The DCMI Tools application profile

Thomas Severiens
Institut für wissenschaftliche
Information e.V.
Universität Osnabrück, Germany
severiens@mathematik.uni-osnabrueck.de

Jane Greenberg
School of Information and Library
Science, University of North Carolina
at Chapel Hill, USA
janeg@ils.unc.edu

Abstract

This paper reports on the DCMI Tools application profile for describing metadata applications. It defines the metadata tools landscape, presents the steps to develop the application profile, and presents the DCMI Tools application profile and taxonomy for tool functionalities. It concludes by highlighting the next steps related to the implementation of the application profile.

Keywords: application profile; metadata tools; metadata applications; interoperability; metadata generation.

1. Introduction

The community of people interested in metadata tools has expanded dramatically over the last decade as more and more people recognize the value of metadata standards for preserving, organizing, managing, and making their digital resources accessible via the Web. With this growth has been an increase in the range of applications being labeled as metadata tools. Examples include algorithms that plug in to various multi-functional software: metadata editors, templates, and generators, software applications (e.g., Microsoft's WORD, instructional course software, mp3 software) that include metadata functionalities, and social software (e.g., Flickr or Del.icio.us) supporting metadata generation, including tags. Add to this, software supporting the creation of taxonomies and ontologies which can be integrated into metadata applications. As a result of the diversity of available metadata applications, it can be difficult to determine which will best support a project's needs and services.

There is a need for a single portal providing one-stop shopping so information professionals and others needing to create metadata can learn about metadata tools, particularly open source applications. The Dublin Core Tools (DCMI Tools) Community recognizes this need and over the last year has been working on an application profile to update the DCMI Tools and Software Program (www.dublincore.org/tools/) which has been supporting this type of service for the last several years. The implementation of the application profile will separate content and presentation information, thus meeting Semantic Web requirements (Severiens & Thiemann, 2007).

2. The Application Profile Approach

2.1. Rationale Behind the Approach

According to Greenberg & Severiens (2007), "An application profile is a declaration of the metadata terms an organization, information resource, application, or user community uses in its metadata… An application profile is not considered complete without documentation that defines the policies and best practices appropriate to the application." Key reasons for developing an application profile are:

- It does not make sense to invest resources in developing a scheme, when others have already developed a scheme (or parts of a scheme) that support a project's need.

- It makes good sense to draw upon and unite the part (or parts) of a scheme (or schemes) that can support a project's needs, when an individual scheme cannot not fully satisfy a project's needs.
- Developing an application profile, consistent with other metadata schemes, will help to achieve greater interoperability on the World Wide Web, and support Semantic Web services.

2.2. Procedures for Developing the DCMI Tools Application Profile

The procedures in developing our application profile included namespace selection, element assessment and raking, element selection, and documentation. Several details about these steps follow below:

1. Namespace selection: We first selected the following three namespaces from which to draw elements: 1. Dublin Core (ISO 15836-2003) purl.org/dc/elements/1.1/, 2. DCTERMS (purl.org/dc/terms) and 3. DOAP (Description of A Project) (usefulinc.com/doap/). We selected these spaces because of the collection of known elements and the goals of the DC Tools Community.
2. Element assessment and ranking. We then reviewed each element in each of the namespaces, and applied an initial three level ranking of each element's usefulness to our goals. Level one indicated necessary, level two indicated potentially valuable, and level three indicated not germane.
3. Element selection: We then reviewed all the elements ranked with a one or two, and selected the elements for the DCMI Tools application file. All level one ranked elements, and slightly over half of the level two ranked elements were selected.
4. Documentation: We have produced an application profile document (draft), a taxonomy of tool functionalities for classifying the metadata tools, and a glossary to aid users in understanding the taxonomy and basic concepts related to metadata tools.

3. DCMI Tools Application Profile

The DCMI Tools application profile contains 17 elements, drawing from the Dublin Core, the DCTerms, and DOAP schemes. Nine of these elements contain qualifiers. Table 1 presents an overview of our application profile, including examples produced with DC-dot.

The application profile can be implemented within a Semantic Web framework, with automatic processes and requires the use of controlled vocabulary wherever possible. This will improve the representation of objects described, allowing for fairly complete metadata descriptions. The most important part of the application profile is the classification of objects by genre. We have also developed a taxonomy of tool functionalities, listed in Table 2.

We will extend these classes as new types of software are developed. Classes not filled with latest software will be deleted, and the list will be revised as needed to allow for appropriate growth. We see this list as being organic, in order to meet the needs of the tools community over time.

Several open questions remain as part of our work in developing the profile. For example, location information requires some additional attention. The most useful and precise would be to give geographical coordinates so a service could link to map serves etc.. An alternative would be to use a list of location vocabulary handling all the problems of local and international naming of locations.

For the agent roles in the application profile we tried to use the roles defined in DOAP namespace (usefulinc.com/doap/) mostly reused from the foam-project results: developer, documenter, maintainer, tester, translator.

TABLE 1. DCMI Tools application profile.

Name-space	Element	Qualifiers/Terms	Example DC-dot
dc	Contributor	doap:maintainer doap:developer doap:documenter doap:translator doap:tester	translator:Rachel Heery
dc	Creator		Andy Powell
dc	Date	dcterms:created dcterms:dateCopyrighted dcterms:modified dcterms:issued	created: 7 July 1997
dc	Description		Extracts and validates metadata from HTML resources and MS Office files. The generated metadata can be edited using the form provided and converted to various other formats (USMARC, SOIF, IAFA/ROADS, TEI headers, GILS, IMS or RDF) if required.
dc	Identifer	doap:repository	www.ukoln.ac.uk/metadata/dcdot/
dc	Language		En-us, en-GB
dc	Publisher		
dc	Relation	dcterms:hasPart dcterms:hasVersion dcterms:isPartOf dcterms:isReplacedBy dcterms:isRequiredBy dcterms:isVersionOf dcterms:replaces dcterms:requires doap: release	requires: Libwww-perl, soif.pl, Jon Knight's MARC module
dc	Rights	dcterms:accessRights dcterms:license	accessRights:open source license: www.gnu.org/copyleft/gpl.html
dcterms	rightsHolder		
dc	Source	dcterms:URI	
dc	Title	dcterms:alternative	DC-dot
dc	Type	dcterms:dataset dcterms:InteractiveResource dcterms:service dcterms:software	dcterms:InteractiveResource
dcterms	Audience	dctools:developer dctools:users	
doap	location		Bath, UK
doap	programming-language		Perl
doap	Operating-system		

TABLE 2. Taxonomy of tools functionalities.

Conversion
Crosswalk
Metadata Creation
Metadata Encoding
Metadata Extraction
Metadata Generation
Metadata Harvesting
Metadata Templates
Search Engines
Translation
Transliteration
Validation

To re-use the collected information in multiple frameworks, it will be requested to clearly define all vocabulary used. For use in a Semantic Web framework this will be offered as RDFS. For human readability there is restriction to textual representation in this article. To assist with our work and further bring the metadata tool user community together, we have also developed a Glossary (www.dublincore.org/groups/tools/glossary.html). This is a new development and will be enhanced and modified over time.

The database from which www.dublincore.org/tools is being generated contains the following structure:

- Title: corresponding to the dc.title field in the app. profile.
- URL: corresponding to the dc.identifier field in the app. profile.
- Description: corresponding to the dc.description field in the app. profile.
- Classification: sorting the service into the different classes.
- Free/commercial: corresponding to the dc.rights qualifier dcterms:accessRights
- Online/download/webservice: corresponding to the dc.type field in the app. Profile and its qualifiers dcterms.InteractiveResource / dcterms.software / dcterms.service, a tag for dcterms.dataset may be added if an entry is being included into the database.
- Country: corresponding to the field doap.location.
- Comment: allowing some free text comments.
- Provider: corresponding to the dc.publisher field in the app. Profile.

With the application profile, the following fields will be added to the database:

- Information on the contributors who can be developers, documenters, maintainers, testers, translators.
- Information on the creator(s).
- Information on the dates associated with the object, e.g.: the date of its creation, date of its latest modification, date it was issued, or the date of its copyright notice.
- Information on the language of the object.
- Information on the relations of the object to other objects in the database.

- Information on the license e.g. a link to the license text, information on the license. holder, while the date of the license was already given with the dates above.
- Information on the source, if they differ from the compiled resource.
- Information on the used programming language, if a source is available.
- Information on the operating systems requested for running the software, if its not an webservice or online service.

4. Conclusions and Future Work

Creating the DCMI Tools application profile has involved team work, and the experience has been a thoughtful and fruitful process, for example, we discussed a first draft of the application profile with the audience at ELPub conference in Vienna in June 2007 (info.tuwien.ac.at/elpub2007/presentations/123.pdf). Our next steps include testing the application profile and revising the DC Tools and Software program Website listing tools. Our implementation will allow us to evaluate the overall effectiveness of the DCMI Tools application profile and identify areas requiring attention and revisions. We then plan to take steps toward developing a collection and maintenance policy for the DCMI Tools and Software program.

References

Greenberg, Jane, and Thomas Severiens. (2007). DCMI-Tools: Ontologies for digital application description. *Proceedings of ELPUB Conference.* Retrieved July 2, 2007, from available from http://elpub.scix.net/data/works/att/123_elpub2007.content.pdf .

Severiens, Thomas, and Christian Thiemann. (2007). RDF database for PhysNet and similar portals. *Cataloging & Classification Quarterly*, *43*(3/4), 127-147.