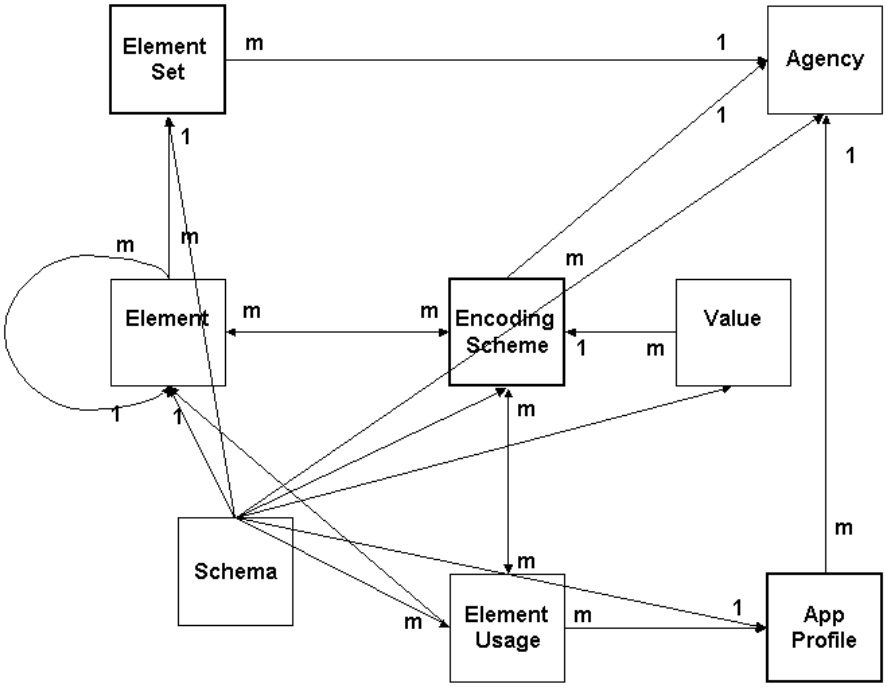

Figure 1: Revised data model with Schema as Entity Type

The property `rdfs:isDefinedBy` is used to describe a relationship between a resource and a Schema (an RDF/XML document describing the resource). It should be noted that the use of `rdfs:isDefinedBy` in this way is not in itself sufficient to support a query that can retrieve the triples provided by a specified schema. If a single resource is described by statements from two schemas, `rdfs:isDefinedBy` arcs record the relations between the resource and the schemas, but not between a schema and a specific set of triples. In order for the registry to provide access to the schema as created by the author, that file must be saved and made available, preferably by the author publishing it on the Web (an alternative would be for the submission process to post the entire RDF/XML document to be stored by the registry).

To summarise, the schema creation tool should, as the first step in the authoring process, capture metadata about the schema, and for each resource described in the schema (agencies, element sets, elements etc), an `rdfs:isDefinedBy` arc should be created to describe the relation between the resource and the schema. The registry Web interface should support the display of descriptions of "Schemas" as a class of entity (which would include the URL of the schema), and for every other type of resource the display would provide links to one or more schemas in which the resource is described.

3.1.5. Vocabularies, Schemas and the reuse of existing RDF schema data

The registry model does not assume a one-to-one relationship between a metadata vocabulary (an element set or application profile) and the RDF/XML documents (schemas) in which a vocabulary is described. That is, an element is a member of

exactly one element set, but information about the elements of a single element set may be provided in multiple schemas, and a single schema may contain information about multiple element sets and their elements.[1]

As noted above, the property `rdfs:isDefinedBy` can be used to describe a relationship between a resource and an RDF/XML document describing the resource. The RDF Vocabulary Description Language (RDF Schema) does not itself provide a class that represents a "vocabulary" resource type, or properties to describe the relations between individual terms and a vocabulary. Because the registry data model regards a vocabulary and a schema as separate resources, it does not assume that the resources described in a single schema form a single vocabulary, and it introduces application-specific properties to describe the relationships between, for example, elements and element sets.

The registry can (and indeed does) read and index the published descriptions of metadata vocabularies created by their owners quite independently of the registry using RDF VDL (RDFS). However, both the schema authoring tool and the registry browse and query interfaces are built on the application-specific data model. For the data to be presented by these tools, it must be merged with supplementary RDF data that describes the relations between resources that are required by the registry application. So, for example, the registry reads and indexes the schemas published by the Dublin Core Metadata Initiative, but because the DCMI schemas do not make an explicit distinction between the schema and the vocabulary, this DCMI data is merged with additional data provided by the registry administrator.[2] Figure 2 illustrates an example of the data provided by the two sources:

[1] This distinction between a functional vocabulary and a schema was clarified in discussions on the dc-architecture mailing list, particularly by Patrick Stickler in a number of messages, perhaps best summarised by the examples provided in: http://www.jiscmail.ac.uk/cgi-bin/wa.exe?A2=ind0301&L=dc-architecture&P=4782

[2] It is worth noting that a similar process appears to be followed in the data indexed by Patrick Stickler's Nokia Semantic Web demonstrator at http://sw.nokia.com/ where, for example, the schema http://sw.nokia.com/schemas/DC/voc_DC-1.1.rdf provides information on the relationships between Dublin Core elements and the DCMES vocabulary.

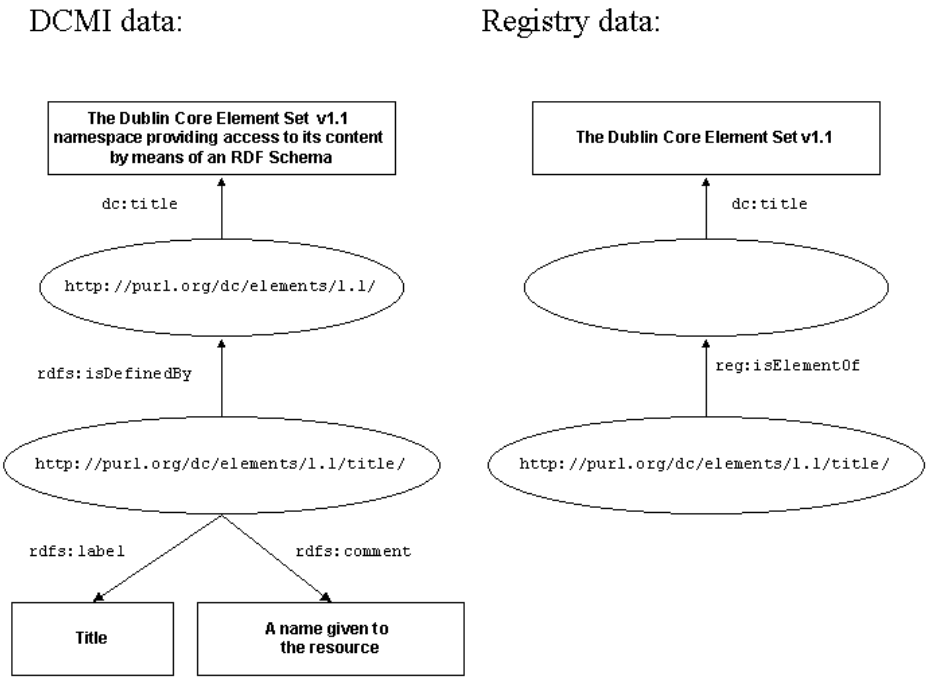

Figure 2: Merging data from two schemas

This makes the reuse of existing RDF data a more complex process than simply reading and indexing the externally created documents.

3.1.6. Encoding Schemes and datatyping

The DESIRE data model on which the MEG/CORES model was based describes the datatype associated with a metadata element as distinct from the use of an encoding scheme. In practice however, specifying that a value is associated with an encoding scheme determines the datatype of the value (e.g. a value using the W3CDTF encoding scheme is by definition a date).

The recently revised RDF specifications introduce support for the association of datatypes with literal values.[viii] The DCMI Architecture Working Group has discussed the possible use of RDF datatypes to model at least some subsets of encoding schemes, but no recommendation has yet been made.

3.1.7. The registry RDF vocabulary

The CORES registry uses an application-specific vocabulary to capture many of the properties of metadata elements, such as datatyping, occurrence and obligation constraints.

The vocabulary should be revised in the light of the following developments

- minor modifications to the conventions used in the DCMI schemas (especially the use of a `dc:description` to capture the "comments" about an element),
- the introduction of support for literal datatyping in the RDF specifications,
- the publication of the Web Ontology Language (OWL) specifications[ix]

to align the registry more closely with other work in the modelling and representation of ontologies. OWL allows for the description of restrictions on how properties can be used by instances of a specified class, and this may provide a better mechanism for modelling element usage.

3.2. Usability of tools

3.2.1. Help/support

One workshop participant noted that users need time to develop familiarity with software tools. More "help" features would be beneficial and a "wizard"-based approach to schema creation might be more appropriate. The availability of training and tutorial materials would be an important factor in the take-up of the tools.

3.2.2. Bugs

A number of bugs were highlighted during the workshop. These included:

1. The schema creation tool allowed an author to describe an element as a refinement of itself, but could not subsequently process that schema. It should not be possible to describe an element as a refinement of itself. (Fixed)
2. The schema creation tool did not support the removal of a previously created element refinement relation. (Fixed)
3. A search from the schema creation tool that generated no hits did not indicate clearly that result. A suitable message should be displayed. (Fixed)
4. The authentication procedure failed to check the agency identifier correctly on submission to the registry (Fixed)
5. If submission to the registry server failed for any reason, this failure was not made clear to the submitter. A suitable error message should be displayed. (Fixed)
6. The search function from the HTML interface produced incorrect results on searching agency descriptions (Fixed)
7. Absolute font sizes are encoded in the Cascading Style Sheet for the HTML interface, which makes it impossible to resize the display.

3.2.3. Requests for features

The following issues are listed as request for improvements, rather than bugs in the software:

1. The registry should provide access to the schemas indexed, either by recording the locations where they are made available by their owners or by caching copies of the schemas and making available those copies.
2. The schema creation tool permits free text literals as the values for the obligation and datatype properties, but those values should be restricted to a controlled list of values, perhaps in the latter case the XML Schema built-in datatypes.
3. The registry data model emphasises that elements are created in the context of an element set, but in some cases the presentation of the data by the schema creation tool and the registry does not highlight this (e.g. the label "Title" should be qualified by an indication that it is a reference to "Title" from the "Dublin Core Metadata Element Set").
4. All data submitted to the registry is publicly visible. The registry should provide a mechanism to support limiting the public visibility of data that is "work-in-progress".
5. To reuse elements when describing an application profile, the schema creation tool permits a basic keyword search on the registry for elements and encoding schemes. However, application profiles typically reference many (even most) elements of a single element set. To facilitate description of such profiles, the tools should support the retrieval from the registry of descriptions of **all** the elements of a specified element set, so that the author can select them for reuse in an application profile. Similarly, to select from amongst existing encoding schemes, more refined searching may be useful, for example to retrieve encoding schemes associated with a specified element.
6. Application profiles are often closely related to each other, and large and complex profiles may differ in only a relatively small number of details. To facilitate the description of such profiles, the tools should support the downloading of the descriptions of all the component resources of an existing application profile so that it can be opened in the schema creation tool and used as the basis of a new profile. Because there is no necessary correspondence between an application profile and a schema, this may not correspond exactly to the contents of a published schema, and it may be necessary to generate the data by querying the registry database. (Care would also be required in the reassignment of identifiers.)
7. From an administrative perspective, it would be useful to query the administrative metadata that is associated with the resources submitted to the registry.
8. The registry could provide other presentations of the submitted data, for example an HTML representation of a complete application profile that uses the conventions described in the draft CEN/ISSS guidelines for DC application profiles[x]
9. The registry could generate forms-based tools for the creation of RDF/XML "instance" metadata records conforming to a specified application profile. While this is possible in theory, it would probably require additional (element-specific) information on the presentation of the form (for example, the size of a text box for a literal value might vary).

10. The registry permits the submission of the URL of an XML Schema associated with an application profile, but the data model and tools might be extended to include XML Schema as an entity and to capture and present metadata about an XML Schema.

3.2.4. Configurability/extensibility of tools

One participant enquired about how easy it would be to extend the tools to permit the creation of additional data about elements, element sets etc.

It would not be too difficult to support such extensibility in the browse and search interfaces to the registry. However, it would be considerably more problematic to provide in the schema creation tool, since the introduction of new properties would require changes to the graphical interface layout and behaviour, and depending on the nature of the data may require major graphical redesign.

3.2.5. Multilingual support

The MEG registry software that the CORES project adapted does not provide multiple language support. It would require quite significant modification to provide that support.

3.3. The Registry API

The registry has a very simple query API, accessible via HTTP GET. The API supports a keyword search on the literal values of (pre-selected) properties of resources of a specified type, e.g. the request below performs a search for the string "condition" in the literal values of the properties of elements, and returns an RDF/XML document (XML namespace declarations not shown here):

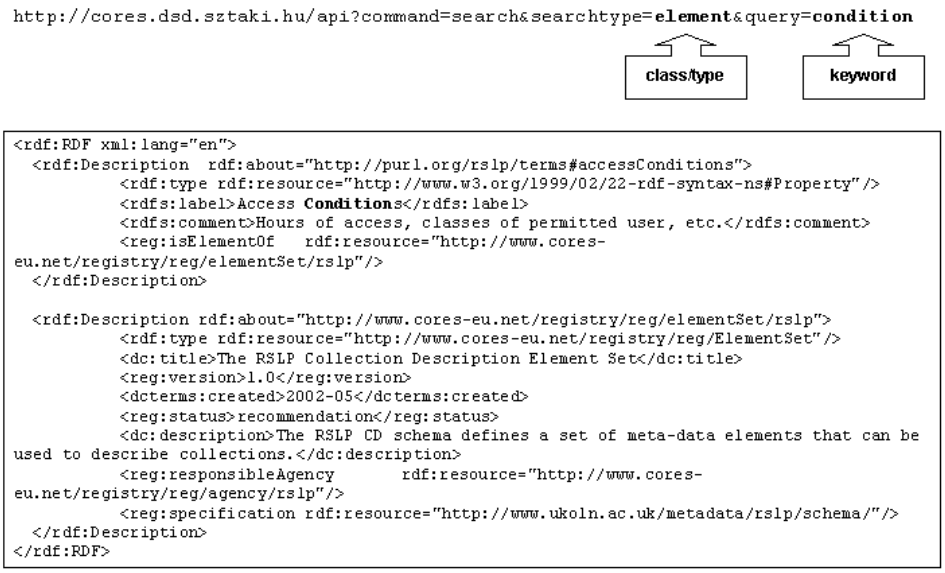

Figure 3: Querying the registry

This API was designed primarily to support the search requirements of the schema creation tool. As noted in section 3.2 above, it would be useful for the schema creation tool to perform more complex queries (such as the selection of all the elements of an element set or all the usages within an application profile). To support external applications, a richer API would almost certainly be required. For example, to provide a "dictionary lookup" function, where the registry returns metadata about a specified resource, it should be possible to select resources by URI.

The DCMI Metadata Schemas Registry has recently provided an API supporting a number of simple queries, and although the CORES registry deals with a wider range of resource types and further investigation of application requirements is necessary, this may provide a useful starting point.[XI] The DCMI API is implemented over SOAP, but the same functionality could be provided using HTTP GET.

3.4. Policy framework for registry

3.4.1. Availability of CORES registry

Both at the workshop and subsequently, potential users of the registry expressed some serious concerns about the continued availability of the current CORES registry server, given the limited duration of the CORES project, and the implications for the longer-term access to schemas created using the CORES software if/when the current CORES registry service ceased to be available. Most workshop participants expressed the view that they

would be reluctant to invest effort in creating data specifically for submission to the registry unless they

3.4.2. Collection policy and quality assurance

Questions were also raised about the "collection policy" of the registry and the scope of the schemas that might be submitted, and also how the administrators of the service could "quality assure" data submitted to the registry and presented through its interfaces.

The decentralised nature of the data indexed by the registry does raise the problem that schema creators may edit or delete resources that have been referenced by others

3.4.3. Access to source code for registry software

At least one workshop participant expressed interest in access to the current source code for the tools.

Over the last year the CORES project has collaborated with the MEG Registry project within the UK, adding enhancements to the registry and schema creation tool software originally developed for the MEG registry project. At least one workshop participant expressed interest in access to the current source code for the tools. If there is a wider interest in the software, it may be appropriate to consider managing it via a centralised open source software environment such as SourceForge.[xii] SourceForge provides support management tools (Tracker) and release management using a Web-based file release system. Release testing and integration of fixes could be co-ordinated. The lead developer of the MEG registry software (Dave Beckett, ILRT) has expressed an interest in leading this work in collaboration with Andras Micsik at SZTAKI. In the short term, however, it is important that the CORES version of the software is made accessible via CVS.

3.4.4. Persistence Policy for CORES registry

At the workshop, the project sought to establish clear distinctions between three resources, and to emphasise that the availability of any one of these resources can and should be considered largely independently of the other two:

- The registry software (the schema creation tool and the registry server)
- The current registry server operated by SZTAKI using the registry server software
- The schemas indexed by the current registry service, some of which have been created using the schema creation tool and some of which are data published on the Web quite independently of the registry

In early June 2003, the project issued a statement clarifying the persistence of the CORES registry beyond the formal end of the project on 30 June 2003[xiii]. This is summarised below:

- UKOLN and SZTAKI -- are committed to maintain the registry server and its Web interface for at least one year (i.e., through June 2004) within the limits of hardware and software resources as of the project end. They will perform periodic checks of the content of the database and reserve the right to remove data that is considered inappropriate for the registry. A second server is provided as a "sandbox" registry for testing and training.
- The CORES registry software will be made available under “open source” license conditions from the CVS server at ILRT, University of Bristol

Responsibility for storing and making available copies of the schemas submitted to the registry server lies with the agencies creating and submitting those schemas. The registry server does **not** store copies of the schemas submitted. Since the schema creation tool does not capture metadata about the schema, schema creators should report the URLs at which they publish their schemas

4. Summary

Participants in the workshop acknowledged the usefulness of the registry as a tool to support the developers and implementers of metadata vocabularies, particularly in the function of disclosing information about implementer deployment of metadata vocabularies.

While they requested clarification on some aspects, workshop participants understood and engaged with the general principles of the application profile model. They appeared to confirm that it reflected real world practice in the deployment of Dublin Core metadata, and schemas for a small number of DC-based application profiles have been registered. However, they expressed some reservations on how well it accommodated metadata vocabularies that do not conform to the Dublin Core model and further investigation of examples such as the IEEE LOM is required.

The tools have proved relatively easy to use, once users are familiar with the data model. The use of "drag and drop" within the schema creation tool provides an easily understood mechanism for the creation of relationships between resources.

However, it is clear that significantly more effort is required to encourage the owners of metadata vocabularies to create schemas for submission to the registry, and to support them in that process.

The registry application also depends at least in part on the use of an application-specific vocabulary and this presents obstacles to the reuse of existing RDF/RDFS data published on the Web. Data can be reused but it must be supplemented with application-specific data. The RDF vocabulary used by the registry should be reviewed to minimise the use of application-specific vocabulary, and to make use of RDF datatyping and the Web Ontology Language where appropriate.

Uncertainty about the continued availability of the CORES registry server has also caused potential contributors to question the benefits of investing effort in creating and registering schemas. The project has sought to address this by clarifying the distinctions between the data, the service currently provided by the CORES server at SZTAKI, and the software, and by making a commitment to continue operating the present server for a 12-month period.

Even with these guarantees that the CORES registry will continue to be available, schema owners may perceive the benefits of submitting their schemas as somewhat limited. As noted above the current API is simple and is designed primarily to serve the requirements of the schema creation tool rather than external applications. The usefulness of the human-readable Web interface depends largely on being able to present a corpus of data that is of interest to a community. This is not simply a question of the number of vocabularies registered but also of the application requirements those vocabularies are designed to address. A registry that indexes a number of schemas from very diverse sources may be of interest to researchers interested in the generic modelling issues: to be useful to metadata implementers working within a domain, a registry must provide access to a "critical mass" of schemas relevant to that domain.

In summary, the CORES registry has provided a context within which schema implementers can test and evaluate the usefulness of the data model and the software tools. This has raised some questions of how broadly applicable the model is, particularly for schemas not based on the Dublin Core "grammatical principles", and of the registry's capacity to reuse existing RDF/RDFS data published on the Web. However, users have also recognised the potential value of the registry as a tool for the disclosure and discovery of information about metadata vocabularies and their implementations.

---

[i] *CORES Schema Creation and Registration Workshop*, MTA SZTAKI, Budapest, 6-7 March 2003. http://www.cores-eu.net/registry/ws1/

[ii] Heery, R and Patel, M. (2000). Application profiles: mixing and matching metadata schemas, *Ariadne* 25, September 2000. http://www.ariadne.ac.uk/issue25/app-profiles/

[iii] DCMI Usage Board. (2003). *DCMI Grammatical Principles*. http://dublincore.org/usage/documents/principles/

[iv] Powell, A. (2000). *RSLP Collection Description Schema*. http://www.ukoln.ac.uk/metadata/rslp/schema/

[v] IEEE Learning Technology Standards Committee (2002). *Learning Object Metadata 1484.12.1 - 2002*

[vi] Nilsson, M., Editor. (2002) *IEEE Learning Object Metadata RDF Binding*. Working Draft, 26 August 2002. http://kmr.nada.kth.se/el/ims/md-lomrdf.html

[vii] Nilsson, M. (2003). *Semantic Issues with the LOM RDF Binding*. 15 January 2003. http://kmr.nada.kth.se/el/ims/md-lom-semantics.html

[viii] Manola, F., and Miller, E., Editors. (2003). *RDF Primer. World Wide Web Consortium W3C Working Draft, 23 January 2003*. http://www.w3.org/TR/2003/WD-rdf-primer-20030123/

[ix] McGuiness, D., and van Harmelen, F., Editors. (2003). *OWL Web Ontology Language Overview. World Wide Web Consortium W3C Working Draft, 31 March 2003*. http://www.w3.org/TR/2003/WD-owl-features-20030331/

[x] Baker, T., Dekkers, M., Fischer, T., and Heery, R. (2003). *Dublin Core Application Profile Guidelines*. Draft CWA. http://www.cenorm.be/isss/Workshop/MMI-DC/application-profile-for-comment.pdf

[xi] Wagner, H. (2003). *DCMI Metadata Registry Application Interface*. http://wip.dublincore.org/

[xii] *SourceFourge*. http://sourceforge.net/
[xiii] CORES Project (2003), *Policy on Persistence of the Registry*. http://www.cores-eu.net/registry/persist/