Approaches To Validation Of Dublin Core Metadata Embedded In (X)HTML Documents

Brian Kelly / Pete Johnston / Andy Powell, UKOLN, University of Bath, Bath, UK
<B.Kelly@ukoln.ac.uk> <P.Johnston@ukoln.ac.uk> <A.Powell@ukoln.ac.uk>

ABSTRACT

This paper describes approaches to validation of (X)HTML-embedded Dublin Core metadata. It takes the view that useful validation must accommodate the particular ways metadata elements are used in the context of a service or application.

1. Background

UKOLN, a national centre for expertise in digital information management based at the University of Bath, UK has been involved in metadata activities for several years, including participation in the Dublin Core Metadata Initiative (DCMI) [1].

The Dublin Core Metadata Element Set (DCMES) provides a simple set of metadata elements that can be used to support resource discovery across a wide range of resource types, and the DCMI has specified a set of conventions for encoding DC metadata using the meta element of (X)HTML [2, 3].

The creators or publishers of (X)HTML documents embed metadata in their documents on the basis that that metadata is to be extracted or harvested by one or more agencies who build services using the metadata aggregated from multiple sources ("service providers"). The encoding conventions specified by DCMI are intended to support semantic interoperability between these two parties.

In practice, implementers build services on the use of DC metadata in a form that is tailored to the requirements of their application. This may mean specifying the use of a subset of elements from the DCMES and/or specifying that elements should be used in particular ways (and perhaps even using DC elements in association with elements drawn from other standard metadata element sets). Researchers have developed the concept of the "application profile" to describe this approach of "optimising" the use of standard metadata elements for a particular context [4]. Some profiles may indeed be specific to a single application; others may be sufficiently generic to be deployed by many different services.

2. Metadata Validation

The services which **use** metadata embedded in (X)HTML documents are typically developed and managed independently of the processes which **create** that metadata. While service providers probably do perform some checking of the harvested metadata records, that process is typically performed by a different agency from the creating agency, perhaps long after the metadata was created. If the results of validation are to be useful to the metadata creator, then they must be available at the time of creation.

Syntactic validation of (X)HTML resources is regarded as important in order to ensure interoperability. In practice, many authors do not perform validation of their documents, preferring to use the display in one or two browsers to test their pages. An approach based on such visual checking is of little value for checking embedded metadata since it is not usually displayed by Web browsers.

Even if an (X)HTML document **is** validated against a standard SGML/XML DTD or XML Schema, that process checks only the basic syntax of the meta element. (X)HTML does not define a fixed set of metadata properties. The names and values of metadata properties are encoded in SGML/XML attribute values, and syntactic validation. confirms only that the attributes required are present and that the values provided conform to the type constraints specified.

More helpful to the metadata creator is a level of validation which checks that their DC metadata conforms to the DCMI-recommended encoding conventions. And when preparing metadata for use within a specific application or service, ideally the creator should be able to check their metadata against the rules of the specific "application profile" used by that service.

3. DC-dot – A Simple Perl-Based Validator

DC-dot [5] was developed at UKOLN as a simple Web-based authoring tool for Dublin Core metadata. It enables Dublin Core metadata to be created and edited and stored in several formats including HTML, XHTML and RDF/XML.

When metadata is already embedded in an (X)HTML document DC-dot extracts that metadata and as part of that process provides a basic validation function. It will identify errors (e.g. use of DC.Author rather than DC.Creator) and give warnings (e.g. flagging that the case of Dublin Core elements may change in the near future).

Use of DC-dot to validate Dublin Core metadata embedded in a HTML resource is illustrated in Figure 1.

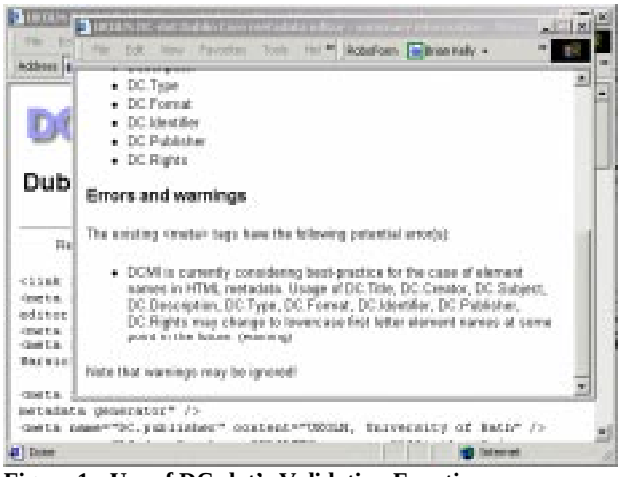

Figure 1: Use of DC-dot's Validation Function

4. Using An RDF Validator

Another approach which has been used is to dynamically convert Dublin Core metadata embedded in HTML resources into RDF/XML format using W3C's Dublin Core Extraction Service [6] and then pass the RDF/XML to W3C's **RDF Validator** [7]. Documents in HTML format can be dynamically converted to XHTML using the online Tidy service [8].

The RDF Validator provides information on RDF errors, together with a graphical display of the RDF relationships, as shown in Figure 2.

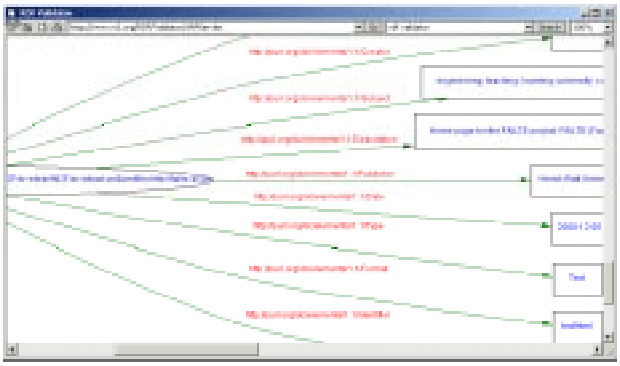

Figure 2: Use of W3C's RDF Validator

5. Limitations Of These Approaches

The two approaches described provide some support for validation of Dublin Core metadata.

In the first case, DC-dot performs basic syntactic checking and confirms that metadata elements indicated to be DC elements are indeed elements from the Dublin Core Metadata Element Set.

In the second case, the visual representation of DC metadata may be a useful aid to the author. However, the validation process is checking the syntax of the RDF/XML generated by the extraction service which itself performs only limited checking of the input. Further, since the RDF Validation Service is a generic RDF validator, its output is given in RDF terminology, which can be difficult to interpret. In addition as it is a general purpose RDF tool it is not possible to use it to validate Dublin Core specific features.

6. dcmeta: an XSLT Approach

In order to address these limitations UKOLN has begun development of a Dublin Core validation tool using an XSLT stylesheet. The tool is known as **dcmeta** [9].

Taking an XHTML document as input, the stylesheet generates a simple (XHTML) report on the DC metadata embedded in that document.

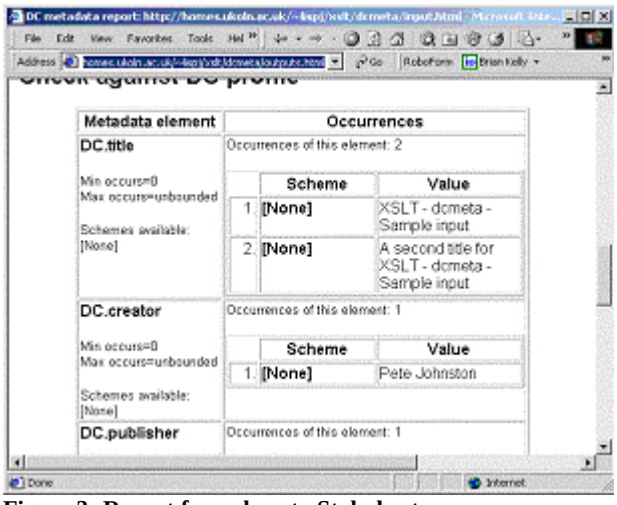

Figure 3: Report from dcmeta Stylesheet

It checks that the metadata embedded in the input document follows the general conventions recommended by DCMI, and it further checks against a particular application profile of the Dublin Core Metadata Element Set.

A "profile" in this context is simply a set of rules which specifies:

- which DC properties are permitted (e.g. only the 15 "elements" or some subset of the larger set of elements and element refinements);
- minimum/maximum permitted occurrences of a specified property (e.g. metadata should include exactly one occurrence of DC.Title)
- "encoding schemes" permitted for a property (e.g. DC.Subject properties should carry scheme = "LCSH")
- permitted values for a property (e.g. DC.Publisher should have the value "UKOLN")

The "profile" is specified as an XML document that forms a secondary input to the stylesheet. At present, the stylesheet caters only for the checking of Dublin Core metadata elements, not elements drawn from other element sets.

7. Deployment

An XSLT engine expects to read a well-formed XML document, so it may be necessary to pass the input document through Tidy as a preliminary step.

The stylesheet can be deployed by the metadata creator using the XSLT engine of their choice.

For example, it can be used with the MSXML XSLT engine provided as part of Microsoft Internet Explorer. The use of a Javascript "bookmarklet" provides a mechanism for invoking the transformation on the currently displayed XHTML document.

The stylesheet could also be deployed in a Web-based service, with the URL of the input document passed as an argument to a script and the transformation invoked on the server side.

8. References

[1] Dublin Core Metadata Initiative, DCMI, <http://dublincore.org/>

[2] John Kunze, Encoding Dublin Core Metadata in HTML, DCMI, RFC2731, Dec 1999, <http://www.ietf.org/rfc/rfc2731.txt>

[3] Andy Powell, Expressing Qualified Dublin Core in HTML/XHTML meta elements, Proposal to DC Architecture WG, Sep 2002, <http://www.ukoln.ac.uk/metadata/dcmi/dcq-html/>

[4] Rachel Heery & Manjula Patel, Application profiles: mixing and matching metadata schemas, *Ariadne*, 25 (Sep 2000) < http://www.ariadne.ac.uk/issue25/app-profiles/>

[5] Andy Powell, DC-dot: Dublin Core metadata editor, UKOLN, <http://www.ukoln.ac.uk/metadata/dcdot/>

[6] Dublin Core Extraction Service, W3C, <http://www.w3.org/2000/06/dc-extract/form.html>

[7] RDF Validation Service, W3C, <http://www.w3.org/RDF/Validator/>

[8] HTML Tidy Service, W3C, <http://cgi.w3.org/cgi-bin/tidy>

[9] dcmeta, UKOLN, <URL to be confirmed>