

# Metadata schema registries in the partially Semantic Web: the CORES experience

Rachel Heery, Pete Johnston  
UKOLN, University of Bath, UK  
{r.heery, p.johnston}@ukoln.ac.uk

Csaba Fülöp, András Micsik  
Computer and Automation Research Institute of the Hungarian Academy of Sciences (SZTAKI),  
Hungary  
{csabi, micsik}@dsd.sztaki.hu

## Abstract

*The CORES metadata schemas registry is designed to enable users to discover and navigate metadata element sets. The paper reflects on some of the experiences of implementing the registry, and examines some of the issues of promoting such services in the context of a "partially Semantic Web" where metadata applications are evolving and many have not yet adopted the RDF model.*

**Keywords:** metadata schema registries, RDF, XML, Semantic Web.

## 1. Introduction

The CORES project has explored the potential for supporting the creation and re-use of metadata schemas using Semantic Web technology [1]. As part of the European Community funded IST Semantic Web Technologies programme, CORES has promoted the use of metadata schema registries to support the disclosure, discovery and navigation of information about metadata element sets stored as schemas distributed on the Web. Such a 'schema navigation service' provides users (both human and software) with information about existing metadata element sets and the terms used within them. In particular, it assists implementers in locating and re-using existing schemas. The project is intended to support an emerging network of schema creation tools and registries, and interest has been expressed in its outcomes not only from the digital library and cultural heritage sectors, but also from the corporate sector, e-Government and e-Science. Registries are seen as part of the Semantic Web infrastructure which, in the future, will act as a foundation for merging and mapping data, and bringing the Web closer to semantic interoperability [2].

In particular, CORES addresses the need to manage the proliferation of metadata element sets within the digital world. Although there are federated systems which mandate the use of particular element sets, the increase in the range and quantity of business (whether education, commerce, government or culture) taking place on the Web means that new and variant schemas are emerging at a significant rate.

Increasingly, as the digital library becomes embedded in the wider sphere of e-Learning and e-Science, implementers are challenged to manage interworking systems based on different metadata standards. CORES envisages a network of schema registries supporting the discovery and navigation of core element sets. By 'declaring' such element sets in structured schemas and making those schemas available to navigable registries, their owners make them accessible to other users who can find and re-use either a whole element set or the component data elements, or even a particular localisation of the element set captured as an 'application profile' [3]. If schemas can be located easily, implementers will be encouraged to re-use existing work, and to take a common approach to the naming and identification of data elements.

In order to enable such core element sets to be shared, there needs to be a common model for identifying data elements and declaring schemas. Building on the specifications being developed within the W3C's Semantic Web activity, particularly RDF and the RDF Vocabulary Description Language (RDF Schema) [4, 5], CORES has been working towards this target over the last year. Firstly CORES brought together standards makers to agree on a common approach to identifying the elements in their standards, and secondly CORES has provided implementers with the tools to create RDF schemas that describe the element sets in use in their projects and applications [6, 7].

CORES has a limited duration and funding so although its vision is wide its aims have of necessity been modest. The project started in April 2002 and runs until June 2003, at the time of writing there are two months remaining until the end of the project. This paper will report achievements to date and highlight issues that have arisen.

## 2. Usage scenario

In addition to navigation, schema registries might offer a number of added value services such as mapping between metadata element sets, providing links to usage guidelines, or presenting multiple-language versions of schemas. The CORES Registry is limited to demonstrating a simple navigation service offering information about agencies,

element sets, data elements, application profiles, encoding schemes, and encoding scheme values, with the capacity for registered users to annotate this data.

A typical usage scenario for a registry offering such a navigation service might be to support interoperable on-line services within a federated organisation, such as an international corporate knowledge management system, a government information framework, or a distributed educational service. So, for example, a government might wish to encourage interoperability between its various systems and might use a registry to encourage effective collaboration on creation and re-use of schemas amongst government departments and agencies. The government 'interoperability manager' might mandate use of simple resource discovery metadata schema such as Dublin Core, whilst acknowledging that government departments and agencies might need additional specialist metadata elements for their particular systems. Any variant schema or usages introduced by departments or agencies would be indexed in the registry. Thereby other implementers across government who wished to build new systems could easily locate appropriate existing data elements and element sets, or be confident in introducing new data elements when necessary. Application developers outside government who wished to exchange data with government systems could use the registry to locate information about element sets in use within government.

Annotations might be added against element sets to indicate details of deployment, whilst annotations against data elements might give guidelines for use.

This scenario envisages a registry that is primarily designed for human use, for people to record and share information about metadata schema. However, if the registry is based on RDF schemas, there is potential for adding RDF based services to the registry such as downloading schema to RDF based applications, or mapping between element sets.

### 3. The CORES Registry and Schema creation tool

The CORES registry builds on earlier work within the Metadata for Education Group (MEG)<sup>1</sup> registry project, which in turn was informed by the work of the DESIRE and SCHEMAS projects [8, 9, 10]. 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" [11]
- 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 [3]

The key entities described in the CORES/MEG 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. Syntax Encoding Schemes specify that the value of an Element conforms to a specified format or pattern; Vocabulary Encoding Schemes enumerate a list of permitted Values.
- **Values:** the enumerated Values specified by a Vocabulary Encoding Scheme. (Values are not specified for a Syntax 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.

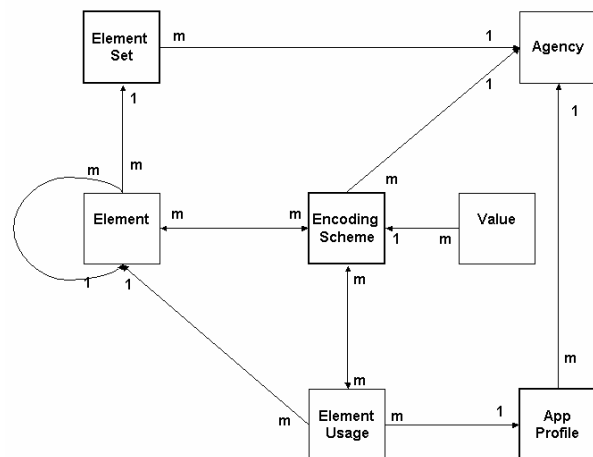
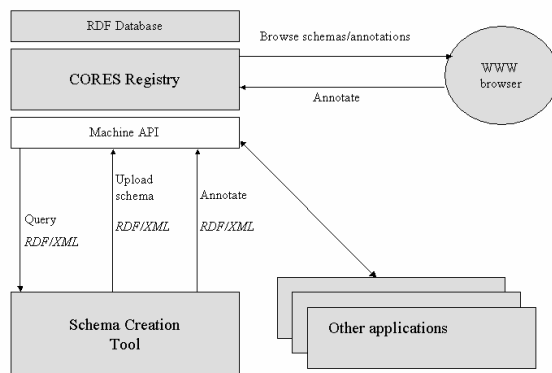


Figure 1: The base registry data model

There is presently no consensus on conventions describing application profiles in machine-processable form, and the CORES data model offers a prototype for doing so. To date, descriptions of application profiles have typically taken the form of human-readable descriptions of the usages of metadata elements within particular applications or domains (e.g. the DC-based application profiles developed by DCMI working groups for the description of particular classes of resource [12]). The registry models an application profile as a set of element usages, each of which references or "uses" a metadata element previously declared as part of an element set.

<sup>1</sup> The UK Metadata for Education Group (MEG) provides a forum for discussing the description and provision of educational resources at all levels across the UK. See <http://www.ukoln.ac.uk/metadata/education/>

The CORES registry provides a human readable interface for navigating the content of schemas, and an API for uploading and downloading schema. The schema creation tool provides an authoring environment and interacts with the registry server via its query and upload APIs. The tool is designed to permit implementers to create schemas without a detailed knowledge of RDF or its XML syntax. It allows the author to save schemas as RDF/XML documents and to submit them to the registry server.



**Figure 2:** The registry architecture

The CORES registry software is an enhanced version of the software developed for the MEG registry, and the schema creation tool is based on the MEG schema creation tool [13].

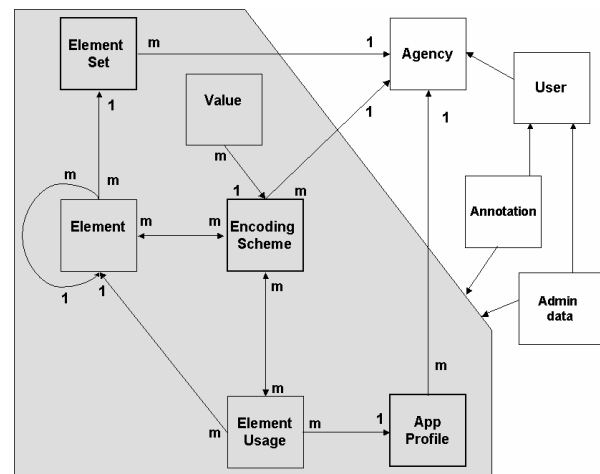
### 3.1. CORES Registry

CORES continued work on the MEG Registry code base, enhancing the software by adding two main pieces of functionality: authentication and annotation, and also enabling the display of administrative metadata associated with schemas. User registration data, annotations and the schema registry are all stored in an RDF database. The registry is implemented as a set of Perl scripts, using the Redland RDF toolkit [14] to query and manipulate the RDF databases.

Collected schemas are served in a similar way to the preceding SCHEMAS and MEG registry prototypes. Users may browse the lists of elements, element sets, encoding schemes, and application profiles, or they may search for word occurrences in the text describing these entities. Selecting an item, a detailed view is shown, containing the complete definition of that item and references to related items, such as refined elements, encoding schemes used. In this way the connections between various application profiles, element sets and encoding schemes become easily explorable.

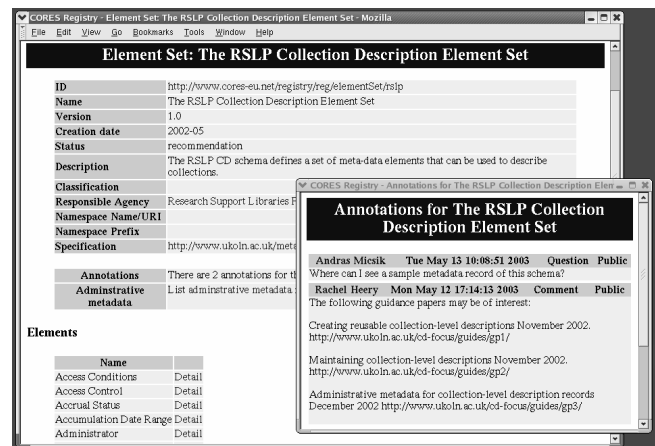
Authentication is based on the concept of agencies. Agencies represent organizations who use, create or publish schemas. Only members of the agency may create or

modify schemas of that agency. Users have their individual accounts in the system, and they may be associated with agencies. This association is a supported process in the system, where the agency leaders may easily accept or deny membership requests using a Web interface. All annotation or schema modification requests are checked for proper authentication and authorisation in the registry.



**Figure 3:** The extended registry data model

Administrative metadata provides the editing history and authors of the schema. Administrative metadata and annotations can be displayed in a separate pop-up window.



**Figure 4:** Annotating registry entries

Annotations can be made by any registered user on all classes of information except agencies. Annotations may be public or private and may also have a type e.g. question, comment. The list of annotations may be used to record discussions, ask experts' opinion, and collect usage data. Annotations can be created using the Web interface and uploaded using the API, but annotation display is available only through the Web interface. The RDF vocabulary for

annotations is derived from Annotea, the annotation toolkit of W3C [15].

### 3.2. Schema creation tool

The schema creation tool is written in Java, making it available on a wide variety of platforms, with a graphical user interface based on Java Swing. RDF support is provided by the Jena RDF toolkit [16]. The client communicates with the registry using the HTTP protocol, exchanging data in RDF/XML.

The schema creation tool provides an easy to use interface for creating and editing RDF schemas. RDF schemas created by the tool can be stored locally and/or submitted to the registry (as long as the user is an authorised member of an agency). The tool can be used in a standalone fashion, or interactively with the registry. This tool can also be used to define element sets, application profiles, and encoding schemes. There is a simple mode for beginners to create application profiles only, and there is an advanced mode in which element sets and encoding schemes may also be defined. From the schema creation tool, a search window queries the registry and offers any elements or encoding schemes found there for re-use. Re-using elements and encoding schemes is achieved with drag-and-drop operations. The annotation facility is also available in the schema creation tool.

## 4. Implementation issues

CORES has been able to explore the potential for deployment of RDF based tools in relation to schema creation, navigation and re-use. The project has been able to consider the role of registries within the Semantic Web, and how creation and maintenance of schemas might be managed. Incidentally CORES has been faced with the contradictions of exploring service provision while being funded as a short term project. A number of issues have arisen.

Several of these issues were discussed with implementers during the CORES workshop early in 2003. The registry and the schema creation tool were introduced during a hands-on workshop held in Budapest in March 2003 [17]. The basics of creating application profiles and the CORES model for the description of application profiles in RDF were explained for the participants, who were prospective implementers of application profiles or representatives of other projects with similar targets. Participants were able to experiment with creating an application profile and uploading it into the registry.

This workshop raised some general issues:

- Populating the registry: To what extent can the registry application reuse existing RDF/RDFS data made available by implementers? What is the potential for a service based on gathering/harvesting schemas distributed on the Web?

- Applicability of registry model: Is the simple DC-like model useful more generally? How well does it represent those cases where metadata elements form a hierarchy not a list or "flat file"? Does inheritance in element usage need to be defined more clearly? For example, are definition, comment, data type, etc. fields inherited or not?
- Persistence: How do questions of the persistence of the CORES registry service affect the willingness of implementers to contribute schemas to the registry?
- Deployment in an XML world: Whilst there has been a lot of interest in the Semantic Web, the reality is that applications need to be deployed in a largely XML world. What are the implications for RDF-based registries?

### 4.1. Populating the registry

The schemas read and indexed by the CORES registry are RDF/XML documents. These documents contain RDF data describing the terms (the elements and encoding schemes) used in the statements that make up metadata records. The schemas also provide metadata about element sets and application profiles and the agencies that manage them. An RDF schema provides unique identifiers for these resources, describes relationships between them, and provides human-readable documentation about them.

The registry draws data describing metadata elements and encoding schemes from two sources:

- (where available) RDF Vocabulary Description Language (RDF Schema) descriptions of "standard" metadata element sets published by the agencies who own/administer them (e.g. the DCMI RDF schemas);
- schemas created by implementers using the CORES schema creation tool.

Although the registry does make use of the basic vocabulary provided by RDF Vocabulary Description Language (RDF Schema), it also relies on a registry-specific RDF vocabulary to express some application-specific semantics, particularly to describe relationships between resources. The registry model seeks to avoid assuming one-to-one relationships between an element set or application profile, the schema(s) (the RDF/XML documents) in which that element set or profile is described, and the XML Namespace names used in the RDF/XML syntax. That is, an element set and a schema are treated as separate resources: 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.<sup>2</sup>

---

<sup>2</sup> This important distinction between a "functional vocabulary", a schema, and an XML Namespace was clarified in discussions on the dc-architecture mailing list, particularly in a number of

The RDF Vocabulary Description Language property `rdfs:isDefinedBy` is sometimes used to describe a relationship between a resource and an RDF/XML document describing the resource (a schema), but the registry data model introduces application-specific properties to describe the relationships between, for example, elements and element sets.

This means that, even where metadata element sets follow the simple "Dublin Core-like" model, and where their managers publish descriptions of those element sets using RDF Vocabulary Description Language, there are some limitations on how the registry processes those schemas. The registry does read and index the schemas published on the web by DCMI; however, the content of those schemas must be supplemented by additional RDF data created by the registry administrator to provide the application-specific metadata required by the registry.

Just as the CORES registry reuses schemas made available for general use, so the schemas created by the authoring tool may be saved and made available for use by other applications – though some of those applications may be programmed to process only some of the statements contained within those schemas.

At the heart of the application profile model (and indeed of the Semantic Web) is the principle that implementers will reference resources (in this case, use metadata element sets and application profiles) defined and published by others within a global space. Since this data is managed in a decentralised manner it will be impossible to guarantee integrity within this growing web of references. It is possible for an agency to register a metadata element and subsequently alter its semantics or delete it from their element set. A second agency may adopt the original metadata element and deploy it in an application profile, unaware of the changes. As an application operating on the aggregation of this distributed data, the registry can highlight that these relationships exist, but it cannot prevent these situations arising. This is a policy issue rather than a technical one, and it is good practice for the publishers of metadata element sets to make explicit policy statements about their use of URIs and the resources identified by those URIs [18].

## 4.2. Applicability of registry model

Firstly, user feedback suggests that some aspects of the current model require clarification, particularly the relationship of encoding schemes and datatypes and the extent to which an element usage "inherits" the attributes of the element it "uses".

The registry model is closely aligned with the Dublin Core model, where a metadata record is a simple "flat" set of attribute-value pairs and the values of metadata elements

are either literals or literals qualified by the name of an "encoding scheme". While this has proved a good basis for exploring and refining the concept of the application profile, it may be that this model is too restrictive to apply more generally.

Work is continuing to explore the application of the model to the IEEE Learning Object Metadata (LOM), building on the draft bindings for the LOM produced by the IEEE Learning Technology Standards Committee (LTSC) LOM RDF binding Working Group [19]. The LOM model is a hierarchical one, with elements grouped into categories, and it has typically been represented in an XML tree-structure. Work on the development of the RDF binding is highlighting issues of reconciling the "structural" and "conceptual" approaches [20].

A second area of complexity (which also arises in the case of the IEEE LOM) is that of schemas that deal with the description of multiple related resources of different types. For example, the Research Support Libraries Programme (RSLP) Collection Description schema provides for the description of a collection, the location of the collection and a number of agents related to those two entities [21]. In the registry model, this is perhaps most easily represented as three different application profiles, but the current model does not provide a mechanism for indicating a relationship between those three profiles.

## 4.3. Persistence

In discussing the persistence of the registry it is important to distinguish between

- The persistence of the data indexed by the registry – the schemas (RDF/XML documents) created by the schema creation tool or by other means and submitted to the registry for indexing;
- The availability of the registry service presently provided by the CORES project partners;
- The availability of the CORES registry software.

The CORES registry provides interfaces to the aggregated data content of the schemas submitted for indexing. However, the registry service should not be regarded as the only source of that data. The registry service is separate from that data. Agencies submitting data to the registry either have their data in the form of an RDF/XML document already, or have used the schema creation tool to prepare it. In the second case, schema creators should save their schema data in the form of an RDF/XML document and should manage and maintain the schemas they create.

At present the registry data model does not include the schema itself as an entity to be described, but consideration should be given to extending the model and the registry software to accommodate this, so that the data generated by the schema creation tool and submitted to the registry can include a pointer to the location of the schema on the Web.

With regard to the current registry service provided by the project, the CORES partners are committed to keep the

---

messages by Patrick Stickler, perhaps best summarised by the examples provided in <http://www.jiscmail.ac.uk/cgi-bin/wa.exe?A2=ind0301&L=dc-architecture&P=4782>

registry running for at least a year after the project's end in June 2003, though strictly speaking such a commitment is beyond the scope of the CORES project per se.

The source code for the registry software will be made available from SourceForge [22] under "open source" license conditions. (At present, it can be obtained on request from the project partners). When the CORES project service is terminated, another implementer can establish their own registry service. If schema creators manage their own schemas on the Web, then re-establishing the core functionality of the current project registry should require only reading and reindexing the content of those existing schemas from their known locations on the Web. However, data not maintained on a distributed basis (for example, the content of annotations) may not be available.

#### **4.4. Deployment in an XML world**

The CORES project, along with other Semantic Web applications, has to face the reality that at present many applications within the digital library world are based on XML rather than on RDF/XML. The project is faced with how best to deploy the registry within a partially Semantic Web.

Much metadata exchanged between applications is not encoded as RDF/XML. Instead, many metadata communities have developed their own conventions for expressing their metadata as tree-structured data represented in XML. Other remote XML applications processing that metadata must interpret those tree-structures as their creators intended, and that depends on the developers of the two applications sharing a common understanding of that tree-structure. There is no shared data model in XML: nothing in the XML specification determines the "meaning" of the elements and attributes in an XML document or of the structural relationships between those elements and attributes, and different designers make different design decisions about what their XML tree structures convey. Increasingly, metadata-based services and applications must handle metadata originating from an ever-increasing number of communities, each with their own different XML encoding conventions.

The use of XML Schema does not change this situation. An XML Schema describes and constrains the structure of a class of XML documents, and individual documents can be validated against an XML Schema to check that their structure conforms to the rules specified [23]. The information provided by an XML Schema and an "RDF schema" is fundamentally different and serves different purposes. An XML Schema describes only the components of an XML document (XML elements, XML attributes) and their structural relationships. The schemas read by the CORES registry, on the other hand, describe various types of "real world" resource: metadata elements and encoding schemes, element sets and application profiles, and the agencies that manage them.

Because the RDF/XML syntax specification [24] defines how a statement made using a metadata element to describe a resource should be represented in RDF/XML, the structure of an RDF/XML instance metadata record can be defined, working from the description of a metadata element set or application profile. However it is impossible to predict how an occurrence of that metadata element might be represented in a metadata record as part of an arbitrary XML tree structure. The registry cannot derive an XML Schema to represent such a tree-structure from the information provided in the RDF schemas submitted to the registry. However, the registry does allow an implementer to provide (as part of their description of an application profile) references to an existing XML Schema created separately and any supporting documentation.

In summary, while the machine-to-machine interfaces to the registry are designed primarily to support metadata applications that use the RDF model, the registry does also provide functionality of value to XML implementers. The developers of XML-based applications can use the registry's human-readable interface to explore metadata semantics and implementer usage; further, the registry contributes to the disclosure and discovery of XML Schemas made available by implementers. The registry does not presently collect metadata specifically about XML Schemas, but it could easily be extended to collect some basic descriptive metadata about those schemas if that was useful to implementers.

### **5. Future work**

There are various ideas for future work in this area.

#### **5.1. Data model and RDF vocabulary**

The registry data model and the registry RDF vocabulary should be reviewed in the light of the following recent developments:

- the introduction of support for literal datatyping in the RDF specifications [4],
- the publication of the Web Ontology Language (OWL) specifications [25]

Consideration also needs to be given to aligning the registry more closely with other work in the modelling and representation of ontologies.

#### **5.2. Collaborative schema creation and maintenance**

Registries serve not only to document the current state of metadata schemas, they also provide the means for implementers to learn from and compare various schemas, to inform each other about new schemas, and to support exchange of knowledge on metadata usage. Registries provide a focus for a community to share the creation of new schemas and ideally that becomes a truly collaborative process. Several schema creators may work together on the definition of a schema and use the experiences in other

schema creator groups. In the CORES project we tried to look at metadata registries as a place for collaborative work, and considered the registry service as a means to support co-operation on metadata creation, only parts of which have been implemented. A fully collaborative system would require complex authentication and editing not implemented within the project.

Joint preparation of schemas raises various issues such as:

- Working with versions
- Quality control
- Learning, testing, commenting schemas

Future work might include developing the registry application to serve the full lifecycle of metadata schemas and to support various collaborative processes.

For example, members of an agency may work together on a schema in the development phase. In this phase the schema is not revealed to the public, only those related to the agency may see the current phase (or older versions), and make modifications to it. When the schema is ready to be made accessible for a larger audience, creators publish it, in which case the service operators may judge the quality and appropriateness of the schema, and may accept or reject it. After this, the schema will be accessible to everybody and references to it will be shown in each relation (e.g. for used elements and encoding schemes).

The registry should include various data export features: a schema will be exportable in RDF format, or as HTML "documentation" giving definitions and annotations in a well-formatted way. It would be useful for registries to generate sample instances of RDF metadata records for a selected schema, and to use a form-generator for providing a fill-in form which helps the user to create their own sample records, thus implementing a simple DC-dot like feature.

### 5.3. Measuring deployment of schemas

An important aspect of the registry is to measure the amount and type of use for various schemas. Annotations may provide a place to collect comments on deployment. The first impression of usage level of element sets or encoding schemes may be given by the number of re-uses (i.e. element usage) by application profiles. A special type of annotation might be prepared to store deployment details in a machine processable way. The information collected in this way might contain number of records available in this format, location and availability of the metadata.

## 6. Conclusion: Bringing the registry to life

Feedback from the CORES workshop and from interested parties indicates that the provision of a human-readable Web interface for navigating schemas is a useful service. However the value of such a service depends on the registry being populated. Over the next few months the

CORES project will be encouraging schema creators, particularly managers of federated systems, to register their existing element sets and application profiles. It is hoped that this will encourage implementers of new systems to use the schema creation tool and the registry.

The project is aware that it is difficult to fully express hierarchical element sets (such as the IEEE LOM) using the Registry data model. However the project is targeting a particular, specialist audience, people who are creating schemas and have an interest in metadata interoperability, not a general audience involved in network technologies. We hope that by focusing on this audience we will be able to generate commitment to sharing information about metadata element sets, which in many cases have resulted from a significant commitment of time and effort.

It is important that users of the registry are able to make judgements on the authority of the registry and whether they can trust the contents. CORES will endeavour to encourage users by issuing a persistence policy, and emphasising that there needs to be a balance of responsibility between registry managers and those creating schemas.

The project has explored moving registries towards provision of machine-to-machine interfaces to enable more automated use whereby schemas can be downloaded to "drive" applications. Whilst acknowledging that there are many issues that will need to be addressed to achieve this, the potential benefits are significant. Some difficult decisions may need to be made in order to limit the scope of the registry in order to make such usage feasible in the shorter term. This might involve creating individual registries with particular data models that are a good fit for particular element set data models. It might involve provision of XML registries that record and serve mandated XML schemas, which sit alongside an RDF schema registry. The project hopes that work in related areas, in particular the development of models and tools for expressing ontologies based on the Ontology Web Language, will benefit ambitions for managing metadata schemas. We intend to collaborate with related efforts in order to move forward the registry from a tool for human users to a Web Service.

## Acknowledgements

CORES is funded under the European Union's Information Society Technologies (IST) Programme. The project is a joint activity involving contributions from all project partners, which, as well as the author's affiliated institutions, include Fraunhofer Gesellschaft and PricewaterhouseCoopers Luxembourg. The authors wish to acknowledge the partners' input to all of the work described in this review, in particular we wish to note contributions to this work from Tom Baker and Makx Dekkers.

We would also like to acknowledge the contributions from partners in the MEG registry project, Dave Beckett

and Damian Steer, ILRT, University of Bristol, and SCHEMAS colleague, Manjula Patel, UKOLN.

## References

- [1] *CORES: a forum on shared metadata vocabularies* (2003). Retrieved May 13, 2003, from <http://www.cores-eu.net/>
- [2] Baker, T., Blanchi, C. et al. (2003). *Principles of Metadata Registries*. White Paper of DELOS Registries Working Group. Retrieved July 21, 2003, from <http://delos-noe.iei.pi.cnr.it/activities/standardizationforum/Registries.pdf>
- [3] Heery, R. and Patel, M. (2000, September). Application Profiles: mixing and matching metadata schemas. *Ariadne* 25. Retrieved May 13, 2003, from <http://www.ariadne.ac.uk/issue25/app-profiles/>
- [4] Manola, F., and Miller, E., Editors. (2003). *RDF Primer. World Wide Web Consortium W3C Working Draft, 23 January 2003*. Retrieved May 13, 2003, from <http://www.w3.org/TR/2003/WD-rdf-primer-20030123/>
- [5] Brickley, D. and Guha, R.V., Editors, (2003). *RDF Vocabulary Description Language 1.0: RDF Schema. World Wide Web Consortium W3C Working Draft, 23 January 2003*. Retrieved May 13, 2003, from <http://www.w3.org/TR/2003/WD-rdf-schema-20030123/>
- [6] Baker, T. and Dekkers, M., Editors. (2002) *CORES Standards Interoperability Forum: Resolution on Metadata Identifiers*. Retrieved May 13, 2003, from <http://www.cores-eu.net/interoperability/cores-resolution/cores-resolution.pdf>
- [7] *CORES Registry* (2002-2003). Retrieved May 13, 2003, from <http://cores.dsd.sztaki.hu/>
- [8] *Metadata for Education Group (MEG) registry project*. (2002). Retrieved May 13, 2003, from <http://www.ukoln.ac.uk/metadata/education/regproj/>
- [9] Heery, R., Gardner, T., Day, M., & Patel, M. (2000) *DESIRE metadata registry framework*. Retrieved May 13, 2003, from <http://www.desire.org/html/research/deliverables/D3.5/>
- [10] *The SCHEMAS Forum*. (2000-2001). Retrieved May 13, 2003, from <http://www.schemas-forum.org/>
- [11] DCMI Usage Board. (2003). *DCMI Grammatical Principles*. Retrieved May 13, 2003, from <http://dublincore.org/usage/documents/principles/>
- [12] Guenther, R. (2002). *Library Application Profile*, DCMI Working Draft. Retrieved May 13, 2003, from <http://dublincore.org/documents/2002/09/24/library-application-profile/>
- [13] Heery, R., Johnston, P., Beckett, D. and Steer, D. (2002) The MEG Registry and SCART: complementary tools for creation, discovery and re-use of metadata schemas. *Proceedings of the International Conference on Dublin Core and Metadata for e-Communities, 2002*. Retrieved May 13, 2003, from <http://www.bncf.net/dc2002/program/ft/paper14.pdf>
- [14] Beckett, D. (2003). *Redland RDF Application Framework*. Retrieved May 13, 2003, from <http://www.redland.opensource.ac.uk/>
- [15] World Wide Web Consortium. (2001-2003). *Annotea Project*. Retrieved May 13, 2003, from <http://www.w3.org/2001/Annotea/>
- [16] Hewlett-Packard Labs. *Jena Semantic Web Toolkit*. Retrieved May 13, 2003, from <http://www.hpl.hp.com/semweb/jena.htm>
- [17] *CORES Schema Creation & Registration Workshop SZTAKI, Budapest, 6-7 March 2003* (2003). Retrieved May 13, 2003, from <http://www.cores-eu.net/>
- [18] Powell, A and Wagner, H., Editors. (2001). *Namespace Policy for the Dublin Core Metadata Initiative (DCMI)*. DCMI Recommendation. Retrieved May 13, 2003, from <http://dublincore.org/usage/documents/dcmi-namespace/>
- [19] Nilsson, M., Editor. (2002) *IEEE Learning Object Metadata RDF Binding*. Working Draft, 26 August 2002. Retrieved May 13, 2003, from <http://kmr.nada.kth.se/el/ims/md-lomrdf.html>
- [20] Nilsson, M. (2003). *Semantic Issues with the LOM RDF Binding*. 15 January 2003. Retrieved May 13, 2003, from <http://kmr.nada.kth.se/el/ims/md-lom-semantics.html>
- [21] Powell, A. (2000). *RSLP Collection Description: Collection Description Schema*. Retrieved May 13, 2003, from <http://www.ukoln.ac.uk/metadata/rsrp/schema/>
- [22] *SourceForge*. (n.d.) Retrieved May 13, 2003, from <http://sourceforge.net/>
- [23] Fallside D., Editor. (2001). *XML Schema Part 0: Primer. World Wide Web Consortium W3C Recommendation, 2 May 2001*. Retrieved May 13, 2003, from <http://www.w3.org/TR/xmlschema-0/>
- [24] Beckett, D., Editor. (2003). *RDF/XML Syntax Specification (Revised). World Wide Web Consortium W3C Working Draft, 23 January 2003*. Retrieved May 13, 2003, from <http://www.w3.org/TR/2003/WD-rdf-syntax-grammar-20030123/>
- [25] McGuinness, D., and van Harmelen, F., Editors. (2003). *OWL Web Ontology Language Overview. World Wide Web Consortium W3C Working Draft, 31 March 2003*. Retrieved May 13, 2003, from <http://www.w3.org/TR/2003/WD-owl-features-20030331/>