What Terms Does Your Metadata Use? Application Profiles as Machine-Understandable Narratives

Thomas Baker[1], Makx Dekkers[2], Rachel Heery[3], Manjula Patel[3], and Gauri Salokhe[1]

[1] Birlinghoven Library, Fraunhofer-Gesellschaft
{thomas.baker,gauri.salokhe}@gmd.de

[2] Dublin Core Metadata Initiative (DCMI)
mail@makxdekkers.com

[3] UK Office for Library and Information Networking (UKOLN)
{r.m.heery,m.patel}@ukoln.ac.uk

Abstract

The SCHEMAS Registry aims at providing a selected and annotated overview of metadata vocabularies and their use in application environments. Based on harvested metadata in RDF (Resource Description Framework), the registry allows users to explore links between "namespace schemas", which declare standard definitions of metadata terms, and "application profiles" – RDF statements about the use or adaptation of namespace terms for particular domains, services, or projects. Where instance metadata does not follow standard namespaces or explicit data models, this style allows implementors to assert an explicit mapping to standard terms. Registering profiles can help harmonize metadata usage in particular domains and, in the longer term, could provide a machine-processable basis for automating crosswalks and conversions.

Keywords: Metadata, Semantic Web, RDF, Application Profiles

1 Motivation for a registry of schemas

> *The concept of machine-understandable documents does not imply some magical artificial intelligence which allows machines to comprehend human mumblings. It only indicates a machine's ability to solve a well-defined problem by performing well-defined operations on existing well-defined data. Instead of asking machines to understand people's language, it involves asking people to make the extra effort.* -- Tim Berners-Lee [1]

Since the emergence of a "Metadata Movement" in the mid-1990s, the proliferation of new standards for describing and processing information has presented a challenge to providers of Web-based resources. Metadata is expected to follow existing and emerging standards in order to facilitate integrated access to multiple information providers over the Web. However, there are many new standards, and most of them are still under development. And it is rare that the requirements of a particular project or site can all be met by any one standard "straight from the box." The broad and generic elements of Dublin Core, for example, must often be refined with qualifiers or extended with additional elements. With hundreds of thousands of new providers coming online, the integration of access to a diversity of providers will depend both on the harmonization of metadata usage ("good practice") and on the development of infrastructures for mapping between their different metadata vocabularies.

"SCHEMAS -- A Forum for Metadata Schema Implementors", an Accompanying Measure of the European Fifth Framework Programme, was designed to serve as a user guide to the diverse and often confusing landscape of new and emerging metadata standards.[2] Its target users are project or service implementers, especially among EU-sponsored projects, who must use these standards to design metadata models for their data. It has done this through compiling roadmaps and databases of projects and initiatives related to metadata vocabularies and through developing a registry of those vocabularies [3,4,5,6]. Using SCHEMAS materials, a reasonably experienced implementor with no prior experience should ideally be able to attain an overview of the problem and guidance on possible solutions.

In order to build this registry, the SCHEMAS Project has examined methods for declaring how a particular project or service has adapted existing standards in a particular "application profile". Our working hypothesis has been that making a large corpus of profiles easily searchable and browsable will promote convergence on good-practice solutions. The mechanics of these profiles and the broader issues they raise are the focus of this paper.

2 Application Profiles

Application profiles as a type of schema have become topical over the past year or so, but the concept itself is not new. The Z39.50 community, for example, has used "profiles" for constraining potential options and parameter values, where left open by standards specifications, to those required by a particular application (e.g., GILS or WAIS), function (e.g., simple author-title-subject searching), or user group (e.g., chemists or musicians). According to the "Framework and Taxonomy of International Standardized Profiles" (ISO TR 10000), a profile specifies how standards, particularly protocols, can be used in combination for meeting such requirements. [7]

In IEEE standardization committees for learning technology, a "standards profile" is "a technique of referencing (in contrast to defining) technical specifications... [permitting] the creation of a bundle of standards, each one tailored, extended, or constrained to meet the needs of the committee developing a standards profile... The point of using standards profiles is to *reuse* existing standards wording without having to recreate the words..." [8]

To users of the Digital Object Identifier, a DOI Application Profile is "the functional specification of an application (or set of applications) of the DOI System to a class of intellectual property entities that share a common set of attributes" for the purpose of enabling particular applications, from simple resource discovery to complex rights management. [9]

Jane Hunter reports that "Significant new initiatives such as TV-Anytime, MPEG-21, and the Open Archives Initiative are demanding application profiles which combine elements from a number of different existing standardized metadata schemas whilst maintaining interoperability and satisfying their own specific requirements through refinements, extensions and additions." [10] Similarly, the Federal Geographic Data Committee distinguishes between its the Content Standard for Digital Geospatial Metadata, and a profile based on that standard, which "describes the application of the Standard to a specific user community". A profile "always contains the Standard, plus modifications to the optionality or repeatability of non-mandatory elements in the Standard" and "may also contain extended elements"; it may be formalized through the FGDC process or used informally by a user community.[11] ISO/DIS 19115, another standard for geographic datasets, likewise provides for the development of "community profiles" within user communities, nations, or organizations.[12]

In the European project DESIRE, which developed and tested new techniques for resource discovery and network management between July 1998 and June 2000, an "application profile" was a set of elements with usage information on associated element values, schemes, or controlled vocabularies used for particular projects, computer programs, interchange formats, or information services. In the DESIRE style, an application profile cannot introduce new data elements; it must take each element from an associated namespace. A profile can group together data elements from multiple vocabularies; and it may declare a scheme of valid values appropriate for a particular application. [13,14]

On the basis of this experience, Rachel Heery and Manjula Patel have defined application profiles as "schemas which consist of data elements drawn from one or more namespaces, combined together by implementors, and optimized for a particular local application". By definition, such profiles depend for their elements on namespaces. Namespaces, in this context, are element sets maintained as stable points of reference. They serve to "identify the management authority for an element, support definition of unique identifiers for elements, [and] uniquely define particular data element sets or vocabularies". Management authorities can range from internationally recognized standards bodies, to maintainers of unofficial or de-facto standards, down to projects or services with special data elements defined primarily for local use. [15] This contrast between "namespaces that declare" and "profiles that reuse" provided the starting-point for our discussion of application profiles in the SCHEMAS context.

3 What users want from registries

The development of application-profile guidelines for the SCHEMAS Registry has been formed largely by our evolving understanding of the user requirements to be addressed by a registry service. The term "registry" covers a broad range of databases, documentation services, or Web-based portals providing access to schemas. The term is sometimes associated with tightly controlled network services, such as URN registries that work with hierarchies of naming authorities to resolve persistent resource names [16]; one design for metadata registries, the standard ISO/IEC 11179-6, similarly envisions a hierarchy of central and domain-specific registration authorities for associating data elements with maintenance agencies. [17] An XML Registry of the Organization for the Advancement of Structured Information Standards (OASIS), in contrast, aims at facilitating the exchange of DTDs, XML schemas, and related specifications seen as modules that can be directly reused to provide interoperability among a set of service providers. [18]

The registry prototyped in the DESIRE Project focused on the disclosure of information about the authoritative usage of metadata -- element definitions, usage notes, allowed schemes, and mappings to other namespaces -- and explored typical user queries.[19] The SCHEMAS Registry builds on this DESIRE experience, aiming at providing a search and browsing interface to a selection of schemas and at wrapping those schemas in a helpful critical and descriptive context. The emphasis is on serving up term-level documentation in response to queries -- cross-sections, for examples, of definitions and usage notes from a range of standard namespaces and local profiles, within specific fields or across domains. (As of June 2001, our intention seamlessly to integrate searches on schemas with searches on related descriptive information and peer-review commentaries has been frustrated by software difficulties, which have delayed the availability of an integrated search interface on the Web.)

Our primary goal has been to help humans find out about metadata terms in use -- their official definitions, local variations and extensions, and the various schemas in which they are embedded. The purpose is to help designers of information services discover metadata terms that have already been created or standardized by others and align their own schemas with those of related information providers. The longer-term goal, however, has been to build a corpus of machine-understandable schemas that can be accessed and processed directly by various software applications, for example to map or convert between schemas or to configure the interface of a metadata creation tool.

Exploring these longer-term goals in more detail with potential users of the registry was an important goal of the three SCHEMAS workshops -- in Bath (May 2000), Bonn (November 2000), and Budapest (May 2001).[20] Some of what we learned confirmed expectations: designers of new schemas want to know if the terms they need have already been defined or standardized somewhere; they want to see how other projects or services in their field use metadata; and they would like to follow links from those schemas to the projects which use them, to any available rules for metadata creation, to documentation or critical reviews that place those schemas into a broader context. Almost universally, registries are seen as our best hope in the medium term for a scalable solution to the problem of mapping and translating between a diversity of schemas.

The Bonn workshop also focused to some extent on issues of quality. Descriptions and links should be sufficiently complete and reliable to be included in the registry. Metadata about these schemas should describe its subject area, genre, and language; indicate its history and status as a draft or standard; and identify its developers and maintainers. Schemas should be syntactically well-formed as XML/RDF, and application profiles should adhere to clear ground rules on content and form. Ideally, the type of schema (eg, namespace or application profile) should be clear enough to use as a search criterion. In an area where terms can have quite different meanings in different contexts, this implies the availability of good FAQs and glossaries.

Some of the most interesting discussions have been about the similarities and differences between metadata schemas and other types of "controlled vocabularies" such as classification schemes, thesauri, and subject headings. In Budapest, there was general agreement that all such vocabularies -- metadata terms included -- belong in the same conceptual framework. Indeed, the distinction between "namespaces that declare" and "profiles that reuse" seems like a useful distinction between a canonical set of subject headings, for example, and selective adaptations of those headings for particular uses. Analogously, the discussants recognized that crosswalks between near-equivalent metadata terms were conceptually similar to mappings between terms in different thesauri. While it was recognized that thesaurus terms may be embedded in rich webs of related terms, making their reuse out-of-context particularly problematic, the clear requirement was to standardize conventions for describing all vocabularies machine-understandably, so that they can be exchanged and cross-linked over the Web.

Whether metadata vocabularies should be accessed through the same sort of registry as other controlled vocabularies was seen as a much different question. While a shared conceptual model would permit this, differences in the nature of vocabularies, their size, granularity, and expected use imply different sorts of interfaces. For both cases, the discussants recognized the importance of editorial control and selection. While it would make sense for a registry of schemas and vocabularies maintained (for example) by the Food and Agriculture Association of the United Nations to cover food- and agriculture-related vocabularies as exhaustively as possible, the SCHEMAS Registry, with its goal of providing a high-level overview across domains, might want to limit its coverage of agriculture to a few exemplary schemas, with pointers to an FAO-maintained registry for further information.

4 "What does your metadata say?"

The style of Application Profile we developed is an answer to the question: "What does your metadata say?", or more precisely: "What terms does your metadata use, and how does it use them?". The answer is best characterized as a set of statements of certain fixed patterns. W3C's Resource Description Framework provides the basic grammar for these statements: a word order of Subject - Predicate - Object, where the Predicate is a verb phrase characterizing the relationship between the Subject and Object.

In practical terms, the sum of such statements is a page or two of XML-formatted metadata looking something like Appendix A (below); this is what gets parsed and indexed by a registry database. But this XML encoding is only intended for consumption by database software (or XML geeks). The logic of the RDF statements is easier to explain with "node-and-arc" diagrams, where the Subject and Object are nodes and the Predicate is an arc.

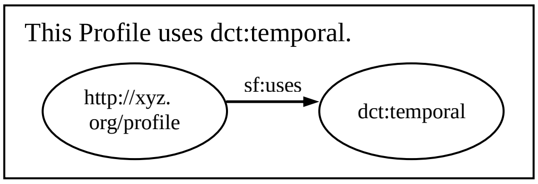

Figure 1. Use a term from a namespace.

Figure 1, for example, says in effect: "This profile uses the term Temporal from the namespace for Dublin Core metadata terms designated here with the prefix (dct:)". Note that each part of the statement -- Subject (http://xyz.org/profile, in this case the URI of the application profile), Predicate (sf:uses), and Object (dct:temporal) -- has a unique Web address, as the prefixes "sf:" and "dct:" resolve to "http://www.schemas-forum.org/terms/" and "http://purl.org/dc/terms/" respectively.[21] These addresses identify the namespace schemas where the terms "uses" and "temporal" are declared and defined. Figure 1 represents the most basic statement of an application profile: "This profile uses this term from this namespace". Figures 2 to 7 will now illustrate various types of additional information that can be associated with these terms when they are adapted for a particular application environment.

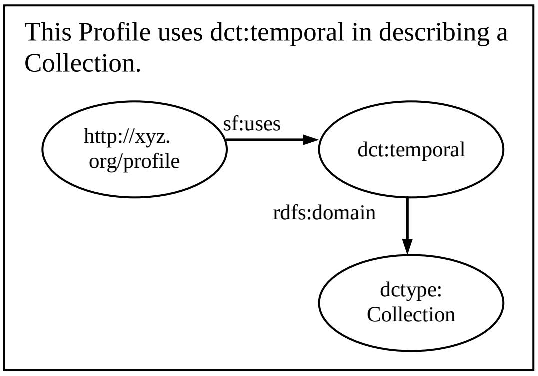

Figure 2. Specify a class of object to which it refers

In Figure 2, the Object of the statement in Figure 1 becomes the Subject of a second statement: "This profile uses dct:temporal, and dct:temporal is used specifically in reference to collections". In other words, the metadata is not about "resources" in a generic sense, but refers specifically to things like manuscript collections, museums, or archives.

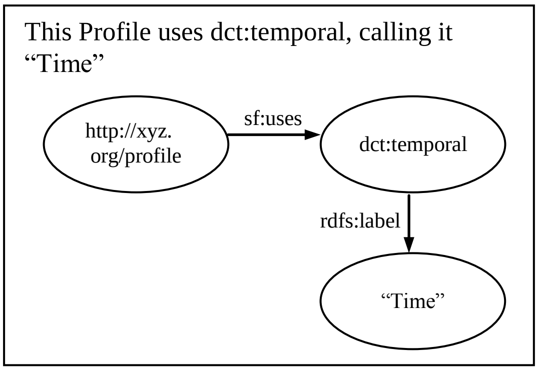

Figure 3. Provide a local label

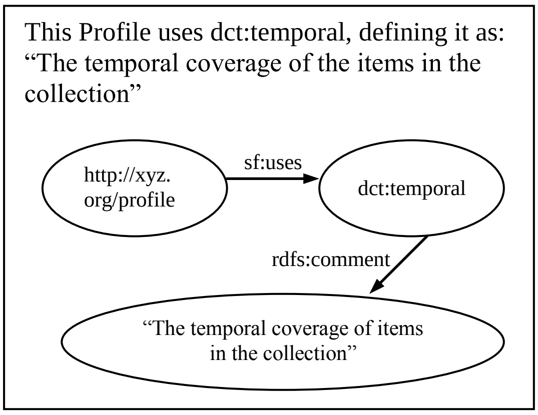

Figure 4. Provide a local definition

In the official documentation of its namespace, the Dublin Core qualifier "dct:temporal" is labeled "Temporal" and defined as "Temporal characteristics of the intellectual content of the resource". Figures 3 and 4 show how these "default" labels and definitions can be replaced, or overridden, with labels and definitions that are more appropriate or understandable for users in a particular application context. In this example, the term "dct:temporal" is labeled "Time" and defined as "The temporal coverage of items in the collection". Figure 5 simply adds a local usage guideline ("This element is optional").

The ability to override standard definitions with local ones evokes a danger of semantic drift, as meanings may be stretched beyond their intended scope. If profiles were to re-use and redefine terms from other profiles, then one could easily imagine a chain of semantically shifting derivations in the manner of the children's game "Telephone". The extent of such drift, however, would be naturally limited to the extent that profiles take their terms directly from official namespaces. That people will misunderstand, stretch, or otherwise transform the intended meanings or scopes of metadata terms is in the nature of how humans use language. In the face of this inevitability, profiles offer a standard form at least for documenting such adaptations, good or bad, and for assessing how consistently or coherently particular metadata terms are implemented in practice.

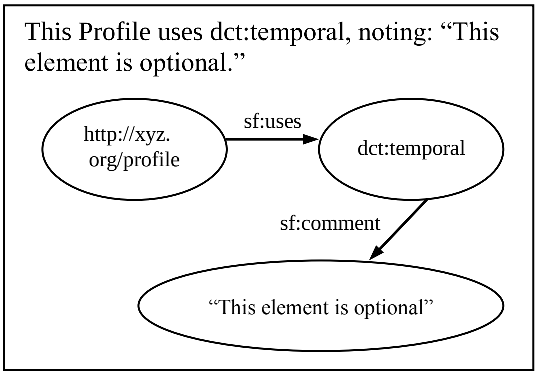

Figure 5. Add a usage guideline

A profile might also include information about permissible element values. In Figure 6, the profile uses the Dublin Core element Subject (dc:subject), and it also uses a qualifier of Subject (dct:LCSH) for specifying that the value of dc:subject is a term taken from the Library of Congress Subject Headings (LCSH). (The diagram shows an additional construct: it says that the range of acceptable values for dc:subject is restricted to the value set signified by dct:subjectScheme. It then defines dct:LCSH as a sub-set of that value set. In this case, dct:LCSH is related to dc:subject through dct:subjectScheme in the DCQ namespace itself, so the additional declaration here may be redundant. This is an example of where clarification is needed, from research and implementation experience on the division of labor between namespaces and profiles.)

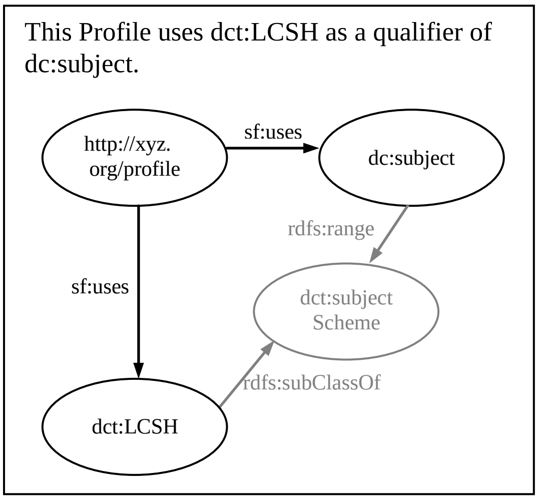

Figure 6. Use an encoding scheme for a term

Notice that the Subject of the statements in Figures 1 to 6 is the Profile itself. Figure 7 tells us about the Profile itself: its name ("XYZ Project Profile"), its type ("sf:ApSchema", identifying it as an application profile), and the application of which it is a profile ("http://xyz.org").

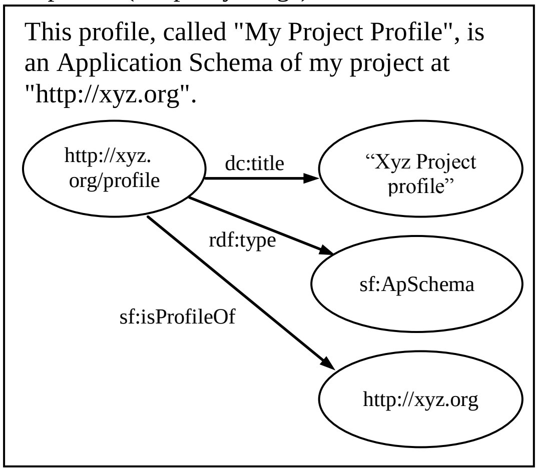

Figure 7. Describe the profile itself, citing the application to which it refers

The practical usefulness of defining a profile as a set of simple sentence patterns is shown by the queries it supports. Creating a searchable index of RDF statements may be pictured as a process of superimposing (joining) multiple statements via their shared nodes. The URIs that associate each part of the sentence with a unique Web address – the Subject (a resource), Predicate (a vocabulary term from a namespace schema), and the Object (another resource or a string literal) – serve as fixed anchor points for merging data from a diversity of sources.

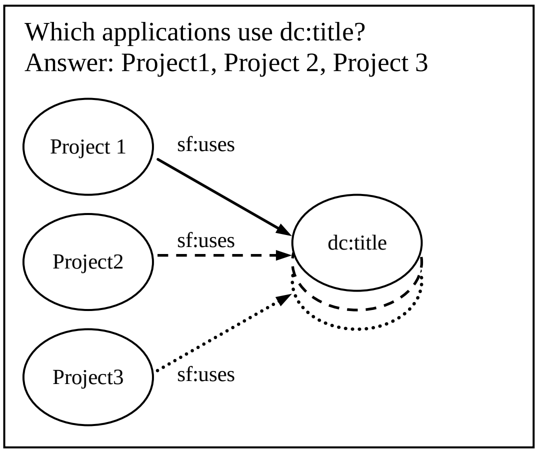

Figure 8. Joining statements as a basis for queries

In Figure 8, three sentences sharing sf:uses as the Predicate and dc:title as the Object yield an answer to the question: "Which applications use dc:title?" Figure 9 takes the query one step further and narrows the search result of Figure 8 to those projects that use dc:title specifically in reference to collections, as opposed to resources more generically.

The joining of sentences in this manner makes clear that the simple model presented in Figures 1 through 7 may require one further improvement. RDF sentences, also known as triples, stand on their own, and it is through joining that they are placed into a context. If an application profile asserts local labels, definitions, and usage notes to be properties of a term defined in a namespace somewhere, then each such local property will appear in a joined graph as a separate property of the namespace term – independently of the other local properties associated with that namespace term in a particular profile.

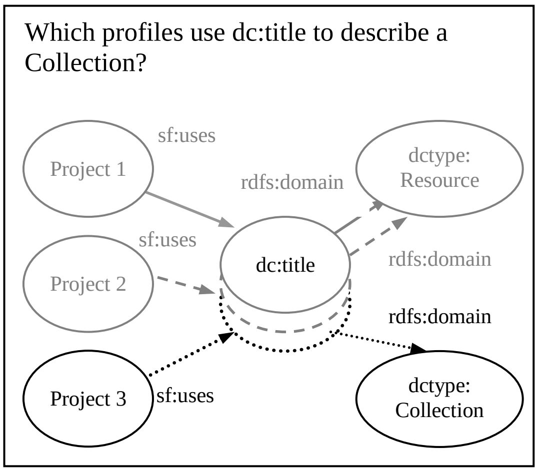

Figure 9. Narrowing a search

From a modeling point of view, it may be preferable for the triples not to refer directly to a namespace term, but to an entity representing the term "as used" in the local context. This can be done with an "intermediate node" – a modeling construct that groups all of the locally defined properties of a namespace term in a way that allows them to appear as a package when the RDF graphs are joined. In Figure 10 (and Appendix B), the intermediate node is "anonymous" – it does not itself have a unique identifier that would allow it to be referenced as such, in this case as a particular adaptation of dct:temporal.

In RDF, one can however assign an identifier to the node, giving it in effect a URI and allowing the locally adapted term to be referenced by other metadata like any other namespace term. In principle, this would allow one application profile to use a namespace term indirectly, by using an adapted term from another profile. Whether such practice should be promoted is an open question. It is easy to picture this getting out of hand, with profiles based on profiles based on profiles, threatening semantic drift. But this is perhaps unavoidably an issue in the linguistics of a Semantic Web generally.

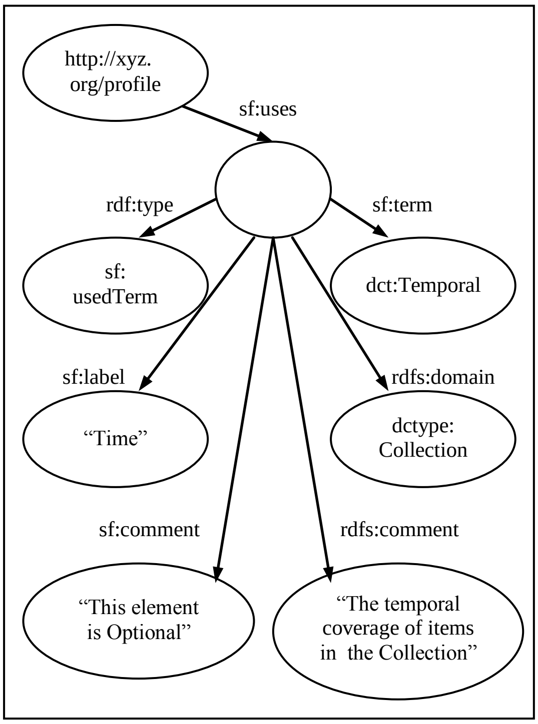

Figure 10. Next step: grouping the properties of a “term used”

5. Interpretation versus validation

Metadata is produced and consumed in variety of imperfectly interoperable encodings and contexts, from commercial databases to embedded headers, protocol streams, and XML files. Just as inconveniently, much of the world's metadata is quite messy conceptually. Even if its format were to be normalized, the metadata may not follow a data model that is adaptable for unanticipated uses or for merging with metadata from other sources.

Figure 11, for example, shows a piece of metadata in well-formed XML. To a human reading this metadata, the intent seems clear enough: it is a description of an author who has a name, affiliation, email address, shoe size, and birthday; the name, in turn, has four components: family, given, nick, and title. Without any additional context, however, a machine would not be able to do much with this information. At a minimum, a Document Type Declaration (DTD) would be needed to list the expected sequence of tags. Without a reference to uniquely identified namespaces, moreover, the machine would have to use heuristics to guess that the Author tag is related to the Dublin Core element Creator. Even if that relationship were clear, a search engine wanting to index the names of creators would have to know (or guess) alot more about the tag structure in order to reliably extract the name "Joe Smith" -- ignoring the shoe size and other (for this purpose) extraneous information.

```
<author>
  <name>
     <family>Smith</family>
     <given>Joseph</given>
     <nick>“Joe”</nick>
     <title>Dr.</title>
  </name>
  <affiliation>NewYork University</affiliation>
  <email>joe.smith@yahoo.com</email>
  <shoeSize>12W</shoeSize>
  <bday>1978-05-01</bday>
</author>
```

Figure 11. XML tags can be arbitrarily nested

This block of metadata may be perfectly useful within a given application environment, and it will be useable by any other application that knows and recognizes this particular nested structure. When described by shared DTDs or XML schemas, such metadata can indeed provide a limited form of semantic interoperability. The problem is there is no inherent limit to the ways such a structure could be nested; a different XML schema would be needed to describe each such structure; and differently nested structures are hard to compare or merge. If we assume that metadata on the open Web will be reused and repurposed for a variety of contexts, it is helpful to limit these possibilities.

When used judiciously, RDF provides a grammar for reducing data relationships to parsable sentences that follow a simple and predictable form. The Application Profile style outlined above uses such sentences to make assertions about the information model used for the metadata of a particular project or service. However, the very exercise encourages the author of such a profile to make an explicit commitment to namespaces that were perhaps never originally consulted and to a data model that was perhaps never clearly intended when the application schema was originally designed. The result, then, could be seen as an interpretation, or view, of an underlying metadata model that may actually be alot less clear.

This is not a bug, but a feature. If the metadata structures of the world really are too messy and arbitrarily structured to merge in any scalable way, then clearly there needs to occur some form of translation into simpler, more predictable, pidgin-like forms such as the Subject - Predicate - Object sentences of RDF. And if such translations are difficult to automate, who is (in principle) better qualified to convey the intention of a metadata structure than its authors? Application Profiles, in this sense, might be seen as a form of Mapping Profile from a particular local language to a more universal and predictable Web language. It involves asking people to "make the extra effort" of translating data into a well-defined form that machines can process, as suggested by Tim Berners-Lee in the quote at the beginning of this paper.

In this sense, the Application Profile style adopted for the SCHEMAS Registry has a certain affinity with the ABC vocabulary developed by the Harmony Project – an RDF-like language for expressing historical sequences of events implicit in metadata records as clear narratives. For example:

> A dinosaur bone was discovered by Richard Leakey in 1995 in Kenya. In 1971 it was acquired by the British Museum in London and added to its collection. In 1991, Jean Smith, the curator of the British Museum, classified the bone as part of a plesceosaur. In 1998, Richard Hill took an image of the dinosaur bone and it was mounted on the museum's web site. [22]

All of this information was presumably already present in the metadata about the bone. The ABC approach is to express those events in metadata that can be compared and merged with metadata about other explicitly described events.

As discussed above, many communities have adopted some notion of "profile" to distinguish standard vocabularies from adaptations of the same. Most of these profile types are designed to be consumed by humans (for example, by standardization committees or database designers) as opposed to being used directly by software (for example, as a basis for automatically validating instance metadata).

Confusingly, this contrast between human-usability and machine-processability evokes a somewhat analogous contrast between two competing W3C specifications for XML-based schemas: the Resource Description Framework Schema (RDFS) [23] and the XML Schema [24]. As characterized by Jane Hunter and Carl Lagoze, each standard has its advantages: RDF Schemas are stronger on declaring the semantics of metadata terms in ways that support flexible, dynamic mapping between vocabularies; while XML Schemas are stronger on modeling local structural, cardinality, and datatyping constraints for automatic validation. Hopefully, W3C will eventually bring about a convergence of these two overlapping standards; for now, they conclude, the most logical approach to application profiles involves using RDF and XML Schemas in combination, exploiting these complementary strengths. [10]

The specific strength of RDF in modeling declarative statements of a known form makes it a good choice for application profiles in the SCHEMAS Registry. Nevertheless, Jane Hunter has suggested that many of the requirements for application profiles discussed in the SCHEMAS Project can be met with XML Schemas and argues that registries should be designed to handle both schema types. [25]

In principle, it would be desirable if XML schemas could be infused directly into the registry and not via a translation into RDF. However, such transformations are notoriously difficult to automate, both for the inherent technical difficulty and for the problems of interpretation discussed above. Experimental methods for embedding RDF within XML schemas or otherwise preparing them for automatic translation are still in the realm of research [10], while ongoing efforts by W3C at convergence between the standards could render such methods obsolete.

The experience of the SCHEMAS Project has been that tools for handling just one of the standards alone are challenging to implement on a production basis. Our current prototype is based on the Extensible Open RDF (EOR) Toolkit, an open-source development project at the Online Computer Library Center (OCLC) [26]. We are coordinating closely in this with developers in the Dublin Core Metadata Initiative (DCMI), which is adapting the toolkit to manage DCMI's namespaces.

The diversity of standards and approaches does, however, suggest a need to clarify, collectively, a functional typology of schemas. The distinction between Namespace and Profile, for example, has appeared in so many different contexts that it seems like a good candidate for a more general agreement. As for Application Profiles, there is an undisputed need for XML schemas as a basis for the automatic validation of metadata records. But semantic interoperability on a broader scale would seem also to require a style of profile, more documentary and interpretive, such as the one presented here.

By reducing the model to a small number of simple statements, our intent is to facilitate the use of fill-in-the-blank templates to help implementors create well-formed profiles without having to work directly with the RDF serialization syntax. If such profiles, by definition, only reuse terms from namespaces, then as a next step we will need to focus also on helping implementors make their own namespace declarations when needed. While emerging policies for the management of the Dublin Core namespace provide one model for doing so, guidelines for good practice generally have yet to become clear. The vision of an interoperable space of namespaces and profiles on the Web will only be realized to the extent that maintenance agencies and projects provide compatibly machine-understandable representations of their vocabularies. Technically, this is within our reach; the challenge lies in reaching a critical mass of uptake.

Acknowledgements

Credit is due to Eric Miller of W3C for suggesting the verb "uses", to Dan Brickley of W3C for first suggesting the grouping mechanism shown in Figure 10. Special thanks to Carl Lagoze of Cornell University for helping to clarify the role of interpretation and narrative and to Andy Powell of UKOLN for helpful suggestions.

References

[1] Berners-Lee, Tim, What the semantic web can represent, September 1998, http://www.w3.org/DesignIssues/RDFnot.html.
[2] SCHEMAS Project. http://www.schemas-forum.org.
[3] SCHEMAS Project, Metadata Watch Report 1, May 2000, http://www.schemas-forum.org/metadata-watch/1.pdf.
[4] SCHEMAS Project, Metadata Watch Report 2, September 2000, http://www.schemas-forum.org/metadata-watch/2.pdf.
[5] SCHEMAS Project, Metadata Watch Report 3, December 2000, http://www.schemas-forum.org/metadata-watch/3.pdf.
[6] SCHEMAS Project, Standards framework report 1, September 2000, http://www.schemas-forum.org/stds-framework/1.pdf.
[7] Z39.50 International Standard Maintenance Agency, About Profiles, http://lcweb.loc.gov/z3950/agency/profiles/about.html
[8] IEEE 1484.18: Platform and Media Profiles, http://edutool.com/pmp/.
[9] Digital Object Identifier System, DOI Application Profile, http://www.doi.org/doi-ap.html.
[10] Jane Hunter and Carl Lagoze, Combining RDF and XML Schemas to enhance interoperability between metadata application profiles, May 2001, http://www10.org/cdrom/papers/572/index.html.

[11] Federal Geographic Data Committee, Guidelines for creating a profile for the Content Standard for Digital Geospatial Metadata, 1998, http://www.fgdc.gov/metadata/csdgm/profile.html.
[12] ISO/TC 211 Geographic information / Geomatics, http://www.statkart.no/isotc211/scope.htm.
[13] DESIRE Metadata Registry Framework, March 2000, http://www.desire.org/html/research/deliverables/D3.5/d35.html.
[14] DESIRE Registry Data Model, December 2000, http://desire.ukoln.ac.uk/registry/docs/datamodel.html.
[15] Rachel Heery and Manjula Patel, Application profiles: mixing and matching metadata schemas, Ariadne 25, September 2000, http://www.ariadne.ac.uk/issue25/app-profiles/intro.html.
[16] William Arms et al. Uniform Resource Names: A progress report. D-Lib Magazine, February 1996, http://www.dlib.org/dlib/february96/02arms.html.
[17] ISO/IEC 11179-6, Registration of Data Elements, http://www.sdct.itl.nist.gov/~ftp/l8/other/Standards/iso11179/iso11179-6.PDF.
[18] XML.ORG, XML Registry, http://www.xml.org/xml/registry_search.jsp.
[19] DESIRE Metadata Registry, http://desire.ukoln.ac.uk/registry/.
[20] SCHEMAS Project, Workshops, http://www.schemas-forum.org/workshops/.
[21] DCMI, Namespace policy for the Dublin Core Metadata Initiative, http://dublincore.org/documents/dcmi-namespace.
[22] Harmony Project, ABC Harmony Data Model Ver. 2, http://www.metadata.net/harmony/ABCV2.htm.
[23] W3C, Resource Description Framework (RDF) Schema Specification 1.0, http://www.w3.org/TR/rdf-schema.
[24] W3C, XML Schema Part 0: Primer, http://www.w3.org/TR/xmlschema-0.
[25] Jane Hunter, An XML Schema approach to application profiles, October 2000, http://archive.dstc.edu/maenad/appln_profiles.html.
[26] The EOR Toolkit, http://eor.dublincore.org.

Appendix A

```
<! -- Namespace inclusion block -->
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999
    /02/22-rdf-syntax-ns#"
xmlns:rdfs = "http://www.w3.org/2000/01/rdf-
    schema#"
xmlns:dc = "http://purl.org/dc/elements/1.1/"
xmlns:dcq = "http://purl.org/dc/terms/"
xmlns:sf = "http://www.schemas-forum.org/
    terms">

<! -- Description of this profile (see Figure7) -->
<sf:ApSchema rdf:about = "http://xyz.org/
    profile">
<dc:title> Xyz Project Profile </dc:title>
<sf:isProfileOf rdf:resource= "http://xyz.org" />

<! -- Use a term, overriding defaults and adding
    notes (see Figures 1-5) -->
<sf:uses>
<rdf:Description about = "http://purl.org/dc/terms/
    temporal">
<rdfs:label>Time</rdfs:label>
<rdfs:comment>The temporal coverage of the
    items in the collection.</rdfs:comment>
<sf:comment>This element is optional
    </sf:comment>
<rdfs:domain rdf:resource = "http://purl.org/dc/
    dcmitype/Collection"/>
</rdf:Description>
</sf:uses>

<! -- Use an encoding scheme (Figure 6) -->
<sf:uses>
<rdf:Description about = "http://purl.org/dc/
    elements/1.1/subject">
<rdfs:range rdf:resource =  "http://purl.org/dc/
    terms/SubjectScheme"/>
<rdfs:domain rdf:resource = "http://purl.org/dc/
    dcmitype/Collection"/>
</rdf:Description> </sf:uses>

<sf:uses>
<rdf:Description about = "http://purl.org/dc/
    terms/LCSH">
<rdfs:subClassOf rdf:resource = "http://purl.org/
    dc/terms/SubjectScheme"/>
</rdf:Description> </sf:uses>

</sf:ApSchema>
</rdf:RDF>
```

Appendix B

```
<! -- RDF Code for Figure 10 -->
<! -- Namespace inclusion block -->
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999
      /02/22-rdf-syntax-ns#"
xmlns:rdfs = "http://www.w3.org/2000/01/rdf-
      schema#"
xmlns:dcq = "http://purl.org/dc/terms/"
xmlns:sf = "http://www.schemas-forum.org/
      terms/">

<! -- Use a term, overriding defaults and adding
      notes (see Figure 10) -->
<sf:ApSchema rdf:about = "http://xyz.org/ profile
      ">
<sf:uses>
<sf:usedTerm>
<sf:term rdf:resource = "http://purl.org/dc/
      terms/temporal" />
<rdfs:label>Time</rdfs:label>
<rdfs:comment>The temporal coverage of the
      items in the collection. </rdfs:comment>
<sf:comment>This element is optional.
      </sf:comment>
<rdfs:domain rdf:resource=" http://purl.org/dc/
      dcmitype/Collection" />
</sf:usedTerm>
</sf:uses>

</sf:ApSchema>
</rdf:RDF>
```