Application profiles: mixing and matching metadata schemas

Rachel Heery and Manjula Patel introduce the 'application profile' as a type of metadata schema.

1. Background

This paper introduces application profiles as a type of metadata schema. We use application profiles as a way of making sense of the differing relationship that implementors and namespace managers have towards metadata schema, and the different ways they use and develop schema. The idea of application profiles grew out of UKOLN's work on the DESIRE project (1), and since then has proved so helpful to us in our discussions of schemas and registries that we want to throw it out for wider discussion in the run-up to the DC8 Workshop in Ottawa in October.

We define application profiles as schemas which consist of data elements drawn from one or more namespaces, combined together by implementors, and optimised for a particular local application. The experience of implementors is critical to effective metadata management, and this paper tries to look at the way the Dublin Core Metadata Element Set (and other metadata standards) are used in the real world. Our involvement within the DESIRE project reinforced what is common knowledge: implementors use standard metadata schemas in a pragmatic way. This is not new, to re-work Diane Hillmann’s maxim ‘there are no metadata police’, implementors will bend and fit metadata schemas for their own purposes. This happened (still happens) in the days of MARC where individual implementations introduce their own ‘local’ fields by using the XX9 convention for tag labelling. But the pace has changed. The rapid evolution of Rich Site Summary (RSS) has shown how quickly a simple schema evolves in the internet metadata schema life cycle.

The Warwick Framework (2) gave an early model for the way metadata might be aggregated in ‘packages’ in order to combine different element sets relating to one resource. The work on application profiles is motivated by the same imperative as the Warwick Framework, that is to provide a context for Dublin Core (DC). We need this context in order to agree on how Dublin Core can be combined with other metadata element sets. The Warwick Framework provided a container architecture for metadata ‘packages’ containing different metadata element sets. Application profiles allow for an ‘unbundling’ of Warwick Framework packages into the individual elements of the profile with an overall structure provided externally by namespace schema declarations.
The Resource Discovery Framework (RDF) syntax has provided the enabling technology for the combination of individual elements from a variety of differing schemas, thus allowing implementors to choose which elements are best fit for their purpose.

2. Who is constructing metadata schemas? Who is managing metadata schemas?

Sometimes it seems as if there are two distinct sets of people involved with constructing and managing schemas:

Standards makers

They use a top down approach, driven by a search for a coherent element set which can be viewed as a ‘standard’, they are concerned with the integrity of the data model, they insist on a well structured element set.

Implementors

Their primary motivation is to produce an effective differentiated service, they are looking for innovative, effective solutions to service delivery. These service providers can, thanks to the flexibility of web technology, choose or construct a metadata schema best fitted for their purpose.

Both sets of people are intent on describing resources in order to manipulate them in some way. Standard makers are concerned to agree a common approach to ensure inter-working systems and economies of scale. However implementors, although they may want to use standards in part, in addition will want to describe specific aspects of a resource in a ‘special’ way. Although the separation between those involved in standards making and implementation may be considered a false dichotomy, as many individuals involved in the metadata world take part in both activities, it is useful to distinguish the different priorities inherent in the two activities. It is a particular strength of the Dublin Core Metadata Initiative (DCMI) that many people are deeply involved in both approaches, and so we hope that within the DC community we will be able to have a fruitful discussion on the requirements of those looking for an ‘authoritative’ version of the Dublin Core Metadata Element Set (DCMES)and those whose primary requirements are to do with ‘practice’.

3. Examples of emerging schemas

In order to illustrate how schemas work in practice we can examine two emerging schemas

DC Education Schema

The DC Education Working Group has proposed a schema (3) for describing educational resources. Jon Mason of Education Network Australia (EdNA) and Stuart Sutton of the Gateway to Educational Materials (GEM) have led this activity, with a particular focus on five areas of interest to educational metadata projects:

- users
Subsequent discussions at meetings and on mailing lists considered whether elements could be identified and evaluated within these areas. The recommendation of the DC Education Working Group suggests a schema incorporating

- Various standard DC elements and recommended qualifiers
- DC Education ‘namespace’ (domain specific) extensions such as
  - DCEducation Element: \textit{audience}
  - DCEducation Audience qualifier: \textit{mediator}
  - DCEducation Element: \textit{standard}
  - DCEducation Standard qualifier: \textit{identifier}
  - DCEducation Standard qualifier: \textit{version}
  - DCEducation Relation qualifier: \textit{conforms to}
- Various IEEE Learning Object Metadata (LOM) (4) elements such as
  - InteractivityType
  - InteractivityLevel
  - TypicalLearningTime

We can see from this schema extract that it consists of DC ‘standard’ elements, domain specific additions to recommended standard DC elements, and particular elements from other distinct element sets.

\textbf{RSLP collection description schema}

Andy Powell of UKOLN has been leading an initiative on collection level descriptions. The purpose of the collection description schema (5) is to describe newly digitised special collection catalogues being created as part of the UK Research Support Libraries Programme, but is intended longer term to have a wider application within the Distributed National Electronic Resource. The schema is intended to facilitate the simple description of collections, locations and related people. Particular areas of interest have centred on the best way to describe collection policy, collection strengths, and the people and organisations with responsibility for the collection. Consensus has been reached within the programme on the schema and a metadata creation tool has been developed. An extract of this schema includes

\begin{verbatim}
\texttt{dc:title} The name of the collection
\texttt{dc:identifier} A formal identifier for the collection
\texttt{dc:description} A description of the collection
\texttt{cld:strength} An indication (free text or formalised) of the strength(s) of the collection
\texttt{cld:accessControl} A statement of any access restrictions placed on the collection including allowed users, charges etc
\end{verbatim}
We can see from this schema extract that it consists of

- DC 'standard' elements
- a 'namespace' schema, in this case the collection level description schema 'cld'
- refinements to the ‘standard’ definitions for DC elements which describe the particular type of resource

We would argue that the treatment shown in these examples is typical of what occurs when DC, or indeed other element sets, are used in practice. As mentioned before this is not new, but the opportunities offered by using the common syntax, RDF, increases the ease of combination and the possibilities for extension of element sets.

Having analysed what happens in practice, we propose a metadata schema architecture consisting of namespace schemas and application profile schemas.

4. Namespaces and application profiles

This paper suggests that we can distinguish 'namespace schema' from 'application profile schema'. Namespace schema contain all those elements defined by the managing body or registration authority (whatever that might be) for a particular namespace. Application profiles are tailored for particular implementations and will typically contain combinations of sub-sets of one or more namespace schemas.

'Namespace' is defined within the W3C XML schema activity (6) and allows for unique identification of elements. Within the W3C XML and RDF schema specifications, namespaces are the domain names associated with elements which, along with the individual element name, produce a URL that uniquely identifies the element. In W3C terms the namespace does not have to be a ‘real’ registration authority, nor does the element identifying URL need to point to a ‘real’ web address. However in order to ensure a well managed metadata environment we would argue that the namespace should refer to a real registration authority that takes responsibility for the declaration and maintenance of their schema.

There is a continuum of formality in such registration authorities from those where the authority is an internationally recognised standards body through to those where the authority derives from national or sectoral de facto standards, and at the other end of the continuum, to self-contained schemas defined within a local project or service.

By means of ‘namespace’ we can

- Identify the management authority for an element set
- Support definition of unique identifiers for elements
- Uniquely define particular data element sets or vocabularies

The DESIRE project constructed a prototype metadata registry schema with a data model within which 'namespace' consisted of three parts:

- Registration authority
- Namespace concept
- Namespace
It may be useful to consider how, in combination, these entities might help us to identify well managed metadata element sets. By use of these entities, a distinctive element set can be identified by a ‘namespace’, that namespace may have different instantiations over time (versioning) each of which require a separate namespace but all are associated with a namespace concept. A namespace concept, is therefore a grouping mechanism for successive versions of a namespace. Each namespace and namespace concept is associated with a registration authority. Within the DESIRE registry this enabled us to consider that one registration authority might have several different element sets associated with it.

5. What is an application profile?

Application profiles consist of data elements drawn from one or more namespace schemas combined together by implementors and optimised for a particular local application. Application profiles are useful as they allow the implementor to declare how they are using standard schemas. In the context of working applications where there is often a difference between the schema in use and the ‘standard’ namespace schema.

Schema application profiles are distinguished by a number of characteristics. They

- May draw on one or more existing namespaces

The application profile may use elements from one or more different element sets, but the application profile cannot create new elements not defined in existing namespaces.

- Introduce no new data elements

All elements in an application profile are drawn from elsewhere, from distinct namespace schemas. If an implementor wishes to create ‘new’ elements that do not exist elsewhere then (under this model) they must create their own namespace schema, and take responsibility for ‘declaring’ and maintaining that schema.

- May specify permitted schemes and values

Often individual implementations wish to specify which range of values are permitted for a particular element, in other words they want to specify a particular controlled vocabulary for use in metadata created in accordance with that schema. The implementor may also want to specify mandatory schemes to be used for particular elements, for example particular date formats, particular formats for personal names.

- Can refine standard definitions

The application profile can refine the definitions within the namespace schema, but it may only make the definition semantically narrower or more specific. This is to take account of situations where particular implementations use domain specific, or resource specific language.

By defining application profiles and, most importantly by declaring them, implementors can start to share information about their schemas in order to inter-work with wider groupings. Typically implementors are part of larger communities, they form part of a sector (education, cultural heritage, industry, government), possibly a subject grouping, they are part of programmes with common funding, they work with others serving the same target audiences. In order to work effectively these communities need to share information about the way they
are implementing standards. Communities can start to align practice and develop common approaches by sharing their application profiles.

Declaring profiles for application areas is a mechanism used elsewhere in computing. In other contexts, agreement on usage by means of a profile will be familiar to readers. For example within the area of resource discovery, Z39.50 application profiles have been used for some years, where implementors reach consensus on compliance with a sub-set of the Z39.50 standard. The Z39.50 Maintenance Agency (see last reference) defines a Z39.50 Profile as follows:

A profile specifies the use of a particular standard, or group of standards, to support a particular:
- application, for example GILS or WAIS;
- function, for example author/title/subject searching;
- community, examples: the museum community, chemists, musicians, etc.; or
- environment, examples: the Internet, North America, Europe, etc.

By "specifying the use" we mean to select options, subsets, and values of parameters, where these choices are left open in the standard.

A number of such profiles are maintained by the Z39.50 maintenance agency and are referenced from its web site, such as the CIMI profile for cultural heritage information, the Bath profile for library applications and resource discovery.

6. Examples

In order to illustrate the difference between namespace schemas and application profiles it may be helpful to refer to the DESIRE metadata registry where a few element sets have been treated in this way:

Examples of Namespace schemas
- DC 1.0
- DC 1.1
- DC recommended qualifiers
- BIBLINK
- cld
- ROADS Templates

Examples of Application Profiles
- BIBLINK Core
- DC Education Working Group proposed schema
- ROADS Document Template

A fully worked example of metadata created in RDF according to the RSLP collection description schema can be found by going to Andy Powell's RSLP collection description tool at http://www.ukoln.ac.uk/metadata/rslp/tool/ and clicking 'show example'.
Expressing the BIBLINK Core Application Profile in RDF Schemas

As part of the SCHEMAS project (7) we are encouraging people to publish their application profiles. Ideally we would like to use RDF schemas (9) since we would like to harvest distributed application profiles automatically.

We propose an expression of an application profile using the RDF Schema Specification syntax. Our example is of the BIBLINK Core application profile (10) which has the following characteristics:

- uses DC elements
- uses BIBLINK specific elements
- refines DC elements
- associates schemes with DC elements

The representation of this application profile in RDF schemas requires the following:

- a BIBLINK namespace (bc.rdfs) to declare BIBLINK specific elements (11)
- an RDF schema (bc-ap.rdfs) for the BIBLINK Core application profile (12)

(Note that it also requires DCMES in RDF schemas, which is not yet available).

Several "instance" records conforming to the BIBLINK Core application profile, bc-ap.rdfs, are available for reference (13), (14), (15).

7. What are the implications?

Application profiles will assist collaboration amongst namespace managers

Schema application profiles provide a basis for different metadata initiatives to work together. By focusing on the requirements of implementations, we see that there is a genuine need to facilitate the combining of ‘extracts’ from standard namespace element sets into application profiles.

Procedure and methods for declaring application profiles need to be agreed

There needs to be an easy way for implementors to disclose application profiles. By declaring application profiles implementors will assist inter-working between co-operating services. Both people and software need to be aware of metadata schema in use. Implementations that wish to work together can begin to share information about the details of their application specific schema, they can align their schema by way of a shared application profile. Software tools can go to application profile declarations in order to 'learn' how particular implementations are using metadata. This might assist in a metadata creation tool presenting the correct options to the user, it would assist in conversion of metadata and controlled vocabularies between applications, and so on.
The SCHEMAS (7) project is addressing this issue as part of its on-going work on providing support for schema implementors. SCHEMAS is funded by the European Community as part of its Information Societies Technologies programme and is providing a series of workshops to implementors to explore their requirements for sharing information about metadata schema.

Policies for metadata schema registries are required

Registries might exist at a variety of places and ‘levels’ as part of the infrastructure for supporting digital information management. Registries might be richly functional databases (the DESIRE registry is a prototype of such a registry), or they might be ‘thin’, merely providing links to schema declarations. Registries might exist at the namespace level (e.g. DC version 1.1) or registration authority level (e.g. DCMI). A registry might have an ambition to register all schemas associated with a namespace concept (e.g. DC) and all application profiles containing elements associated with that namespace. Or there might be separate registries for namespaces and for ‘communities of use’, the latter containing application profiles used by a particular implementor community.

Discussion on the role of registries is taking place within SCHEMAS, and more particularly it is an issue for the Dublin Core Registry Working Group (8).

8. Issues
How do we deal with conformance?

Dublin Core is flexible as regards conformance, albeit that conformance has not been defined in practice. Similarly MARC is flexible allowing for use of individual elements. But can individual elements from other element data sets be used in such a flexible way? Can an implementor take one or two IEEE LOM elements and combine them with Dublin Core?

The potential for parallelism and overlap

Application profiles might contain elements that overlap in their semantics. For example a simple form of an author and a more complex form. It might be argued that this is valid, in that a particular application might want to use such ‘overlapping’ elements for different purposes. For example a person's name as an unstructured data element might be used for searching purposes, whilst an structured name separated into elements for first name, second name, might be used for display. However obviously such overlapping and parallelism in use of elements would make manipulation and re-use of metadata more complex. In real implementations where large collections of metadata are being managed it seems more likely that dynamic mappings will take place from an underlying database according to the appropriate application profile for the operation in hand.

Specifying conventions and constraints on usage

There is a need for further investigation as to whether the likely syntaxes for expressing application profiles (RDF Schema, XML DTDs, XML Schema) have the means to specify rules for the content of elements, rules that do not exist in the vanilla namespace. For example
an application may want to make certain elements mandatory or it may want to specify that particular controlled vocabularies must be used for certain elements. (REFERENCE http://www.mailbase.ac.uk/lists/dc-general/2000-08/0043.html)

9. Conclusions

Taking existing implementation of metadata schema one recognises that rarely is ‘the complete standard schema’ used. Implementors identify particular elements in existing schemas which are useful, typically a sub-set of an existing standard. Then they might add a variety of local extensions to the standard for their own specific requirements, they refine existing definitions in order to tailor elements to a specific purpose, and they may want to combine elements from more than one standard. The implementor will formulate ‘local’ rules for content whether these are mandatory use of particular encoding rules (structure of names, dates) or use of particular controlled vocabularies such as classification schemes, permitted values.

We see application profiles as part of an architecture for metadata schema which would include namespaces, application profiles and namespace translations. This architecture could be shared by both standards makers and implementors. This architecture reflects the way implementors construct their schemas in practice as well as allowing for the varied structures of existing metadata schemas. We believe by establishing a common approach to sharing information between implementations and standards makers will promote inter-working between systems. It will allow communities to access and re-use existing schemas. And by taking a common approach to the way schemas are constructed we can work towards shared metadata creation tools and shared metadata registries.

Acknowledgements

We would like to thank Michael Day, Tracy Gardner, and Andy Powell and Tom Baker for discussions which led to the formulation of the ideas and concepts in this paper. Particular thanks to Carl Lagoze, Tom Baker, and Priscilla Caplan for their thoughtful comments on the initial draft.

References

1. DESIRE metadata registry: a prototype registry developed as part of the EC funded DESIRE project http://desire.ukoln.ac.uk/registry/
7. The SCHEMAS project home page is at http://www.schemas-forum.org/
8. Dublin Core Registry discussion list http://www.mailbase.ac.uk/lists/dc-registry/
10. The BIBLINK Core Application Profile used is at http://www.schemas-forum.org/registry/schemas/biblink/BC-schema.html
11. The BIBLINK namespace in RDF schemas is at http://www.schemas-forum.org/registry/schemas/biblink/1.0/bc-rdfs
12. The BIBLINK Core Application Profile expressed in RDF schemas is at http://www.schemas-forum.org/registry/schemas/biblink/1.0/bc-ap-rdfs
13. A record conforming to the BIBLINK Core Application Profile is at http://www.schemas-forum.org/registry/schemas/biblink/bc-ap-eg1-rdf
14. A record conforming to the BIBLINK Core Application Profile is at http://www.schemas-forum.org/registry/schemas/biblink/bc-ap-eg2-rdf
15. A record conforming to the BIBLINK Core Application Profile is at http://www.schemas-forum.org/registry/schemas/biblink/bc-ap-eg3-rdf
17. Examples of such conventions are given by Priscilla Caplan in a mail to the dc-general mailing list, see http://www.mailbase.ac.uk/lists/dc-general/2000-08/0043.html. Proposals for expressing these in XML Schema are suggested by Jane Hunter see http://www.mailbase.ac.uk/lists/dc-general/2000-08/0050.html

Rachel Heery and Manjula Patel
UK Office for Library and Information networking (UKOLN), University of Bath

Article Title: "Application profiles: mixing and matching metadata schemas" Author: Rachel Heery; Manjula Patel
Copyright and citation information  File last modified: Monday, 09-Oct-00 11:41:50 BST