Application Profiles for Learning

Erik Duval, Neil Smith, Marc Van Coillie
Katholieke Universiteit Leuven, Knowledge Integration, Ltd., Eifel
erik.duval@cs.kuleuven.be, neil.smith@k-int.com, marc.van.coillie@eife-l.org

Abstract

Application profiles enable “mixing and matching” metadata elements, in order to meet specific requirements for a particular context. As an example, some communities may want to make certain elements mandatory or restrict the value space of a particular element.

However, there is much confusion and only limited experience and expertise in the development and deployment of application profiles. That is why the CEN/ISSS Learning Technologies Workshop has decided to develop guidelines on the use of application profiles for (e-)learning. This paper highlights the main results of this work, and was co-authored by the three experts that drafted the CEN document [4].

1. Introduction

The goal of standardization is to produce a broadly acceptable specification which does not impose unnecessary restrictions that may mitigate against its wider uptake and use. The nature of standards dictates that they must cover every conceivable circumstance. On the other hand, implementers of a standard focus on the needs of their particular constituency and therefore choose subsets of possible options and interpretations, which, whilst conforming to the underlying standards, may limit potential for interoperability in future.

The normal way of addressing the need for interoperability is to define a profile of a standard. For example, the Library community in the 1990s saw wide interest in new standards such as Z39.50/ISO23950 (Information Retrieval). Users and vendors interested in implementing these standards set up Implementers’ Groups (ZIG) to work on interoperability issues. These groups developed profiles of the relevant standard and many of the profiles developed in the 1990s (e.g. the Bath Profile of Z39.50) are still in use today.

In this paper, we aim to provide guidance to communities of interest who have identified a need to interoperate. We use the term ‘Application Profile’ in a very broad sense to refer to any profile based on a standard or specification related to eLearning. In our work, we have focused primarily on Learning Objects and their associated metadata but we have also included examples of application profiles relating to learner information. An Application Profile is typically developed for a particular application with a particular constituency. Such a community may be large (for instance: the European Academic context) or small (for instance: a small enterprise in a particular domain).

Within the eLearning community we do not currently have mechanisms in place for formally approving ISPs or registering IRPs. However, the CEN/ISSS WorkShop on Learning Technologies is sponsoring work to put in place a registry of (Application) Profiles relating to eLearning. This could in future form the basis of some form of recognition system as defined in ISO TR 10000. What is apparent, though, from experience within the eLearning community to date, is that interoperability is normally best served by stating the requirements in an open and accessible manner.

An Application profile seeks to address the interoperability requirements between systems by:

• retaining conformance with a base standard or specification; and
• defining any new requirements in an open manner.

In section 2, we provide some further background on the role of application profiles. Section 3 is the core of this paper: it describes in some detail how application profiles can be developed in a methodological way. In our experience, the lack of such a methodology leads to a loss of efficiency and effectiveness in many organizations that are developing application profiles to meet their specific needs.

2. Background

As outlined above, metadata application profiles take one or more base standards or specifications as their starting point. The profile then imposes additional restrictions on this baseline. Modifications of this kind limit the options available to a subset of those available in the original specification (e.g. reducing a vocabulary list). The goal is to increase interoperability beyond the level of the base standard, without breaking
interoperability with those applications that are unaware of the profile. Consequently, a target system that conforms to the base standard will be able to process a metadata instance from a source that conforms to the application profile. Similarly, a target system that conforms to the application profile will be able to process a metadata instance that conforms to the base standard. However, in this case, the target system may be unable to process the values of elements based on the base specification but restricted in the application profile.

Not permitted are modifications to the base specification that break its conformance rules. Such modifications could result in the inability of a receiving system conforming to the base standard to process instances that conform to the profile.

3. How to develop an Application Profile

3.1. Start from your own requirements

The basic goal of an application profile is to support specific requirements of a particular context through a profile of a generic standard. In order to bootstrap this process, it is important to have an explicit understanding of those specific requirements. In the case of a metadata application profile, this means that a clear scope and purpose statement must be developed: in other words, what are the metadata going to be used for?

As an example: in the ARIADNE Foundation, the basic aim is to facilitate share and reuse of learning objects [1]. Metadata are used to enable end users to focus on relevant resources. Thus, the main purpose of metadata in the ARIADNE Knowledge Pool System is to facilitate search, and the main scope is a learning context. As another example, UK LOM CORE [2], has been developed from a slightly different stand-point. It aims to identify common practice and provide guidelines that are optimized for use within the context of UK education. However, it is not indented to be too restrictive in nature. Instead, it is designed to be used as a basis for other, narrower, communities of interest (see for example, the JORM profile [5]).

A particularly effective way to elicit requirements is the definition of so-called use cases that describe how an end user would make use of the application to be developed. From such a description, requirements can be deduced for the application. From those application requirements, metadata requirements may be deduced. For instance, a use case might mention that end users are able to filter the available resources so that only those in their mother tongue are retained. This interaction requirement strongly suggests that a metadata element should be included that describes the language of the resource. As another example, a requirement to identify learning resources which are available free of charge might be identified in a use case. This would lead the application profile developers to look closely at the ‘cost’ element within LOM to see how this can be achieved.

3.2 Selection of data elements

Once the requirements are clarified, a first important decision in the actual development of a metadata application profiles is the selection of data elements that the application profile will be built from. Often, the profile developers will start from a metadata schema that has a scope and purpose similar to that of the application profile:

- for a simple, generic context, the Dublin Core Metadata Element Set might suffice;
- in the case of more complex audio-visual resources with a generic context, the MPEG-7 metadata schema may be more appropriate as a starting point;
- in a learning context (be it schools, academic, corporate or military), the IEEE LTSC Learning Object Metadata (LOM) standard typically provides the basic data element set to start from.

As mentioned before, we will focus in this paper on LOM based application profiles. In practice, application profile developers often start from a simple spreadsheet where the rows correspond to the standard data elements.

3.3 Size and smallest permitted maximum

Values for some data elements may be allowed to be present multiple times in one metadata instance: in LOM, this is defined through the size of the data element. If the size equals one, then the data element can only have one value in an instance. If the size is more than one, then the data element can have multiple values in an instance. In that case, a smallest permitted maximum (spm) is defined: this is the smallest number of occurrences of a field that an application should support when reading, writing or otherwise processing metadata instances.

As a general rule, an application profile can reduce the size of a data element, or keep it equal to the value in the base standard. An application profile cannot increase the size of a data element.

If the size of the data element is one in the base standard, then the application profile can:

- Reduce the size to zero: in that case, the data element is not present in the application profile. As an example, the size of the data element 1.6:General.Coverage is reduced to zero in the ARIADNE application profile.
- Maintain the size of one: in that case, the data element can have a value in an instance of the application profile.

For data elements with a size larger than one in the base standard, an application profile can:
- Reduce the size to zero: in that case, the data element is not present in the application profile and the smallest permitted maximum is no longer applicable.
- Reduce the size to one: in that case, the data element is still present in the application profile, but it can only have one value in a metadata instance. The smallest permitted maximum is no longer relevant.
- Reduce the original size, but not below two and
  • Maintain the original smallest permitted maximum, if that is not larger than the new size.
  • Reduce the smallest permitted maximum: though this is not encouraged, it doesn’t formally break interoperability, as the smallest permitted maximum does not impose a strict requirement.
  • Maintain the original size of larger than one and
    - increase the smallest permitted maximum: in that case, the data element can still appear multiple times, and applications are encouraged to process more instances of the values than the base standard suggests.
    - Maintain the original smallest permitted maximum: in that case, the application profile is identical to the base standard.
    - Reduce the smallest permitted maximum: though this is not encouraged, it doesn’t formally break interoperability, as the smallest permitted maximum does not impose a strict requirement.

As an example, the ARIADNE application profile is more restrictive than the LOM base standard, as it excludes the data elements 1.6:General.Coverage and 1.7:General.Structure. Moreover, it only allows one instance of the data elements 1.1:General.Identifier, 1.3:General.Language and 1.4:General.Description, whereas the LOM standard allows multiple values for those data elements. The EUN application profile is identical to the LOM base standard.

3.4 Data elements from multiple namespaces

In principle, an application profile can be based on more than one base metadata schema. In one sense, the LOM standard itself illustrates this, integrating for instance the complete vCard schema to describe people or organisations.

However, there seems to be very little practice doing something similar for a particular metadata application profile.

3.5 Adding local data elements

Besides mixing and matching data elements from several base standards, an application profile may also include local data elements.

For example, LOM contains a single optional element (6.1) entitled cost with a value space of ‘yes’ or ‘no’. UK LOM Core gives this element a status of ‘recommended’ and adds an implementation guideline “If "yes", details of the actual cost should be included in 6.3 Rights. Description”. The Curriculum Online application Profile goes beyond this and makes the element mandatory (http://www.curriculumonline.gov.uk/). It also mandates that, if the value is ‘yes’ then a number of local data elements are also completed. These data elements cover pricing information (including a controlled vocabulary for license model, cost and currency) and support information.

3.6 Obligation of Data Elements

Once the full set of metadata elements to be included in the application profile has been decided upon, the status of these data elements can be defined. Typical options include:

- Mandatory: a value for the data element shall always be present in a metadata instance;
- Conditional: if a certain condition is satisfied, then a value for the data element shall be present in a metadata instance; if the condition is not satisfied, then a value for the metadata element may or may not be present. As an example, in COL, either the author or the publisher must be included in every metadata instance. In other words, if the author is not included, then the publisher must be included; if the publisher is not included, then the author must be included.
- Recommended: Some application profiles recommend including values for specific metadata elements.
- Optional: a value for the data element may or may not be present in a metadata instance;

As always, an application profile can impose more stringent obligations on data elements than the base standard does. An application profile cannot relax such obligations: for instance, a mandatory element cannot lose its mandatory status in an application profile. In the LOM standard, all data elements are optional: that means that application profiles can make some of the data elements mandatory or conditional, based on specific requirements.

Note that the ‘Recommended’ status of a data element does not imply any strict requirement. It should be viewed more as guidance to implementers than as a hard obligation. Therefore it is possible for an application profile to alter the status of ‘Recommended’ data elements to ‘Optional’, though such an approach is not encouraged.

3.7 Value space

Besides the obligation of the data elements, their value space must be defined. In this context, the value space defines the set of values that the data element
shall derive its value from. Again, the application profile may be more restrictive about the value space of a data element than the base standard is; it cannot be less restrictive.

In LOM, a value space is typically defined through one of the options below:
- A vocabulary: in that case the set of values is enumerated.
- A reference to another standard (e.g. ISO/IEC10646-1:2000) or specification (e.g. vCard)

The application profile can restrict the value space through the corresponding options below:
- A vocabulary can be restricted to a subset of the vocabulary in the original base standard.
- A reference to another standard or specification can be replaced by a reference to an application profile of that other standard or specification.

As an example, in ARIADNE, the values for LOM data element 5.1:Educational.InteractivityType are restricted to “active” or “expositive”, i.e. the value “mixed” is not retained from the LOM value space. As another example, in the Europortfolio ePortfolio Content packaging Application Profile, the title for the main items in the organization of the manifest has been restricted to use only five possible values from “PortfolioParts”, “Owners”, “Views”, “Presentation”, and “Relationships”.

### 3.8 Relationship and dependency

More complex inter-relationships and dependencies between data elements can also be defined in an application profile.

As always, the application profile may be more restrictive about such inter-relationships than the base standard is; it cannot be less restrictive. As the LOM standard does not include such restrictions, a LOM application profile can include arbitrary such restrictions.

As an example, in the ARIADNE application profile, the value space of 5.2:Educational.LearningResourceType depends on the value of 5.1:Educational.InteractivityType: if the value of the latter is “active” then the value space for the type element is restricted to “exercise”, “simulation”, “questionnaire”, “exam”, “experiment”, “problem statement”, and “self assessment”. If the value of interactivity type is “expositive”, then only one of the other values of the value space in LOM for resource type is allowed.

As another example, in the LOM FR Application Profile, the value space of 4.4.1.2:Technical.Requirement.OrComposite.Name depends on the value of 4.4.1.1:Technical.Requirement.OrComposite.Type: if the value of the latter is “operating system” then the value space for the name element is restricted to “pc-dos”, “ms-windows”, “macos”, “unix”, “linux”, “multi-os”, and “none”. If the value of type is “browser”, then the value space for the name element is restricted to “any”, “netscape communicator”, “ms-internet explorer”, “opera”, “amaya”, “firefox”, and “safari”.

### 3.9 Data Type profiling

In the LOM standard, the datatype “indicates whether the values are LangString, DateTime, Duration, Vocabulary, CharacterString, or Undefined” (LOM, p.4).

In effect, the data type in LOM is a metadata schema in its own right. All the rules defined above for application profiles of metadata schemas are thus also applicable to data types.

### 3.10 Application Profile Binding

The above sections deal with conceptual restrictions on application profiles. The general rule is that an application profile can be more restrictive than the standard it is based on, not less restrictive. In this way, any instance of the application profile is by definition also an instance of the base standard and interoperability between different application profiles of the same standard is ensured.

There is a similar rule on the level of a binding of an application profile, for instance in XML or RDF: the goal is to make sure that any instance that conforms to the relevant binding of the base standard also conforms to the binding of the application profile.

### 3.11 Conclusion

In this section, we analyzed step-by-step how application profiles can be defined.

It is important to note that the requirement to include a value for a particular data element from a particular value space in a metadata instance does not imply that this value must be provided as such by end users [3]! As an example, in ARIADNE, the data element 1.1.1:General.Identifier.Catalog is mandatory. However, end users never see it, never provide it, and can in no way make use of it. The value of that data element in ARIADNE is always “Ariadne”. Its only function is to indicate the source of the learning object and the metadata when they are exposed to external infrastructures.

### 4. Cascading Profiles

In many cases, authors of an application profile will take as their starting point not a base specification but an already existing application profile of that specification. As the focus of the community producing a profile increases, it is possible to have several ‘layers’ of refinement with each successive layer representing a narrower community of interest with a more specific focus. A metadata instance that conforms
to a narrow profile must also conform to each of the higher level profiles on which it is based, including the base specification.

Cascading profiles - example

Where the situation becomes potentially more complex is where a community wishes to develop a profile, which is based on two or more parent profiles whose only common reference point is the base specification, or a profile thereof. In the hypothetical situation illustrated below, the Oxford University Computer Science Department wishes to develop an application profile of IEEE LOM which is conformant to both the ACM Computer Science profile and the Oxford University profile, which is itself based on the UK HE profile which, in turn, is based on UK LOM Core,

5. Conclusion


We hope that this methodology will enable the numerous communities that are developing their own application profiles to do so in a more rigorous, precise, effective and efficient way.

References