One Standard For All:

Why We Don't Want It

Why We Don't Need It

Stephen Downes
National Research Council

January 17, 2003

1. Why Standards?

With the help of Norm Friesen's *What Are Educational Objects* we can approach a list of some objectives:

- *Discoverability* – metadata standards make it easier to *find* learning objects
- *Interoperability* – metadata standards help learning objects work together

http://www.careo.org/documents/objects.html

1.1 Discoverability

“What makes objects "discoverable", "accessible" or "searchable" is the Metadata used to describe and categorize it. Metadata, "structured data about data" (DC doc), works much like information provided in a library catalogue.”

1.1.1 Searchable Information

“Instead of describing the data contained in books and journals, this metadata provides searchable, standardized information about digital objects --data such as authorship, subject classification, format, size, delivery requirements, or interactivity level.”

1.2 Interoperability

“With its close relationship with/dependence on these standards, educational objects would be "in many ways the crowning achievement of the standards initiative" --as Bryan Chapman says. ‘You can't have interoperable learning objects without industry-wide standards.’”

1.2.1 Information Exchange

"…integration is possible only with open protocols, which allow an organization or system to exchange information with suppliers, partners, and customers in a format that accommodates each organization's system."

1.3 Why We Need Standards

Looking at the two major areas in which standards are used, it seems clear that standards are intended to help us in two major ways:

1. To describe
2. To communicate

But wait a minute. Isn't this what we use *language* for? Of course it is!

1.3.1 Standards and Machines

It is easy to misunderstand the need for standards because we are working in a technological domain. It is easy to imagine that standards for learning objects are like standards for machines.

But we are not machines.

1.3.2 Standards and Semantics

The use of standards in technology is to establish a physical connection. That is why precision is important.

But the use of standards in learning is to establish a semantical connection. The need for precision is much less, and indeed, it is a certain degree of flexibility and vagueness that makes it work at all.

2. We Don’t Need Standards

There are several ways to approach this topic:

1. By analogy
2. Through an understanding of language
3. Through an examination of XML

I will proceed through each approach in turn, making the same point each time.

2.1 Analogy: Roads and Rail

The objectives of roads and railway systems is exactly the same: to move people and goods from one place to another. So why do we have so many more roads than we have railways?

2.1.1 Roads and Rail (2)

The answer, of course, is that roads are much more tolerant than rail.

To run along a rail, a train must be precisely defined. Even a small variation in wheel width would cause disaster.

A road, by contrast, allows for a great variety in the types of vehicles allowed.

2.1.2 Intent

More to the point: in a railway system, the purpose and intent of the system is defined by the system. You could never use a railway system for a 20 km road race, for example. But the purpose and intent of a network of roads is defined by the users of the roads. Roads are used for a variety of purposes never imagined by their creators: road races, road hockey, cars, buses and 18 wheelers.

2.1.3 Interpretation

A railway can never be anything more than it was designed to be.

A road, by contrast, can be whatever we interpret it to be. A road doesn't impose a structure or even a use upon us. It is merely a flat surface. What a road can be used for varies according to our various needs for flat surfaces.

2.2 Language

One reason language works for description and communication is that it allows for precision. You and I both mean the same thing by the word "apple."

But the other reason language works for description and communication is that it allows for vagueness. You can mean something very different from "apple" than I do (and yet we can still exchange ideas).

2.2.1 Language and Meaning

In a nutshell: *words do not have fixed meanings*

You and I can use the same words to refer to different sets of objects.

You and I can use different words to refer to the same set of objects.

This is true *even if we speak the same language*.

2.2.2 Examples (1)

What does “Paris” mean to you?

- The capital of France
- Good cooking
- A type of plaster
- A Roman God

This is *not* a random use of the same word to stand for different things. All of these uses are connected.

2.2.3 Examples (2)

Where is Edmonton?

- In Alberta, Canada
- In school division number 1
- In county 42
- In the Palliser Triangle
- On the Prairies

This is *not* a consistent system of classification. We use *multiple systems* to designate even something as simple as Edmonton’s location.

2.2.4 Interpretation

Language allows us to *mean* different things with the same concept, and to use different concepts to mean the same thing.

If standards were *required* for description and communication, then language would not be able to make description and communication possible at all.

But language is a lot like a road: its use is not determined by its structure, but by its users.

2.3 XML

XML is viewed by many as the Holy Grail because (they believe) it gives us a standardized way of describing things. Take an entity, assign it a canonical set of properties, give each property one of a set of canonical values, and you're done.

But what is the basis of XML?

2.3.1 XML and Description

Look at it this way: why do we even use XML at all? Well, to describe objects, of course.

But why not simply use a relational database? It does everything we need. Each entity is a row. Each property is a column. Values of columns in one table can refer to rows in another table.

2.3.2 XML and Properties

There is much more to describing an object than merely ascribing to it a collection of properties and values.

How we describe an object depends on the context in which we wish to use that description.

Knowing that Edmonton’s “location” is “Canada” is useless to someone wondering whether they can grow a garden in the city.

2.3.3 XML Statements

XML represents more than a list of properties of a given entity. An XML document is a *series of statements* about an entity.

<name>Joe’s Garage</name>

means “The name of this entity is Joe’s Garage.” The tags and syntax create a punctuation of statements. We could have used certain words and periods to obtain the same result.

2.3.4 Interpretation

In XML, a statement means what you want it to mean (that's the whole point of being eXtensible). The meaning – and the possible meanings – of a statement depend on the *context* in which it is used. In different contexts, we would want a wide variety of statements about the same entity. XML provides us with the means of making those statements.

3. Learning Object Metadata

Let’s pull these concepts together…

We want to be able to describe learning objects, and to allow them to communicate with each other.

In order to do this, we need a language.

But for this language to be *useful*, we need a language that is extensible, that depends as much on *context* as it does meaning.

3.1 Range of Description

So what does IMS, say, tell us about our language?

It allows 157 (or whatever) tags to describe our learning object.

That means we can make only 157 statements about our learning object.

3.1.1 Describing Goodness

- How many ways are there to say that an object is good or bad (hint: more than 157).
- An object may deserve a <thumbs>up</thumbs>
- It may be <AMA>certified</AMA>
- It might be <LDS>approved</LDS>
- It could be <GC>current</GC>
- … and on and on

3.1.2 Describing Category

What are the many ways to classify an object? Sure, IMS allows a reference to an external taxonomy.

- What about non-hierarchical taxonomies?
- What about dynamic categories?
- What about cross-classification?

There is no one way to classify. I don't mean, not one system of classification. Rather: not one method of classification.

3.1.3 Describing Use

Learning objects are intended to play a *role* in a *system*. They have (as Norm Friesen reminds us) a pedagogical purpose.

But we need a range of vocabulary that allows us to specify:

- Non-pedagogical roles for leaning objects
- Pedagogical roles for non-learning objects

3.1.4 Description: Reprise

Examples could be multiplied. But the point is this:

- It makes no sense to try to establish the vocabulary of description *in advance*
- Rather, the appropriate vocabulary of description will be established *by its use*
- Learning object metadata, therefore, becomes “standardized” *after the fact* and only in the context of particular applications

3.2 Communication

Very much the same sort of point could be made with respect to communication among learning objects and instructional management systems.

What would leaning objects want to say to each other? “I’ve started.” “I’ve stopped.” “I have an error.”

Sure, but how about: “The subject’s pulse is 98.”

3.3 Object Ecology

The point of this section is to establish that learning objects do not exist in isolation. They belong to an *ecology* of related objects.

This ecology includes people, organizations, classrooms, equipment, and more.

Even within the realm of learning objects, there is an ecology of courses, classes, lessons, exercises, tests, examples, discussions…

3.3.1 A Learning Environment

For example: what’s the best way to designate an *author* of a learning object?

Surely not with a *text string!*

The author of a learning object is a separate entity, with a wide variety of properties (including, but not only, a name).

3.3.2 Learning Entities

Indeed, almost everything that could be said of a learning object refers to some sort or another external entity.

Who is the publisher? Where is the object stored? Who has approved the object? Who manages the sale and rights? What is its classification? What format is it in?

3.3.3 Referencing Entities

There are various ways to refer to external entities (of which the most popular is RDF).

But it’s easiest to think of it this way:

<author>URI</author>

The URI is the XML file describing the author. The entry may be the actual URI, or a pointer to a location where the URI is stored (such as an object database).

3.3.4 Describing People

Now the question becomes: why would learning object metadata have anything to do with describing people?

And obviously, it doesn't.

Indeed – we should be wondering whether there is any such a class of objects as "learning objects" at all!

3.3.5 Learning Objects?

Let me say it in its controversial form first:

There’s no such thing as a learning object

Less contentiously:

Learning object metadata is simply *one way* of describing objects used in learning

Huh?

3.3.6 Objects as Objects

Think of learning objects *as objects* first. Describe them in the manner appropriate to their nature. Thus, images (for example) would contain attributes PDF files do not. PowerPoint slide decks would contain attributes that classrooms do not.

Describe each object in its own way using some (external) metadata specific to that object.

3.3.7 Learning Metadata

On this view, learning object metadata is thus a (loosely defined) set of tags which:

- are applied in specific educational contexts
- vary according to that context
- describe the *educational* properties of an object
- and in a manner specific to the (potential) user

4. Interlude

Philosophical foundations for this interpretation of metadata (or, I’m not just making this up).

Language (and therefore metadata) has three dimensions (Charles Morris):

- Syntax – sentence structure and grammar
- Semantics – reference and representation
- Pragmatics – context of use

4.1 Pragmatics

Pragmatics is the idea that the exact same sentence, with the exact same signification, can mean different things to different listeners.

"The apple is red" may signify a denotation of colour to a child, and it may signify an assessment of ripeness to an apple picker. It may, if we have anthropomorphized the apple, signify that the apple is embarrassed.

4.1.1 Naming

W.V.O. Quine ('On the Indeterminacy of Translation', Word and Object): there is no fixed determination of *names*.

Example: does the term *Gavagi* refer to a rabbit, an adult rabbit, potential food…?

We can at best construct working hypotheses as to what people mean when they apply a name to something. This is inevitable and inescapable.

4.1.2 Causaility

Norwood Russell Hanson (Patterns of Discovery) – there is no single thing that is the *cause* of something else.

Example: what was the cause of the accident? To the driver, the judge, the road designer, the meteorologist, it’s a different thing.

4.1.3 Explanation

Bas C. van Fraassen (The Scientific Image) – there is no single thing that constitutes an *explanation* of something.

All explanations are expressed, “X happened *instead of Y* because of Z.” The counterfactual, ‘instead of Y,’ varies according to your point of view.

Example: why did the rose grow?

4.1.4 Categorization

George Lakoff (Women, Fire and Dangerous Things) – there is no ‘natural’ system of categorization. Pleople classify objects according to their relation to the object.

Example: ‘Women, fire and dangerous things’ as a single category.

4.2 … and Metadata

We can conclude this:

We say different things about an object depending on our different relations (eg,. Contexts of use) of an object

And thus:

The more we restrict what we can say about an object, the more the meanings of the terms we do use will vary according to context.

4.2.1 The Lesson

The lesson is this:

If we attempt to restrict the vocabulary used to describe learning objects, then because of pragmatics we are almost guaranteeing that the words in our vocabulary will lose their fixed meaning.

This will make it impossible for *machines* – as well as humans – to understand what is being said.

5. Why We Don't Need It

What is it that we don't need?

- A canonical vocabulary to describe (so-called) learning objects

What do we need instead?

- A way to express statements describing the learning-related properties of objects

5.1 Statements

All statements have the following (basic) form:

<Subject><Property><Object(s)>

The ‘subject’ is the entity we are discussing.

The ‘property’ is the quality or relation we are ascribing to the object.

The ‘object’ is the value of that property or relation.

5.1.1 Statements in XML

In XML, each ‘layer’ typically refers to a distinct object:

<item about=http://wherever.com/item.xml>

<color>Red</color>

</item>

(Notice how the form leads us to believe that our statements consist only of two elements, the property and the value).

5.1.2 Identifying Objects

There is no good way (yet) to identify unique objects in XML. The typical method (using a URI) is not satisfactory because URIs change. Ideal would be an *object database* assigning a unique identifier to each object and associating that identifier with a (current) URI. A good object database would also reference an *owner* (who would have the last word on the URI).

5.1.3 Storing Statements

Think of an XML database as containing, not records with values, but rather, a series of statements.

Thus each record would look like this:

Entity: 234553222

Property: color

Value: red

In other words, the object, property and value are not part of the database structure, but variables within that structure.

5.1.3 Storing Statements (2)

In practice, we need more information about each statement:

- Key: identity of the statement
- Language: human language of the statement
- Schema: vocabulary used
- Entity: the entity identifier
- Property: the property or relation being described
- Value: the value of that property or relation
- Source: who said this statement

5.2 Objects

What is an object?

On this view:

> An object is anything that can have properties.

To find the objects in your metadata (there may be several), locate the tags with subtags. The presence of subtags indicates that there is some thing being referred to that has properties. This thing is an object.

5.2.1 Representing Objects

In an ideal world, each object will be described in a separate file that can be referred to via a URI or object index.

That probably won't happen. Objects will also be described *implicity* through reference to its unique properties (in a sense, referring to an object by its URI is one way of implicitly referring to an object).

5.2.2 Categories of Objects

A *category* of objects is a set of objects having the same properties. For example, we may designate an object as being a “lesson” by virtue of it having a duration property within a specified range.

A *category* is therefore a form of shorthand for referring to a specific set of of objects (that is what makes categories useful for searches).

5.2.3 Types of Objects

A *type* of object is an object that may have a certain property.

For example, an object might be considered to be a “leaning object” if it has a duration. Something that does not have a duration, on this account, is not something that can be a leaning object.

(I never said it was a good account.)

5.2.4 Identifying Types

An object is identified as being of such-and-such a type by being described with a specific set of properties.

Therefore, an object may be identified as being a certain type by identifying the *schemas* used to define the set of properties describing the object.

A schema defines an object type as a set of (possible) properties an object may have.

5.2.5 Some Major Types

Major types of objects include:

- People
- Organizations
- Documents
- Images
- Buildings
- Devices

5.2.6 Multiple Types

It should be evident that a given object may belong to more than type (and therefore, more than one category).

A web page, for example, may be a document, a learning resource, an opinion on the war, a directory, a home page…

A person may be a person (properly so-called), a director, a detractor, a learning resource, a judge…

5.2.7 Object Types

An object becomes a given type of object by being described in terms relevant to that type. Any given object will be described using multiple sets of terms, because any given object may be any number of types.

There is no *a priori* means of determining what type of object may also be of another type. That is why almost anything (and not just web pages) can be learning objects.

5.3 Properties

A property is a *quality* or *relation* possessed by an object. Though we designate properties with names, properties don’t exist in their own right (in other words, properties aren’t objects).

5.3.1 Property Types

There is no (and could never be) one standard ‘canonical vocabulary’ for describing properties.

The same property (e.g., colour) may be described in different ways with differing precision in different vocabularies (e.g., hue, tint, shade, wavelength, 24 bit pixel colour).

The choice of vocabulary depends on the context of use.

5.3.2 Property Values

The value of a given property may be another object or it may be an index value on a range of possible values.

We could say: if the value is another object, then the property is a relation, and if it is an index value, then the property is a quality.

5.3.3 Naming Properties

In language, we typically refer to a property value using a *name* designated for that purpose. Hence, we may see the colour of an object described as “red” and the author of an object described as “Fred Smith”.

But these names have no inherent meaning except insofar as they are fixed as a reference to an external object or index value.

5.3.4 Imprecision

Many metadata specifications use strings as through they could (without an external reference) represent a fixed entity or index value.

But this is a fallacy. Using a string, for example, to attribute an author to an object builds in a certain degree of imprecision.

Strings should only be used for proper names referring to the *current object.*

5.4 Finding Objects

When searching for an object, we need to ask:

- What *type* of object are we looking for?
- What *category* of object are we looking for?

(We may be looking for objects of multiple type, of multiple category.)

By specifying type and category, we identify a subset of all possible objects.

5.4.1 Specifying Type

We specify type by selecting the schemas we wish to use in our search.

To search for learning objects, for example, we would look for objects described using CanCore metadata.

Sometimes similar typs are designated using overlapping schemas. We might, for example, search for objects described using either CanCore or SCORM metadata.

5.4.2 Specifying Type (2)

We may specify type *explicitly* or *implicitly*:

Explicitly, by naming or selecting the schemas we wish to be considered

Implicitly by searching collections of objects containing only a certain type of object (a CanCore-specific repository, for example)

5.4.3 Specifying Category

We specify the category of object we seek by identifying a range of allowable values of given properties. Any given search on a given type of object is a string of property-value pairs.

We obtain precision in searching by requiring more specific property-value pairs.

5.4.4 Search Practicalities

It is a general rule that you can't make a fuzzy search inherently precise. If a person searches for 'objects about Rome' then no mechanism exists to narrow th search.

Searches must therefore be made more precise. This may be enabled in two ways:

- Default search preferences
- Context-specific parameters

5.5.5 Default Parameters

A *default search parameter* is a parameter intended to be applied automatically for a group of searches.

For example, a searcher may opt to to view only those objects that are in English, free, LDS approved, and in HTML format.

Default search parameters are properties of the *searcher* applied to select properties in the set of desired objects.

5.5.6 Contextual Parameters

A contextual parameter may be set by detecting the context in which a search is being performed. For example, an instructional designer looking for an object to include as a lesson will be interested only in learning objects of a given length. A well-designed authoring environment will detect this, and add a length stipulation to the search.

5.5.7 Lessons

Search precision will not be obtained by limiting *a priori* the type and categories of objects being sought. This is because it is impossible to determine *a priori* what parameters may be relevant in a given context.

Instead, search precision is obtained by considering the *context* of the search, and specifically, who is performing the search, and where they are performing the search.

5.6 Communication

I have left the topic of communication between learning objects to one side because most of what holds for describing and locating objects also applies to communications between objects.

For after all, a communication between objects is nothing more or less that a set of statements attributing values of properties to objects.

5.6.1 Messages

In order for communication between objects to be possible, we need nothing more than to specify the location of any message sent by one object to another.

There is no need to restrict objects to a canonical vocabulary.

6. Concluding Remarks

Objects are best described using multiple vocabularies.

There is no way to determine which vocabulary will be relevant to either an author or a user of a given objects.

Trying to stipulate a canonical vocabulary *a priori* needlessly reduces the effectiveness of a system of communication.