A Configurable RDF Editor for Australian Curriculum

Diny Golder, Les Kneebone, Jon Phipps, Steve Sunter and Stuart A. Sutton

JES & Co., Tucson AZ, U.S.A.
{dinyg, jonp}@jesandco.org
Curriculum Corporation, Melbourne, Australia
{ les.kneebone, steve.sunter}@curriculum.edu.au
University of Washington, Seattle, WA, U.S.A.
sasutton@uw.edu

Abstract. Representing Australian Curriculum for education in a form amenable to the Semantic Web and conforming to the Achievement Standards Network (ASN) schema required a new RDF instance data editor for describing bounded graphs—what the Dublin Core Metadata Initiative calls a 'description set'. Developed using a 'describe and relate' metaphor, the editor reported here eliminates all need for authors of graphs to understand RDF or other Semantic Web formalisms. The Description Set Editor (ASN DSE) is configurable by means of a Description Set Profile (DSP) constraining properties and property values and a set of User Interface Profiles (UIP) that relate the constraints of the DSP to characteristics of the user interface. When fully deployed, the editor architecture will include a Sesame store for RDF persistence and a metadata server for deployment of all RESTful web services. Documents necessary for configuration of the editor including DSP, UIP, XSLT, HTML, CSS, and JavaScript files are stored as web resources.

Keywords. DCMI Abstract Model (DCAM), Sesame, description set profile (DSP), description set editor (DSE); resource description editor (RDE); DCMI Application Profile (DCAP); RDF

1 Background

In Australia, the process of developing a national curriculum from kindergarten to Year 12 in specific areas is underway with the support and participation of all governments. The Australian publicly-owned Curriculum Corporation partnered with JES & Co., a U.S. nonprofit dedicated to the education of youth [1], to use the Achievement Standards Network (ASN) [2] to ensure that existing and future digital curriculum resources will link seamlessly to all curricula in the country, including the emerging national curriculum.

2 Introduction

The Achievement Standards Network (ASN) provides an RDF-based framework for the description of achievement standards promulgated by governments and other organizations [hereafter, "promulgators"] to prescribe what K-12 students should know and be able to do as a result of specific educational experiences. The achievement standards of interest are frequently called curriculum objectives in the cataloging literature as well as academic standards, curriculum standards, learning indicators, benchmarks and an array of other names peculiar to each promulgator. For our purposes here, we shall refer to these variously-named achievement standards as 'learning objectives'. The correlation or mapping of learning resources such as lesson plans, curriculum units, learning objects as well as student achievement through portfolios and standards-based assessments (e.g., report cards) to these formally promulgated learning objectives is a growing imperative in the education environment [3]. International interest is high in sharing access to learning resources using learning objectives systematically described.

One thing missing until the work described here is the means to systematically engage promulgators in the process of describing their learning objectives using the ASN framework in a manner that supports global interoperability. The tasks in achieving RDF representations of learning objectives requires: (1) careful analysis of existing or planned objectives; (2) modeling the learning objectives as Semantic Web amenable resources; (3) development of an ASN-based schema with local extensions; and (4) application development to generate a suitable editor to support creation of the RDF graphs representing the learning objectives. What has been needed to support the ASN framework and these processes is a precise documentation process and a resource description editor that totally masks the conceptual complexities of RDF from the people creating the RDF data. Because the ASN framework supports both the use of local or project-specific extensions to its property set as well as integration of locally defined value spaces (controlled vocabularies), any ASN editor devised has to be configurable to accommodate varying attribute and value spaces and user interface configurations. This paper chronicles the first phase in the development of such a configurable editor—we call it alternatively the Resource Description Editor (RDE) and the Description Set Editor (DSE) depending on the context.

3 General Architecture—Dublin Core

The ASN framework is based on the Dublin Core Metadata Initiative's (DCMI) syntax-independent abstract information model (DCAM) [4]. The DCAM is intended to support development of Dublin Core Application Profiles (DCAP) of which the ASN DCAP is an example [5]. Based conceptually on Heery and Patel [6], a DCAP provides for a principled means of mixing and matching properties from disparate schemas sharing a common abstract model and describes constraints on the use of those properties within a given domain. The DCMI Usage Board has developed

Recommended Resources to document best practices in DCAP development including evaluative criteria for development [7] and guidelines [8].

In 2008, DCMI released a Recommended Resource describing the Singapore Framework that "...defines a set of descriptive components that are necessary or useful for documenting an Application Profile and describes how these documentary standards relate to standard domain models and Semantic Web foundation standards. The framework forms a basis for reviewing Application Profiles for documentary completeness and for conformance with web-architectural principles."[9] The Singapore Framework refined the understanding of application profiles and identified a specific documentation component that described the DCAP "Description Set" using a Description Set Profile (DSP). This was followed by a DCMI Working Draft of a description set constraint language that provides the means for translating a nascent DCAP into machine-processable RDF/XML or XML that can then be used to configure metadata generation applications. [10]

The ASN work described in the following paragraphs builds on the DCMI DCAP suite of specifications and practices and is the first example of DCAP deployment in the form of an RDF instance data editor configurable by means of any DCAM-conformant Description Set Profile. The ASN property schema is itself a component of the ASN DCAP since it adheres to the DCMI Abstract Model and defines usage of properties drawn from multiple namespaces. In this sense, the ASN AP is similar to other DCAPs such as the *Scholarly Works Application Profile* [11] and the *Dublin Core Collections Application Profile* [12].

4 ASN DSE/RDE Editor

The reasons for developing an architecture for a configurable editor for describing RDF graphs of learning objectives are twofold: (1) one reason stems from ASN data model's embracing open extensions to the ASN schema to accommodate properties and controlled vocabularies peculiar to national and organizational needs; and (2) the other reason stems from the need (requirement) to insulate the people generating the graphs of learning objectives from the complexities of RDF.

The use of the ASN schema in Australia for describing its new national curriculum illustrates the first of these reasons. The DSE editor had to be configured using Australian controlled vocabularies, and property labeling in the editor had to align with Australian common usage. The editor had to be incrementally and simply extensible (reconfigurable) over time as the Australian DCAP evolves. While sharing the underlying ASN data model to support interoperability, aspects of the Australian Application Profile (AU-AP) are different from those same aspects in the application profile configuring the editor for the United States. As a consortium with global participation, ASN anticipates such variance will be common as more nations and organizations embrace the ASN framework and bring their learning objectives to the Semantic Web.

4.1 The Description Set Editor Architecture

When fully deployed, the editor architecture described here will include a Sesame triple store [13] for RDF persistence and a metadata server for deployment of RESTful web services. As noted above, the editor will be configured by means of a DSP, and one or more (depending on the nature of the editor to be instantiated) User Interface Profiles (UIP). All documents necessary for configuration of the editor including the DSP, one or more UIP, XSLT, HTML, CSS, and JavaScript files are stored as web resources either in the editor's immediate triple store or elsewhere on the web. The basic relationships among the instantiating documents, the resulting editor, and the triple store are illustrated in Figure 1. We will briefly introduce the components of Figure 1 here and set them out more fully below.

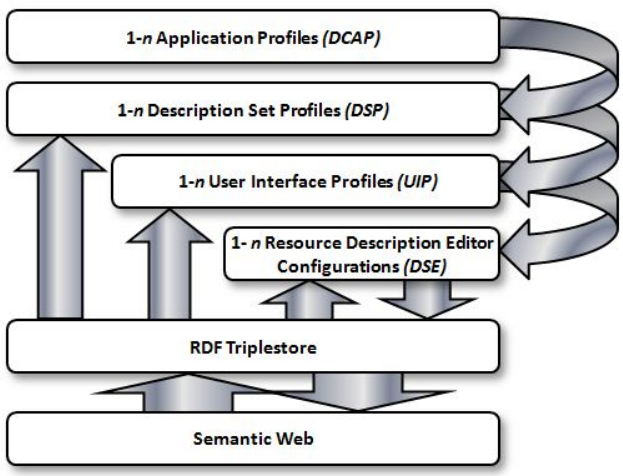

Fig. 1. Description Set Editor (DSE) architecture

While the DSP defines the constraints on permitted entities and their associated properties, it does not specify the treatment of those properties in the human interface of the editor generated. The DSP defines *what* can be present (i.e., properties and controlled vocabularies) and the UIP defines *how* those properties will be treated--e.g., a UIP controls: (1) local property labels; and (2) whether a controlled vocabularies will be treated as a small set of check boxes, a drop-down pick-list, or a separate search/browse interface. The UIP works in concert with XSLT, HTML and CSS. As already noted, all documents necessary for configuration of the editor are stored as web resources—represented in the figure by the upward arrows from the triple-store to the DSP and UIP.

The Figure denotes the DSE in terms of 1-*n* different configurations when fed 1-*n* configuring documents. As a result, the DSE core implementation may be viewed as an enabling shell with the resulting editor shaped largely by the configuration files it is fed.

4.2 The Description Set Profile

The DSP, as the machine-readable expression of the DCAP, is the principal component in defining the RDF data to be generated by the DSE. A DSP describes a *description set* defined by DCMI as "[a] set of one or more *descriptions*, each of which describes a single *resource*". [4] A record is defined by DCMI as an "instantiation of a *description set*…." [4] Thus, a description set (or record) is a bounded graph of related descriptions with each description in the set identified by a URI. So defined, an ASN 'record' might entail a description of the document *Social Studies Strand 4—Geography (2006)* and 1-*n* descriptions of individual learning objective assertions of which that document is comprised.

A DSP contains the formal syntactic constraints on: (1) classes of entities described by the 'set', (2) the desired properties and their form (literal or non-literal), (3) permitted value spaces (controlled vocabularies) as well as syntax encoding schemes (data types), and, (4) cardinality. As developed in DCMI's Singapore Framework, the DSP takes the form of a well defined templating structure that instantiates the DCAM. Figure2 graphically summarizes this templating.

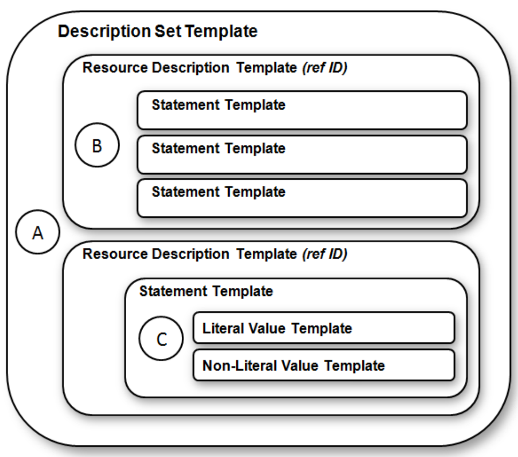

Fig. 2. DCAP constraint language templating: (a) a *description set template* is comprised of one or more resource description templates; (b) a *resource description template* is comprised of one or more statement templates; and (c) a *statement template* is comprised of either a literal or non-literal value template.

Since a description set is comprised of a set of related descriptions, a relationship between any two Resource Description Templates in the set is established by means of a Statement Template in the *subject* Resource Description Template referencing the refID of the *object* Resource Description Template.

4.3 User Interface Profile

As noted above in section 4.1, while the DSP defines the constraints on entities and properties for a description set, the User Interface Profile (UIP) defines the human interface—how those entities/properties are presented to the user of the editor. While the DSE is configured through a single DSP, in the current implementation, there is a UIP defined for each "view" of an entity and its properties. Currently, there is a UIP pair for each entity in the DSP—an *edit* UIP used for generating the editing environment, and a *view* UIP for generating the display-only environment. There are several other standard UIP present to support display for common functions of the editor such as the "pocket" used for managing relationships between entities (discussed below) and the overall visual frame in which the entity-specific UIP functions.

Like the DSP, UIP are web resources encoded in either RDF/XML or XML and are retrievable from the local triple store or elsewhere on the web. The various UIP control how properties are displayed and data entry is managed as well as specifying how value spaces are handled—e.g., whether the interface provides a simple drop-down picklist, check boxes or a full search and browse interface environment needed for large controlled vocabularies such as the Australian *Schools Online Thesaurus* (ScOT) [14].

4.4 Description Set Editor (DSE)/Resource Description Editor (RDE)

An instance of the editor configured by the means described above can be either a description set editor (DSE) where the intent is to describe sets of related resources or as a resource description editor (RDE) where the intent is to describe individual resources. The only difference between a DSE configuration of the editor and a RDE configuration is that the latter involves a DSP comprised of a single entity (i.e., a single resource description) while the former is be comprised of any number of related entities (i.e., a set of resource descriptions). In either configuration, the editor functions in the same manner.

5 The DSE Design Metaphor

The editor is designed around a "describe and relate" metaphor that builds on the underlying abstract information model and its templating framework in the DSP while eliminating exposure of the person using the editor to the information modeling complexities or the RDF encodings generated. In essence, the creator of RDF descriptions is asked to "describe" individual entities using straightforward input templates and then to “relate” those described entities where appropriate. For example, in the Australian national curriculum application profile, there are two entities in the domain model as defined by ASN—a curriculum document entity and a curriculum statement entity. The curriculum document entity describes a strand in a national curriculum such as “Mathematics.” In analyzing the curriculum document for representation, an analyst atomizes its text into individual learning objectives that will then be represented in the RDF graph as structurally and semantically related individual curriculum statement entities. Thus a curriculum document representation may be comprised of hundreds or many hundreds of curriculum statement entities.

Once the DSE is configured using this two-entity model, people generating RDF graphs create instances of model entities by selecting the appropriate entity from a menu. Once an entity is selected, the DSE presents the corresponding entity data template. Figure 3 is a screen shot of the data template generated for editing an existing curriculum statement from the Illinois Fine Arts curriculum as viewed through the Australian-configured DSE.

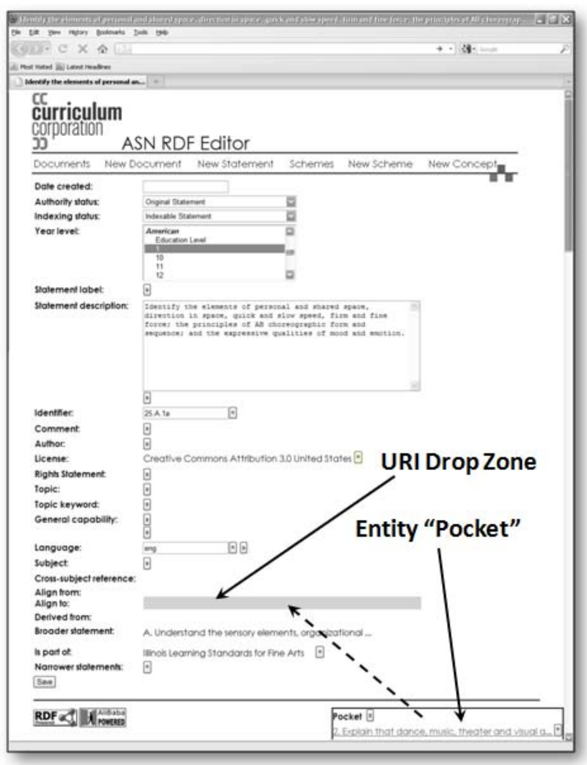

Fig. 3. DSE configured for editing a curriculum statement entity using the Australian DSP.

In addition to the otherwise conventional data entry and selection mechanisms apparent in the Figure, there are two features of note: (1) the Entity "Pocket", and (2) the URI Drop Zone for select properties. Both features are used in the processes of relating one entity in a description set to another entity either in- or outside the set. The "Pocket" is used to hold one or more URI entities for subsequent use. As shown by the broken-line arrow in Figure 3, any resource URI in the Pocket that has been defined as a class within a property's range of permitted classes in the DSP can be dragged from the Pocket to that property's Drop Zone as a value URI (grayed area in Figure 3). If the resource URI is not within a property's declared range, the resource URI will not be assigned.

Figure 4 illustrates a resource being dragged to the Pocket from a hierarchical display of related ASN statements in the New Jersey Visual and Performing Arts curriculum. The resource URI in the Pocket is now available to serve as a value URI of the *alignTo* property in Figure 3 when dragged from the Pocket to the Drop Zone for that property—thus asserting that the Illinois statement is semantically aligned to this New Jersey statement.

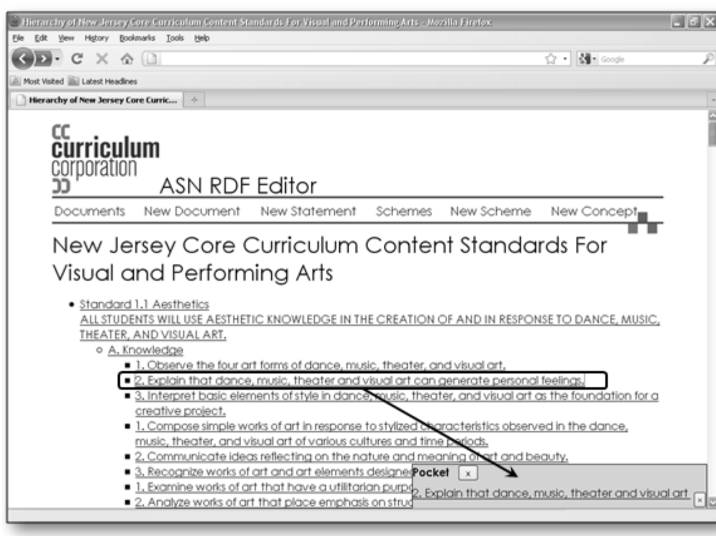

Fig. 4. Dragging a resource URI to the "Pocket".

6 Future Work

While the discussion here has been cast in terms of the ASN core schema and the Australian ASN application profile, the DSE can be configured to generate RDF graphs for any DCAM conformant description set. For example, work is underway to create a DSP and accompanying UIP for the Gateway to 21st Century Skills digital library to configure their metadata generation tools. [15]

Currently, the DSP and UIP controlling configuration of an instance of the DSE are created manually. We are working to integrate into the DSE editing environment an interface to facilitate the creation of both the DSP and all accompanying UIP.

7 Acknowledgements

This work is partially funded by the National Science Foundation under grant number DUE: 0840740. The authors acknowledge both the support of the Australian Curriculum Corporation and the Le@rning Federation in development of the Australian DSP and the functionality of the DSE and of Zepheira, LLC for partial development of the DSE solution stack.

8 References

[1] JES & Co., http://www.jesandco.org/
[2] Achievement Standards Network, http://www.achievementstandards.org/
[3] Sutton, S. & Golder, D. Achievement Standards Network (ASN): An Application Profile for Mapping K-12 Educational Resources to Achievement Standards. In, DC-2008 Proceedings of the International Conference on Dublin Core and Metadata Applications, pp. 69-79, http://dcpapers.dublincore.org/ojs/pubs/article/view/920/916
[4] Powell, A., Nilsson, M., Ambjörn, N., Johnston, P. & Baker, T. DCMI Abstract Model, http://dublincore.org/documents/abstract-model/
[5] ASN DCAP, http://www.achievementstandards.org/documentation/ASN-AP.htm
[6] Heery, R. & Patel, M. Application Profiles: Mixing and Matching Metadata Schemas, *Ariadne* 25, September 2000, http://www.ariadne.ac.uk/issue25/app-profiles/intro.html
[7] DCMI Usage Board. Criteria for the Review of Application Profiles, http://dublincore.org/documents/2008/11/03/profile-guidelines/
[8] Coyle, K. & Baker, T. Guidelines for Dublin Core Application Profiles, http://dublincore.org/documents/2008/11/03/profile-guidelines/
[9] Nilsson, M., Baker, T., & Johnston, P. The Singapore Framework for Dublin Core Application Profiles, http://dublincore.org/documents/singapore-framework/
[10] Nilsson, M. Description Set Profiles: A constraint language for Dublin Core Application Profiles. http://dublincore.org/documents/2008/03/31/dc-dsp/
[11] Scholarly Works Application Profile (SWAP), http://www.ukoln.ac.uk/repositories/digirep/index/SWAP
[12] Dublin Core Collections Application Profile, http://dublincore.org/groups/collections/collection-application-profile/
[13] Sesame, http://www.openrdf.org/
[14] Schools Online Thesaurus, http://scot.curriculum.edu.au/
[15] The Gateway to 21st Century Skills, http://www.thegateway.org/