Proc. Int. Conf. on Dublin Core and Metadata for e-Communities 2002: 125-132
© Firenze University Press

The MEG Registry and SCART: Complementary Tools for Creation, Discovery and Re-use of Metadata Schemas

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

Dave Beckett, Damian Steer
ILRT, University of Bristol
dave.beckett@ilrt.bris.ac.uk, pldms@mac.com

Abstract

SCART is an RDF schema creation tool designed for use by implementers working within digital library and learning environments. This schema creation and registration tool is being developed to work in conjunction with registry software. SCART will provide implementers with a simple tool to declare their schemas, including local usage and adaptations, in a machine understandable way based on the RDF Schema specification. This tool is optimised for use by the Metadata for Education Group, projects and services within the UK providing resource discovery in the domain of education. By providing a complementary creation tool and registry the aim is to facilitate easy discovery of existing schemas already registered in a schemas registry, and to enable implementers to re-use these existing schemas where appropriate.

1. Introduction

The Metadata for Education Group (MEG) Registry and Schema Creation tools aim to provide implementers of educational systems with the means to share information about their metadata schemas and to re-use existing schemas. MEG is a loose confederation of educational organisations concerned with the description and provision of educational resources at all educational levels across the United Kingdom [1]. Currently there are over sixty members of the MEG group with approximately twenty known to be active in creating schemas to describe educational resources. The existence of such a focused group offers great potential for sharing and collaboration regarding design and re-use of schemas. Facilitating 'declaring and sharing' schemas in use by members of the group will benefit their system designers, and their funders, by saving the time and effort currently spent in researching existing schemas and in re-inventing schemas.

The MEG registry work builds on previous activity within the DESIRE and SCHEMAS projects [2, 3] which established data models for declaring schemas and local usage within a schema registry, and implemented prototype registries [4, 5, 6]. The MEG registry development is based on the DESIRE data model, but is a completely new implementation which seeks to address some of the problems of sustainability and scalability encountered with the DESIRE approach. The DESIRE registry was implemented as a relational (MySQL) database. The aim within the MEG development is to explore the benefits, and drawbacks, of implementation using specifications and tools emerging from the Resource Discovery Framework (RDF). The status of the development is work in progress, with a final delivery date planned for September 2002. Prototypes are now available which can be accessed and used for demonstration, with draft schemas loaded into the Registry [7].

A schema creation and registration tool (SCART) is being developed to work in conjunction with the new MEG Registry software. The SCART will enable implementers to declare their schemas, including local usage and adaptations, in a machine understandable way. This tool is optimised for use by projects and services providing resource discovery in the domain of education, whether for discovery of information or learning objects. By providing a complementary creation tool and registry the aim is to facilitate easy discovery of existing schemas already registered in a schemas registry, and to enable implementers to re-use these existing schemas where appropriate, whilst creating new usages where required. The SCART is designed to interact with the

MEG Registry, so both are being developed in parallel.

The MEG registry will provide browsing and searching facilities for all data elements contained within the registered schemas, and schemas can be entered or updated by use of the SCART. Experience from previous prototypes (DESIRE, SCHEMAS) has established that it is vital for implementers to have this facility for entry and update under their local control: it is not scalable in terms of effort to manually enter schemas centrally. This imposes the requirement that local implementers construct their schemas in a well-formed manner, and the SCART is designed with this in mind. An additional benefit is that once a well-formed schema has been produced locally it can be used for local applications, and provided to (or gathered by) other registry-like applications. That is, the SCART and the MEG registry are loosely coupled, in that schemas created using the SCART may be used by other applications and schemas may be prepared using other tools and submitted to the registry.

In order to follow existing recognised standards as far as possible, and to take advantage of emerging open source tools the MEG developments have been based on the RDF Schema specification [8, 9]. RDF provides a common data model which is particularly appropriate for exchange of data with unknown semantics by enabling common naming and identification of data.

This paper will provide some context to the MEG developments, then go on to describe the data model used within the SCART and Registry. Lastly some brief detail will be given of the design and features provided within the SCART, while keeping in mind that this is at present in prototype, with delivery scheduled for September 2002.

2. Background

A short overview of activity within the Semantic Web regarding vocabulary sharing will give some context to the MEG registry work, as this has informed development of the SCART and MEG registry. The vision of the Semantic Web is built on an infrastructure of interoperable metadata, where software can infer semantic meaning to ‘unknown’ metadata, albeit with an acceptance that the understanding of these semantics may be partial. The Semantic Web envisages software being able to treat the Web as ‘a global database’ [10] where data (and metadata) can be fetched and manipulated. In order to achieve this, there needs to be a way for metadata to be exchanged both at syntactical and semantic levels.

The Resource Description Framework (RDF) provides a common data model for making statements about resources, and a means of expressing those statements in a common syntax (XML). This combination of model and syntax means that independently created statements describing the same resource can be shared and ‘merged’. Such data aggregation offers a powerful means to re-use existing (RDF/XML) metadata that resides on the Web or in accessible RDF compliant databases. RDF provides a data model. However there still needs to be consensus on identification and naming to enable shared use of metadata, both for the naming and meaning of ‘properties and classes’ (data elements), and for identification of the resources being described. The importance of naming has been acknowledged by the TAP project [11] and it is a complex area which will take time to solve. However we see deployment of schema creation tools and schema registries as a means to assist in reaching a common approach to naming data elements, which can then be shared by re-use.

Within the digital library and wider Semantic Web community there has been some exploration of options for declaring and sharing metadata schemas. Various approaches to establishing common vocabularies have been put forward:

Relying on dominant market forces, whereby the core vocabulary of e-commerce will prevail [11].

Mapping between core data elements whether mapping of data elements [12] or more complex ontology mapping [13].

Enabling ‘base-line’ interoperability by agreement on a minimal metadata element set, whereby heterogeneous metadata is normalised to a minimum common data element set [14].

These various options may be seen as transitory stages to a more sophisticated solution, or the solution may be a more complex amalgamation of all approaches. However it is clear that all of these approaches require flexibility and extensibility, which in turn will require manipulation and mapping of different vocabularies. Registries are seen as providing such services. Within the digital library community a first step might be to enable declaration of vocabularies in use, in particular to publish and share vocabularies. For some time metadata registries have been seen as a way to do this, to encourage re-use where appropriate:

> *... registries will need to be managed, coordinated, and ultimately connected. Registries will define the elements of metadata schemas in a machine-readable syntax (e.g., RDF) and offer authoritative listings of legal values, local extensions, mappings to other schemas, and guidelines for good usage. They will serve both humans, with readable text, and programs, with structured content that can automatically be parsed. Their role will be both to promote and to inform, thereby encouraging the use of standard formats. [15]*

Acknowledging the need to reach consensus on data elements that can be used in common, it seems likely that this might best be achieved incrementally by agreeing common vocabularies amongst shared ‘communities of interest’.

The standard vocabulary need not all come from one source. We provide a core kernel of terms, which can be extended by communities and their applications. Some of the terms defined by the applications will over a period of time get absorbed into the kernel. The use of XML namespace qualifications allows data providers to define their own extensions. The evolution of this ecology of names will likely mirror the evolution of operating system

APIs, wherein, over a period of time, the operating systems incorporated the APIs offered by the more successful platform-like applications running on the operating systems [11].

3. The role of the Registry

The functionality of particular metadata registry implementations differ, but the overall role can be encapsulated as facilitating extensibility and interoperability in the context of networked services. Within the digital library community there have been a variety of approaches, so for example some registries have been human readable only, whereas others machine-readable. Some registries offer descriptive information of element sets whereas others provide search and browse access to data elements. In some cases, those elements are from schemas owned by a single 'registration agent'; in other cases, they are drawn from many schemas from various sources. Some registries might allow the interpretation of different metadata element sets by means of crosswalks, mappings or translations. Registries might provide a service to a specific community of practice, such as the education domain, or the museum sector, or might be focused on a group of implementations with a common business model. A brief overview of registry activity is included in a recent account of the DCMI Registry activity [16]. The DCMI Registry is an example of a standards making body providing information about its element sets by means of a registry service [17].

The MEG Registry is a formal system that discloses authoritative information about the semantics and structure of the data elements within the registered schemas. The MEG registry uses schemas primarily to provide a descriptive or documentary function, and it should be noted that the validation of instance data against schemas is not part of that function. The MEG Registry will provide information about its contents to both humans and software. This means the information within the registry needs to be stored in a syntax that is machine-readable as well as in human readable form. Users of the Registry System would typically be implementers seeking appropriate schemas, developers comparing schemas, publishers of standards, and metadata creators seeking assistance in using particular schemas correctly.

Usage scenarios for the MEG Registry System include:

Publishing a description of an Element Set: A UK organisation provides a resource discovery service for Web-based educational materials that utilises a simple metadata schema developed specifically for that purpose. The organisation wishes to publish this information to the MEG community via the registry. Using the SCART the Element set publisher can create an Element Set description, and add descriptions for each Element. Encoding Schemes can be added to Elements. On completion the Element Set can be saved locally and submitted to the Registry

Publishing a description of an Application Profile A UK organisation provides a resource discovery service for Web-based educational materials. That service utilises a simple metadata schema that uses a number of Elements drawn from the cross-domain Element Sets of the Dublin Core Metadata Initiative; a domain-specific Element which was created by another portal service for their own schema; and a number of new Elements specific to this service. The organisation has developed a number of controlled vocabularies for several of the Elements in this schema; the service also specifies the use of some standardised forms for dates and identifiers within metadata instances. The organisation wishes to publish this information to the MEG community via the registry. In the terms of the registry data model, this organisation's schema is an Application Profile.

Indexing a standard schema for an Element Set An international standards body makes schema for their cross-domain Element Set available in RDF/XML on their Web server. Various MEG members wish to 'use' Elements from the Element Set in their Application Profiles. Either the representative of standards body or the registry administrator can use SCART to add the schema to the Registry.

Exploring Element Usage A schema developer wishes to survey the usage of the DCMI 'audience' element, and particularly the use of any controlled vocabularies to control values of this element.

4. The MEG Registry data model

Underpinning the data model for the MEG registry is the recognition that implementers deploy and adapt 'standard' metadata Element Sets in a pragmatic way. While 'standard' schemas are widely available, use-oriented adaptations, which are often localised and service-specific, tend to be less visible. Researchers on schema usage have introduced the idea of the 'application profile' as a means of capturing this information on adaptations and constraints of Element Set usage [18].

The data model for the MEG registry is designed to support the description of the following classes of entity and the relationships between instances of those classes. Descriptions of all instances include a unique identifier (or token) - for use (primarily at least) by software tools - and a label or title for the human reader.

Elements: the formally defined terms which are used to describe attributes of a resource. The description of an

Element must include a unique identifier, a name or label, and a description of its meaning. It may include information on the usage of the Element and information on the relationship between this Element and semantically-related Elements.

Element Sets: sets of functionally-related Elements which are defined and managed as a unit. The description of an Element Set must include a unique identifier, a title, a textual description of its intended scope/area of use, and the name (URI) of an XML Namespace associated with the Element Set. It may include version information, a classification of the Element Set and references to descriptions of the Element Set.

Usages of Elements: in the context of particular applications. The description of an Element Usage must include a unique identifier and the identifier of an Element. It may include:

- a new name or label for the Element in this application context;
- a description of the meaning of the Element in this context (the Element Usage may refine the definition of an Element to make it narrower or more specific to an application context);
- a description of the obligation to use the Element, and/or any constraints on its occurrence, in this application context;
- a description of constraints on the value of the Element in this application context, either as data type specifications or more narrowly through association with one or more Encoding Schemes (see below).

Application Profiles: sets of functionally-related Element Usages, created for the purpose of a particular function or application and managed as a unit. An Application Profile may include Element Usages of Elements from one or more Element Sets. The description of an Application Profile must include a unique identifier, a title and a textual description of its intended scope/area of use. It may include version information, a classification of the Application Profile, references to external descriptions of the Application Profile, and references to XML Schema based on the Application Profile.

Encoding Schemes: mechanisms that constrain the value space of Elements. The description of an Encoding Scheme must include a unique identifier, a name or label and a textual description of its intended use. It may include version information, a classification of the Encoding Scheme, and references to external descriptions of the Encoding Scheme. Encoding Schemes may be of two types:

- a Scheme which enumerates a list of permitted Values: the list of Values may be recorded by the registry (see below);
- a Scheme which specifies a set of rules that define or describe permitted values: such rules cannot be captured by the registry, but can be indicated by a reference to an external description of the Encoding Scheme.

Values: the individual Values which an Encoding Scheme enumerates may be recorded. The description of a Value must include a unique identifier and a label. It may include a textual description providing more information about the Value. The practicality of recording Values within the MEG Registry may depend on the size of the "vocabulary" and whether or not it already exists in a suitable machine-processable form.

Agencies: persons or organisations responsible for the ownership or management of Element Sets, Application Profiles and Encoding Schemes. The description of an Agency must include a unique identifier and a name; it may include a reference to an external source of further information.

The principal relationships between entities are represented graphically in Figure 1, with an indication of whether the relationship is many (m) to one (1).

The main points to note on the relationships between entities in this diagram are:

- Each Element Set, Application Profile and Encoding Scheme must be associated with exactly one Agency responsible for its maintenance; an Agency may be responsible for multiple Element Sets, Application Profiles and Encoding Schemes;
- An Element Set contains multiple Elements; and an Element must be a member of exactly one Element Set;
- An Element may be associated with multiple Encoding Schemes which specify constraints on its value; and an Encoding Scheme may be associated with multiple Elements;

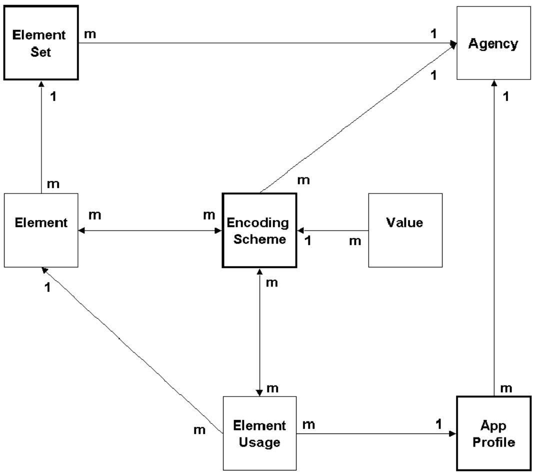


Figure 1. Registry data model

- An Element may be described as a semantic refinement of a second Element; and an Element may have multiple refinements;
- An Application Profile contains multiple Element Usages; and an Element Usage must be a member of exactly one Application Profile;
- An Element Usage must use exactly one Element; but an Element may be the object of multiple Element Usages;
- An Element Usage may specify the use of multiple Encoding Schemes; and an Encoding Scheme may be deployed in multiple Element Usages.

This model is based on that used by the DESIRE metadata schema registry [4]. The MEG registry also builds on the experience of the SCHEMAS project which suggested conventions for describing Application Profiles in machine-processable form using the RDF model [19, 20]. In particular, the MEG registry adopts the suggestion by the SCHEMAS project that the entities described here as Element Usages might usefully be modelled as resources, and the RDF vocabulary used by the MEG registry defines a class "reg:ElementUsage" for this purpose. Elements are modelled as resources of type rdf:Property.

Figure 2 is a graphical representation of how an "Element" and an "Element Usage" might be described using this data model.

The lower part of this diagram represents the description of the Element with the identifier http://purl.org/dc/elements/1.1/title, which is part of an Element Set defined by the Dublin Core Metadata Initiative, i.e. the element often referred to by the XML qualified name "dc:title". DCMI assigns this Element a name or label, the string "Title", and provide a definition of the Element as the string " A name given to the resource". This is represented in the diagram by labelled arcs linking a node representing the Element to two separate nodes representing these strings. A fourth linked node makes explicit the "type" of this resource. So the lower part of the diagram represents the statements:

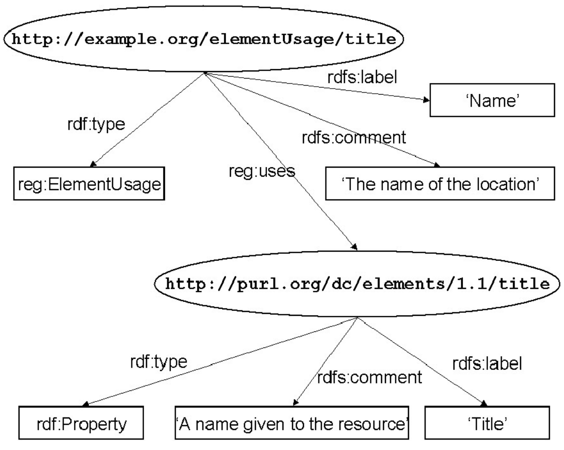

http://example.org/elementUsage/title
rdfs:label
'Name'
rdf:type
rdfs:comment
reg:ElementUsage
reg:uses
'The name of the location'
http://purl.org/dc/elements/1.1/title
rdf:type
rdfs:comment
rdfs:label
rdf:Property
'A name given to the resource'
'Title'

Figure 2. Using an Element

- The resource http://purl.org/dc/elements/1.1/title is of type rdf:Property;
- The resource http://purl.org/dc/elements/1.1/title has a label, "Title";
- The resource http://purl.org/dc/elements/1.1/title has a definition, "A name given to the resource";

The upper part of the diagram represents the description of a Usage of this same Element in the context of a particular application. An implementer has chosen to adopt the "dc:title" Element but to modify the human-readable label of the Element and also to refine the semantics of the Element to make them more specific to the context of the application.

In the MEG registry model, the Element Usage is represented as a second resource - a separate node in the diagram. The relationship between the Usage and the Element is represented by the arc between the two nodes, and that arc is labelled to identify the nature of that relationship. The additional arcs and nodes represent the application-specific label and definition which the Element Usage prescribes for the Element http://purl.org/dc/elements/1.1/title. And a final linked node makes explicit the type of the Element Usage resource. So there are now four additional statements:

- The resource http://example.org/elementUsage/title is of type reg:ElementUsage;
- The resource http://example.org/elementUsage/title "uses" the resource http://purl.org/dc/elements/1.1/title;
- The resource http://example.org/elementUsage/title has a label, "Name";
- The resource http://example.org/elementUsage/title has a definition, "The name of the location".

A more complex example of an Element Usage might introduce constraints on the value of the Element in the context of the application by mandating the use of specified Encoding Schemes. These would be represented by additional nodes linked to the Element Usage node, i.e. additional statements about the Element Usage, of the form:

- The resource http://example.org/elementUsage/date specifies use of the resource http://purl.org/dc/terms/W3CDTF (which is of type reg:EncodingScheme).

Further, to simplify the diagram above, some of the relationships that would be mandatory within the context of the registry - and would be enforced by the schema creation tool - are not illustrated here. e.g. the Element node would have a further arc to a node representing a resource of type "Element Set" and the Element Usage node would have a further arc to a node representing a resource of type "Application Profile".

Some points to note include firstly, that the RDF vocabulary for the MEG registry defines a property "reg:uses" to express the relationship between an Element Usage and an Element. The data model specifies that the value of a “reg:uses” property must be a resource of type Element. The value of a “reg:uses” property cannot be a resource of type Element Usage. The schema creation tool enforces this constraint. Element Usages are, however, assigned URIs and this does open up the possibility that, in a distributed environment, the creators of new Application Profiles might make their own statements about Element Usages previously created by others. The SCHEMAS project noted that such a possibility risks “semantic drift”, but also that it may be difficult to avoid in the decentralised context of the Web [20].

Secondly, the “Element Usage” class defined in the MEG registry vocabulary is not defined as a sub-class of the “rdf:Property” class i.e. Element Usages are not RDF properties and resources of this type cannot be used as predicates in RDF statements. So, where an Application Profile specifies an Element Usage of the Element http://purl.org/dc/elements/1.1/title, the creator of instance metadata conforming to this Application Profile continues to use the term http://purl.org/dc/elements/1.1/title as the predicate in their RDF statements.

Thirdly, an Element Usage can specify a narrowing of the definition of an Element in a “standard” Element Set; an implementer might achieve a similar result by defining a new Element (in a “local” Element Set) and specifying that this Element is an “element refinement” of a standard Element. It may be useful to explore further the advantages and disadvantages of the two approaches.

5. SCART software development

The main capabilities required by the client software (SCART) were to:

Create and edit application profiles, element sets and encoding schemes.

Encourage re-use of existing elements and encoding schemes.

Allow submissions to a remote registry server.

The client and remote registry were required to use a common data model based on RDF Schema specification. The creation of RDF data is clearly impractical for non-experts, so the client needed to simplify the process greatly. Happily the data model is relatively simple. Encouraging reuse was the second major issue. The natural solution is to ensure that it is easier to find suitable pre-existing items than it is to create new ones. The MEG Registry will, when in production, convieniently provide a store of existing elements and encoding schemes. In the future it is expected other compatible schema registries will become available too. So SCART provides the user with the facility to search remote registries, and results are presented so that users can make informed decisions as to whether found items are appropriate for their application profiles.

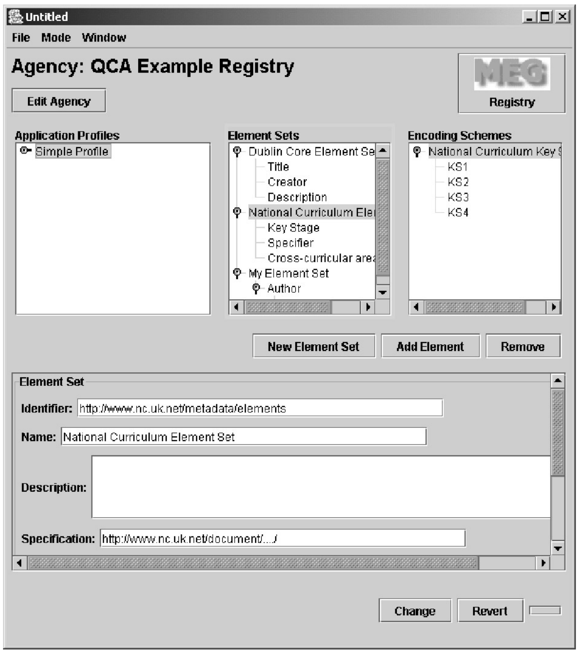
Existing software offerings were considered as a basis for the client. Currently there are few options for RDF authoring and three applications were investigated: RDFAuthor [21], IsaViz [22], and Protégé [23]. RDFAuthor and IsaViz are similar applications, presenting a graphical representation of the RDF data model, that is a graph structure with nodes. For the purposes of SCART neither application was thought suitable. Both probably could have been augmented to talk to registries, however although they hide the syntax of RDF from the user neither hides the graph model. Using either tool would require a familiarity with the registry data model that is unreasonable to expect in the target audience.

Protégé is quite different in scope and intention. It is a complex tool allowing users to create ontologies, use these ontologies to create interfaces for entering instance data, and query that data. Protégé includes plugins for the DAML [24] and RDF Schema vocabularies to describe the ontology. The latter was particularly interesting since the registry uses RDF Schema. Protégé showed promise for schema creation; and submission to a remote registry probably could have been accomplished with an add-on tool; however re-use of existing data elements would have been problematic. In addition Protégé’s notion of an ontology is a great deal stronger than required by the registry and the redundant functionality would result in a confusing interface for the target audience. In summary, Protégé, IsaViz, and RDFAuthor essentially are general purpose tools, and whilst suitable for the tasks for which they are designed, the MEG project needed a tool tailored for its intended audience.

SCART therefore was custom built for the project, using Java, which had a clear advantage due to its multi-platform nature. SCART is known to run on Windows 2000, Mac OS X and Linux. The promise of Java is that it will run on other platforms (e.g. other Windows versions, BSD). It should be noted that the user interface toolkit in Java (sometimes called ‘Swing’) has revealed some quirks, but generally proved useful, as did the RDF toolkit for Java, Jena [25]. A walk through of the functionality of the SCART is available on the MEG Web pages showing in detail the process for creating application profiles, schemas and encoding schemes [7]. Here only a brief mention of some features is possible.

SCART supports multiple documents, each a self contained window associated with a Registration Agency. When a new document is created the user has to supply a name for the agency, and an identifier (a URI). The identifier is particularly important due to the nature of RDF. However this is the only time the user is required give an object an identifier as a default identifier will be created automatically if left unspecified by the schema creator.

A ‘Search Registry’ window provides the re-use incentive to a schema creator. It provides an interface to ‘external’ elements and encoding schemes whether these are located at a remote registry, or in a local file



Identifier: http://www.nc.uk.net/metadata/elements

Name: National Curriculum Element Set

Description:

Specification: http://www.nc.uk.net/document/..../

Figure 3. Creating an application profile with the SCART

allowing offline use. Searching is carried out using the HTTP protocol and, of course, other registries searched need to be structured according to a common data model. Users can perform simple keyword searches to find elements and/or encoding schemes. Results are displayed so they can be dragged and dropped into the application profile window.

Drag and drop is used extensively throughout SCART. Encoding schemes and elements can be dragged to any place where their inclusion makes sense. In this way the user can create application profiles made up of existing data elements.

Separate windows are provided for the main tasks. One for creating application profiles and element usages, one for element sets, and one for encoding schemes. The most typical use of the tool within the MEG context will be for creating application profiles so the default view in the window is of the 'application profile', however when required two further views to the document window are made available for creating elements sets and encoding schemes.

Resulting schema can be saved locally (in RDF/XML), then submitted to the registry, or reloaded to the SCART later for further editing. On completion the schema can also be made available for harvesting by other applications, with the advantage that other agencies can reuse the data.

6. Conclusions

The primary purpose for describing Application Profiles and Element Usages is to provide a means by which schema implementers can disclose information about service-specific or application-specific practice in using metadata schemas. The information serves primarily a 'documentary' function [26, 20]. The ability to express this information in the form of machine-processable schemas facilitates the exchange and reuse of that information. This suggests that it is possible for metadata schema registries to broaden their scope to index and publish not only the descriptions of Element Sets provided by 'standards bodies', but also information on the local, 'real world' experience of implementing those Element Sets.

Information on implementation forms a larger and more rapidly changing set of data than the descriptions of the relatively static standard Element Sets. The effective capture of this information by services such as schema registries will depend on the provision of tools appropriate to the distributed and decentralised nature of the environment. The development of the schema creation tool seeks to address that challenge.

Information on implementation is also potentially much less uniform than the description of standard Element Sets. The data model outlined here has emerged primarily from the experience of working with the Dublin Core Metadata Element Set, which is a small, simple Element Set. We believe that the principles on which the work is based are extensible to other more complex schemas, but that hypothesis remains to be tested. The provision of a registry for the MEG community, with a number of implementers of schemas based on the IEEE Learning Object Metadata Specification and the IMS Learning Resource Meta-data Specification, will provide an opportunity to do so.

Acknowledgements

This work was made possible through funding from the JISC and BECTa The authors would like to acknowledge Tom Baker and Manjula Patel for their contribution to discussions of registries and schema creation tools within the context of previous projects.

References

[1] MEG, *Metadata for Education Group*, 2002. Available from: <http://www.ukoln.ac.uk/metadata/education/> (Last visited: June 2002).

[2] DESIRE, *Desire Project*. 2000. Available from: <http://www.desire.org/> (Last visited June 2002).

[3] SCHEMAS Forum, *The SCHEMAS Forum*, 2001. Available from: <http://www.schemas-forum.org/> (Last visited: June 2002).

[4] Heery, R., Gardner, T., Day, M., & Patel, M.,

DESIRE metadata registry framework, 2000. Available from <http://www.desire.org/html/research/deliverables/D3.5/> (Last visited June 2002).

[5] DESIRE, *Desire Metadata Schema Registry*, 2000. Available from: <http://desire.ukoln.ac.uk/registry/> (Last visited June 2002).

[6] SCHEMAS Forum, *The SCHEMAS Forum Registry,* 2001. Available from: <http://www.schemas-forum.org/registry/> (Last visited: June 2002).

[7] MEG, *MEG registry project*, 2002. Available from: <http://www.ukoln.ac.uk/metadata/education/regproj/> (Last visited: June 2002).

[8] Brickley, D. and Guha, R.V. Editors, *Resource Description Framework (RDF) Schema Specification 1.0. World Wide Web Consortium W3C Candidate Recommendation, 27 March 2000,* W3C, 2000. Available from <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/> (Last visited June 2002).

[9] Brickley, D. and Guha, R.V. Editors, *RDF Vocabulary Description Language 1.0: RDF Schema. World Wide Web Consortium W3C Working Draft, 30 April 2002.* Work in progress, W3C, 2002. Available from <http://www.w3.org/TR/2002/WD-rdf-schema-20020430/> (Last visited June 2002).

[10] Berners-Lee, T., *Semantic Web Road map*, 1998. Available from: <http://www.w3.org/DesignIssues/Semantic.html> (Last visited 22 June 2002).

[11] Guha, R.V. and McCool, R., *A System for integrating Web Services into a Global Knowledge Base,* 2002. Available from <http://tap.stanford.edu/sw002.html> (Last visited June 2002).

[12] Blanchi, C. and Petrone, J., Distributed Interoperable Metadata Registry. *D-Lib Magazine,* Volume 7 Number 12 (Dec 2001). Available from: <http://www.dlib.org/dlib/december01/blanchi/12blanchi.html> (Last visited: June 2002).

[13] Lagoze, C. and Hunter, J., The ABC Ontology and Model. *Journal of Digital Information*, Volume 2 Issue 2 (Nov 2001). Available from: <http://jodi.ecs.soton.ac.uk/Articles/v02/i02/Lagoze/> (Last visited: June 2002).

[14] Lagoze, C., Van de Sompel, H., Nelson, M. and Warner, S. Editors, *Open Archives Initiative Protocol for Metadata Harvesting, Protocol Version 2.0 of 2002-06-14*. OAI, 2002. Available from: <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm> (Last visited: June 2002).

[15] EU-NSF Working Group on Metadata, *Metadata for digital libraries: a research agenda*. Draft 10. Le Chesnay: ERCIM, 1998. Available from: <http://www.ercim.org/publication/ws-proceedings/EU-NSF/metadata.html> (Last visited: June 2002)

[16] Heery, R. and Wagner, H., A Metadata Registry for the Semantic Web. *D-Lib Magazine,* Volume 8 Number 5 (May 2002). Available from: <http://www.dlib.org/dlib/may02/wagner/05wagner.html> (Last visited: June 2002).

[17] DCMI Registry Working Group, *The Dublin Core Metadata Registry*. Prototype, 2002. Available from: <http://wip.dublincore.org:8080/dcregistry/index.html> (Last visited: June 2002).

[18] Heery, R. and Patel, M., Application Profiles: mixing and matching metadata schemas. *Ariadne* 25 (Sep 2000). Available from <http://www.ariadne.ac.uk/issue25/app-profiles/> (Last visited June 2002).

[19] SCHEMAS Forum, *The SCHEMAS Forum Registry: sample RDF encodings of Application Profiles*, 2002. Available from: <http://www.schemas-forum.org/registry/schemas/> (Last visited: June 2002).

[20] Baker, T., Dekkers, M., Heery, R., Patel, M. and Salokhe, G., What terms does your metadata use? Application profiles as machine-understandable narratives. *Journal of Digital Information* Volume 2 Issue 2 (Nov 2001). Available from: <http://jodi.ecs.soton.ac.uk/Articles/v02/i02/Baker/> (Last visited: June 2002).

[21] Steer, D., *RDF Author,*. 2002. Available from: <http://rdfweb.org/people/damian/RDFAuthor/> (Last visited: June 2002).

[22] Pitriga, E., *IsaViz: a visual authoring tool for RDF*, 2002. Available from: <http://www.w3.org/2001/11/IsaViz/> (Last visited: June 2002).

[23] *Protégé*, 2002. Available from: <http://protege.stanford.edu> (Last visited: June 2002).

[24] DAML, *DARPA agent markup language homepage*, 2002. Available from: <http://www.daml.org/> (Last visited: June 2002).

[25] Hewlett Packard Labs, *HPL Semantic Web activity*, 2002. Available from: <http://www.hpl.hp.com/semweb/jena-top.html> (Last visited: June 2002).

[26] SCHEMAS Forum, *The SCHEMAS Forum: A Retrospective Glossary*, 2001. Available from: <http://www.schemas-forum.org/info-services/d74.htm> (Last visited: June 2002).