# Maintaining a vocabulary: Practices, policies, and models around Dublin Core

Thomas Baker
Birlinghoven Library
Fraunhofer-Gesellschaft
Bonn, Germany

## Abstract

The Dublin Core Metadata Initiative (DCMI) maintains a vocabulary of several dozen metadata terms, most notably the fifteen-element Dublin Core. Within DCMI, however, discussion has focused more on the practices, policies, and models around the Core. Its initial single-resource data model is evolving towards a fuller framework for evaluating the diversity of Dublin Core implementations. Terms (and their historical versions) are identified with URIs, documented on Web pages and in formal schemas, indexed in registries, and cited in application profiles. With low-overhead maintenance processes, DCMI seeks to balance the growth of a central standard against recognition and re-use of other, complementary vocabularies.

### Keywords

Metadata, Dublin Core, Semantic Web, Uniform Resource Identifiers, controlled vocabularies, application profiles, metadata registries

## 1 A DCMI model of practice

Over the past nine years, the fifteen elements of Dublin Core — Title, Description, Date, and the rest — have become part of the digital landscape. They have been printed on tee shirts, used in millions of metadata records, and enshrined in an international standard, ISO 15836 [6]. Its user community, the Dublin Core Metadata Initiative (DCMI), has evolved into a maintenance network [7].

Within DCMI, however, discussion has focused less on the Core itself than on a broader set of contextual issues around its edges. The initial problem of 1994 — to agree on some descriptors for embedding in Web pages — has evolved into a cluster of problems regarding technical architecture, naming policy, and administrative process for maintaining a standard. Much of this discussion has aimed at clarifying how the elements can be used with a broad and increasingly sophisticated array of Web technologies, from the simple HTML of the early years through XML and XML Schemas to the Resource Description Framework and, looking ahead, to ontology languages still under development. Much of this effort has been motivated by the vague but powerful vision of a Semantic Web in which well-defined data and metadata underpin applications that are increasingly automated and intelligent.

This paper describes the principles and practices that provide the context for Dublin Core — "everything but" the fifteen elements themselves. Specifically, it examines four aspects of Dublin Core as a Semantic Web vocabulary:

- its grammatical principles and abstract model,
- policies for identifying metadata terms,
- the documentation of metadata terms, and
- open processes of maintenance and development.

Taken together, these features provide a model for declaring and maintaining a metadata vocabulary in a general sense, and DCMI's experience may hold lessons for maintainers of vocabularies quite different in scope and purpose.

## 2 Metadata principles

### 2.1 Towards a data model

The initial idea for Dublin Core was of a small set of elements so generic and well-understood that they would cover the most basic requirements for simple description. The analogy was that of a metadata "pidgin" — a phrase-book vocabulary good enough for helping "digital tourists" find resources in an intellectually and culturally diverse Internet. The initial metaphor was that of a library catalog card, and the drafters of Dublin Core asked themselves which elements a user would want to see in a search form.

Early in the process, however, it was recognized that new requirements for machine-processability on the Web would require a coherent data model. The question shifted from "what do we see on the screen?" to "how will machines make sense of this?". The notion of an architectural context for associating multiple alternative or complementary types

of metadata with a single resource — the "Warwick Framework" — was introduced at an early workshop in 1996, providing a conceptual basis both for limiting the scope of Dublin Core itself (to "resource discovery") and for using the Dublin Core alongside other vocabularies to address requirements beyond the scope of any one type of metadata [22]. In 1997, when the World Wide Web Consortium (W3C) set about designing a Resource Description Framework (RDF), part of the working group's motivation was to provide a formal mechanism "for defining 'schemas' for descriptive vocabularies like the Dublin Core" and to support Dublin Core requirements such as modularization and qualification [25].

The elaboration of a formal model for RDF in W3C had the important effect of reinforcing a simple though limited data model for Dublin Core — sometimes called the Hedgehog Model — of a single entity (the "resource") bristling with attributes ("elements"). Each element, in turn, was associated with a string value, sometimes called an "appropriate literal". In 2000, the notion of "qualifying" Elements with Element Refinements and Encoding Schemes with optional and additional information about the scope or values of an element was introduced in response to a user need to customize the Dublin Core for particular uses or applications. [10]. Though couched in DCMI-specific terminology, the Dublin Core model was designed to be fundamentally consistent with RDF [2]:

- Elements were what RDF calls "properties".
- Element Refinements (one type of qualifier) were in RDF terms properties that were semantically narrower than other properties. For example, the DCMI element refinement "Date Created" was characterized in RDF terms as a subproperty of the element "Date".
- Encoding Schemes (the other type of qualifier) provided a context for interpreting an element value and were modeled as RDF "classes". For example, the encoding scheme "DDC" was used to indicate that a metadata value came from Dewey Decimal Classification.

It is perhaps inevitable that metadata, like any other form of human language, be caught between a need for simplicity and an urge to complexify. The challenge of allowing specificity within the limits of a generic vocabulary — and without breaking or extending the Hedgehog Model of a single resource with attributes — was met with a principle known as "Dumb-Down". According to the Dumb-Down Principle, the qualified description of a resource should be simplifiable, albeit lossily, into a more generic unqualified form. For example, a "Date Created" should be interpretable (albeit with less specificity) as a generic "Date".

## 2.2 Lesser and fuller models

Since the early years, users have chafed at the Hedgehog Model and its limited forms of qualification. The model itself has not provided a way to describe several different Hedgehogs within one framework — for example, to describe not just the properties of an information resource (e.g., Creator, Title, and Subject) but also the properties of its Creator (e.g., Name, Affiliation, and Birthdate). There are of course numerous ways to transcend this limitation within the context of particular applications, but the challenge has always been to solve the problem in a general way that will make sense to applications far removed from the original context, where the machines may have no way to understand or infer any ad-hoc extensions to the Hedgehog Model that may have been used as a basis for the metadata.

One such ad-hoc approach, for example, has been to shoehorn the attributes of an author — Name, Affiliation, Fax Number — into a single string value for the element Creator. For applications that know how to parse such a string to derive its multiple components, such a "structured value" solves a concrete problem without too much fuss. For applications that harvest such metadata along with metadata from dozens of other providers, however, the extraneous information embedded in such strings has the effect of polluting the common index.

Another approach has been to articulate conventions for using the capability of more sophisticated data models, such as RDF, to associate a string value (a "label") with an abstract entity that itself may have an arbitrary set of attributes (a "resource"). According to one method for expressing Dublin Core in RDF, for example, a Creator may be both modelled as an entity with its own attributes such as Birthdate and Affiliation and, in parallel, represented by the creator's name as a literal string value [21].

The known limitations of the Hedgehog Model are currently being addressed with a proposal from Andy Powell for an extended Abstract Model for Dublin Core metadata [23]. This proposal acknowledges that in practice, metadata creators work within the limits and assumptions of a broad range of models and technologies. Through articulating a full set of modelling features in common use — for example, the distinction between values seen as resources and values represented by literal strings — the model is designed to provide a common point of reference against which the features and limits of specific technologies and encoding methods may be compared.

As currently proposed, the Abstract Model can be implemented most fully using RDF. As has long been recognized, however, flat sets of HTML META tags lack constructs for distinguishing multiple entities (e.g., a resource and its creator), so HTML implementations cannot support specific features of the Abstract Model. Other encoding technologies, such as XML, lie between these two extremes.

The Abstract Model also provides a basis for

proposing guidelines of good practice for the use of Structured Values. The empirical analysis of metadata records suggests a typology for Structured Values that includes Unlabelled Strings (e.g., a simple date format), Markup (e.g., a text tagged with HTML or TeX), and Labelled Strings (e.g., a vCard with multiple tagged components). Beyond these three types are Structured Values which constitute, in effect, Related Resource Descriptions — sets of attributes related not to the resource described but to entities associated with attributes of the resource described. The Abstract Model reaffirms that string values must be appropriate to a given element while providing for Related Resource Descriptions as related but logically separate entities.

The range of methods and models used for Dublin Core is but a reflection of the diversity in methods and models used for metadata generally. As interoperability presupposes a shared basis for comparison or translation, the Abstract Model and the analysis on which it is based could be of use in addressing interoperability between different metadata communities. Representatives of the IEEE LOM, a metadata standard comparable in scope but quite different from Dublin Core in terms of modeling, have helped formulate the Abstract Model. Potentially, the model could form a basis for understanding interoperability between Dublin Core and other metadata standards as well.

# 3 Identifying metadata terms

## 3.1 DCMI Namespace Policy

The Internet was revolutionary because it made the resources of any connected server accessible via a single global address space. The vision of a future Semantic Web further generalizes this notion of a global space of addresses to that of a global space of identifiers. According to Tim Berners-Lee, "The most fundamental specification of Web architecture, while one of the simpler, is that of the Uniform Resource Identifier, or URI. The principle that anything, absolutely anything 'on the Web' should be identified distinctly by an otherwise opaque string of characters... is core" [4].

URIs can provide unique identity not just to "information resources" — Web pages, scientific preprints, satellite photos, video clips, and the like — but also to any metadata terms used to describe those resources. As compact character strings associated with known institutional domain authorities, URIs can stand alone as self-contained references to metadata terms. While relevant to all data technologies, they are usable most directly in Web-based description technologies such as XLink, Topic Maps, and RDF.

DCMI began to experiment with URIs in 1997, which led to the formulation in 2001 of a formal Namespace Policy [13]. This policy declares URIs for three DCMI namespaces:

- http://purl.org/dc/elements/1.1/
- http://purl.org/dc/terms/
- http://purl.org/dc/dcmitype/

to designate (respectively) the fifteen-element Dublin Core, all other DCMI elements and qualifiers, and a controlled vocabulary of values for the Dublin Core element Type. A URI is constructed for a DCMI term by appending its character-string "name" to the URI of a DCMI namespace. For example, the URIs

- http://purl.org/dc/elements/1.1/title
- http://purl.org/dc/terms/extent
- http://purl.org/dc/dcmitype/Image

respectively identify Title (one of the fifteen "core" elements), Extent (an element refinement) and Image (a term in the DCMI Type Vocabulary).

Aside from the namespace for core elements, which predated the Namespace Policy, the URIs for DCMI namespaces do not show versioning information. Rather, acknowledging that terms can and will change over time, the policy focuses on articulating the consequences of change for unique identity. "Minor" or "substantive" errata may be corrected without consequence for URIs. Changes of a semantic nature, however, such as significant changes in the wording of a definition, must trigger the creation of a new term with a new URI. To support the future interpretation of legacy metadata, the Namespace Policy commits DCMI to maintaining formal documentation for all assigned URIs — even for terms that might some day be assigned a status of "obsolete".

The ability to reference metadata terms unambiguously is a precondition for developing semantic maps and interoperability services across multiple standards. In November 2002, DCMI participated in the "CORES Resolution", an agreement among the maintainers of eight major semantic standards — Dublin Core, MARC21, UNIMARC, GILS, ONIX, CERIF, DOI, and IEEE/LOM — to identify their metadata elements with URIs. The agreement characterized the "elements" of the various standards as "units of meaning comparable and mappable to elements of other standards". The agreement also called for maintenance organizations to articulate explicit policies regarding the stability, persistence, and maintenance of their URIs [3].

## 3.2 Identifying versions

Within the limits of the Namespace Policy, the DCMI vocabularies are subject to growth and change over time — new terms are added, a bibliographic reference cited in a usage comment may be updated, the status assigned to a term may change. The fifteen-element Dublin Core was initially versioned as a set and, as noted above, the version number "1.1" is hard-coded into the string used as the URI of its DCMI namespace.

As of July 2000, new terms were issued without such a version number because the model of periodic, batched releases seemed a bad fit to a vocabulary that was expected to grow by increment. At the same time, the ability to reference a term set as of a given date was seen as potentially useful for library automation contracts, translations of DCMI term sets into another languages, or the future interpretation of legacy metadata. The pragmatic solution to this problem has been to version both individual terms (which evolve at different rates) and Web pages which document batches of terms as of a particular date (which are updated whenever a term is added or anything else in the term set changes).

Individual terms are versioned by saving a snapshot of their attributes whenever any one of their attributes changes and assigning to that snapshot a URI such as the following:

- http://dublincore.org/usage/terms/history/#Image-002
- http://dublincore.org/usage/terms/history/#Image-001

Although such URIs are not currently supported by the DCMI Namespace Policy, they effectively function as identifiers for successive versions of a term (in this case Image). At present, these URIs resolve to anchors in a Web document which holds a periodically updated snapshot of all past and present versions of all DCMI terms.

The Web pages documenting DCMI term sets are versioned according to the method used for all other important DCMI documents — with a URI for the specific historical version and a URI for its "latest version" along with pointers to immediate previous and successive versions. For example, the March 2003 version of the DCMI Metadata Terms document shows the following:

- Identifier: http://dublincore.org/documents/2003/03/04/dcmi-terms/
- Latest Version: http://dublincore.org/documents/dcmi-terms/
- Replaced By: http://dublincore.org/documents/2003/11/19/dcmi-terms/

where the Identifier resolves to the permanently archived and unchanging version of the document displayed, Replaced By resolves to the next version that followed, and Latest Version resolves to a continually updated pointer on the DCMI Web site to the most up-to-date version of DCMI Metadata Terms. In DCMI practice, in other words, the versioning methods for metadata terms and for Web documents are analogous: in each case, identifiers are assigned both for the resource in a generic sense (the namespace-policy-supported URI for a term or Latest Version for a document) as well as for a specific historical version.

### 3.3 Registering controlled vocabularies

The set of qualifiers approved in July 2000 included eleven Vocabulary Encoding Schemes. Vocabulary Encoding Schemes are used in metadata records to indicate that the value of a Dublin Core element, usually Subject, is taken from a controlled vocabulary such as Library of Congress Subject Headings, Dewey Decimal Classification, or the Getty Thesaurus of Historical Names. Metadata creators have long requested that DCMI provide a streamlined process for the "registration" of several dozen other vocabularies in active use. In response, DCMI developed a Web tool and fast-track process to efficiently handle a proposal from its submission through to the creation of a unique DCMI-maintained name (and URI) for use in metadata records.

Having conducted a trial run of the fast-track process in 2003, DCMI is currently preparing for a second run, though questions remain. Should the persistence policy for DCMI namespaces be extended without modification to vocabulary encoding schemes registered by a process of higher volume and lower control? How much effort will it take to process proposals and maintain legacy data on a production basis? Moreover, the controlled vocabularies themselves will be subject in an uncontrolled variety of ways to changes, versioning, even obsolescence. How much work will be required to maintain descriptions and Web links? Issues regarding the construction of unique names and URIs that have been solved for the small set of DCMI terms could reappear in the context of a much bigger namespace — for example, with respect to clashing acronyms, language translations or other derivative works, and generic "works" (such as "Dewey Decimal Classification") as opposed to specific versions ("DDC 16th Edition of 1958").

In the meantime, it is becoming increasingly likely that URIs usable as Vocabulary Encoding Schemes in Dublin Core metadata will be declared by some maintenance agencies to designate their own vocabularies. The US Library of Congress, for example, has begun to do this. In order to fully implement the registration of encoding schemes, DCMI will need to put into place a "good neighbor policy" (for cross-referencing DCMI-maintained URIs with newly coined non-DCMI URIs), channels for publicizing such cross-references (i.e., Web pages and registry databases), and mechanisms for giving these references formal expression (e.g., RDF schemas). In some cases, then, DCMI registration of encoding schemes will in effect serve as a stop-gap measure to support referencing of a controlled vocabulary ahead of the subsequent assignment of a URI by its own maintainer.

One more general solution to this problem is suggested by the recent formulation of an "info" URI scheme [1]. The "info" URI scheme is designed to provide a space of global identifiers within which Namespace Authorities can coin URIs to identify their own resources without the expectation that a

URI resolve to a resource or service on the Web. These URIs could be used for a broad range of resources including controlled vocabularies and their member terms. As currently planned, the "info" URI scheme will be managed by a public registry with a lightweight registration process [19]. Analogously to considerations around the registration of encoding schemes by DCMI, it is acknowledged that the "info" Registry could provide "a lightweight early URI registration mechanism to support references of public information assets ahead of any possible subsequent URI scheme or URN namespace application" [1].

# 4 Documenting terms

## 4.1 Schemas and Web pages

The DCMI Namespace Policy specifies that all of the URIs identifying DCMI terms "will resolve to a machine-processable DCMI term declaration for all the terms within that namespace" [13]. As of July 2003, the URIs for DCMI terms redirect to an RDF schema documenting the attributes of a particular set of terms [14]. (The formulation is vague enough to support any alternative conventions for namespace resolution that may emerge, such as the Resource Directory Description Language under consideration by the W3C Technical Architecture Group [24]). This RDF schema expresses the semantic relationships between terms in a form processable by machines — for example, by declaring that the term "Date Created" is semantically narrower than (i.e., a "subPropertyOf") the broader term "Date". The technical specifics of the schema model is periodically reviewed for conformance to W3C specifications and evolving practice in the wider Web and ontology communities.

The publication of DCMI term declarations as RDF schemas does not replace the need to publish term declarations as ordinary Web pages as well. Keeping the Web pages and schemas aligned with each other poses a significant problem of workflow. For such small term sets, it has proved efficient to edit a master data file in XML and use XSLT scripts periodically to generate fresh versions of both the schemas and Web pages. It is worth noting that DCMI has not yet seen more sophisticated vocabulary management tools that would manage this process in a significantly more practical or user-friendly manner, suggesting that the challenge of documenting evolving term sets larger than Dublin Core in multiple formats is one that will require further attention.

## 4.2 Application profiles

Application designers often draw on standards such as Dublin Core rather freely, selecting a a subset of available terms, specifying constraints on how terms are used (e.g., which elements are mandatory, which date format to use), or using terms from multiple sources — e.g., Dublin Core elements together with elements from the IEEE/LOM standard or with elements defined in-house for application-specific purposes. In order to document such practices, standards communities as different as DCMI, DOI, MARC, and IEEE/LOM have developed roughly analogous notions of a "profile" or "application profile".

In the DCMI context, the notion of Application Profile emerged in response to a strong though vaguely expressed need for documenting and sharing metadata models within communities of interest. Application Profiles have been developed for purposes ranging from the description of existing record structures to the negotiation of good-practice guidelines for formats yet to be implemented. Application Profiles have helped identify emerging semantics "around the edges" of existing vocabularies as candidates for formal standardization.

A Workshop on Dublin Core Metadata in the European Committee for Standardization (CEN) has put forward guidelines for a consistent documentary format for Dublin Core Application Profiles (DCAPs) [5, 28]. These guidelines suggest workaround methods for documenting the use of non-DCMI terms that are not yet formally identified with URIs. Also addressed is the use of DCAPs for describing metadata records based, implicitly or not, on underlying structures that are more complex — or simply less coherent — than DCMI's Hedgehog Model. In this respect, drafting a DCAP may be seen as an intellectual effort — that of mapping a given metadata structure to a target format — which may not be achievable by automatic methods alone.

Building on this work, the CEN workshop is developing a formal model for DCAPs that are machine-processable. This further step is necessary if Application Profiles are to have a practical use for automating processes such as metadata conversion or normalization or if they are to be merged into "registry" databases (discussed below). In any case, it seems likely that work in this area will progress through a mixture of specification development and pragmatic experimentation. Ongoing work on DCMI's Abstract Model could eventually provide this work with a modelling basis broader than the Hedgehog Model alone.

## 4.3 Metadata registries

Another emerging channel for publishing information about metadata vocabularies and application profiles is DCMI's "registry" — a Web-based index of multiple vocabularies which serves up information about metadata terms in response to queries [9]. The DCMI registry is part of a broader effort to build a cluster of compatible and complementary registries both of general scope and on specific topics such as learning materials and economic development [8, 17].

Aside from supporting DCMI grammatical principles and the Hedgehog Model, many of these registry efforts share an orientation to RDF as a technology which, in principle, could help users better to grasp

the interdependencies between different vocabularies, design semantic crosswalks and indexing strategies, harmonize metadata practice across a body of applications, and discover emerging semantics from empirical usage patterns. Many are based on the notion that term declarations, application profiles, and controlled vocabularies would be available as RDF schemas for harvesting over the open Web directly from their maintainers.

Proponents of this distributed model face a dilemma inasmuch the proof of concept for the model requires pushing the provision of vocabularies out to their maintainers. However, the lack of well-established conventions for metadata vocabularies has hampered the development of user-friendly tools. The lack of user-friendly tools, in turn, has made it unreasonably difficult for providers to maintain their own vocabularies in formal structures such as RDF schemas. A premature proliferation of schema models, on the other hand, could create a legacy corpus and act as a drag on progress. Ultimately, the success of the distributed model will depend on the extent to which stable and well-understood conventions for a typology of schemas — "term declarations" as opposed to "translations", "controlled vocabularies", or "application profiles" — emerge and are followed in the wider Web community.

# 5 Open processes

## 5.1 DCMI Usage Board

In its evolution from a workshop series into a maintenance agency, the Dublin Core Metadata Initiative has developed formal processes of editorial control. Since 2001, proposals for extensions or clarifications to the standard are evaluated for conformance to grammatical principle and usefulness by a nine-member Usage Board [15]. Each decision of the Usage Board is assigned a URI, and links are created to supporting documentation, decision texts, and to the historical term declarations of any metadata terms affected by the decisions [16]. The mailing list used by the Usage Board, along with all relevant meeting materials, are archived on the open Web.

The emerging process model has tried to balance a need for democratic participation and collective review — the notion that "all of us are smarter than any of us" — against the reality that technical work is often done most efficiently by small teams of dedicated authors. Most proposals of a semantic nature — e.g., proposals for new terms or clarifications of usage for existing terms — are formed in open working groups and posted to the general DCMI mailing list for a one-month comment period before coming before the Usage Board for approval. Approval by the Usage Board requires near-consensus, with just one dissenting vote allowed [18]. Proposals of a more technical nature, such as guidelines for encoding Dublin Core metadata in XML, are typically drafted by one or two authors, discussed and iteratively improved on open mailing lists, and given a final careful review by an ad-hoc set of reviewers appointed by the DCMI Directorate.

## 5.2 Coining new terms versus re-use

Over the past two years, the Usage Board has shown a bias towards keeping the vocabularies maintained by DCMI relatively small and generic. This bias has been motivated in part by a desire to avoid the slippery slope of complexification. The danger is that adding ever more-specialized terms to meet the perceived needs of particular communities could sacrifice the generic simplicity that constitutes much of Dublin Core's appeal. Keeping the overhead of review and approval light also helps ensure that DCMI's vocabularies can continue to be maintained by networked committees of busy experts on a voluntary basis.

DCMI has at times considered serving as a "namespace host" — the home for a maintained namespace with known URI policies where metadata terms coined by ad-hoc user groups might be published without any sort of review or implied approval on the part of DCMI. In principle, this could enable communities of practice to use or reference a new metadata term as soon as they were created. It has likewise been proposed that terms put forward for review be assigned a DCMI name or URI at the time of proposal, then simply be moved or upgraded in status when officially approved.

Variants of these proposals have their appeal, but they would have the net effect of making it easy for the term space to expand quickly while leaving unclear how the terms should be maintained over the longer term. If DCMI were to undertake the creation of cross-references to semantically overlapping terms elsewhere in the DCMI namespaces or hosted namespaces, this task could rapidly become a burden. If that task were considered a responsibility of DCMI, the terms would be perceived as having DCMI approval. Either way, the implied commitment and role of DCMI over the longer term is not well understood.

In order to keep its own vocabularies small, the Usage Board has put effort into clarifying how DCMI-maintained vocabularies can be used in conjunction with more detailed or domain-specific vocabularies declared and maintained by specialised communities of expertise outside of DCMI. This has involved discussion with maintainers of other vocabularies and standards on forms of mutual recognition and support. For example, DCMI is currently working with the Library of Congress on the mechanics of formally declaring MARC relator terms (such as Adaptor, Artist, and Translator) to be refinements of the Dublin Core element Contributor. The current plan calls for Library of Congress to post an RDF schema declaring this sub-property relationship for each MARC Relator term, and for DCMI to find appropriate ways to point Dublin Core users to the MARC Relator list. As discussed above, the creation by Library of Congress of a URI for Library of Congress Subject Headings suitable as an alternative

to the DCMI Encoding Scheme "LCSH" likewise implies the declaration and maintenance by DCMI of a reference from the latter to the former.

### 5.3 General lessons

DCMI has evolved a set of principles and practices for declaring and maintaining metadata vocabularies in the historically new and rapidly evolving context of a Semantic Web. Its typology of terms, Hedgehog Model, and more recently the draft Abstract Model provide a conceptual foundation for metadata practice of the sort that a grammar provides for any other sort of language. Models for term declarations and application profiles, published both as Web pages and as formal schemas, address emerging requirements for machine-processability. Peer review within a global network provides a low-overhead model of process.

DCMI's choices reflect a tension between building up a central standard versus pushing maintenance responsibility out to more specialized communities. Distributing responsibility implies not just shared technical models or grammars, but shared conventions that are essentially social in nature, such as the etiquette of mutual recognition between DCMI and maintainers of complementary vocabularies. Whether or not DCMI has yet found the ideal solution, any future Semantic Web will need comparable conventions and policies for declaring and maintaining its underlying semantics. Where the specifics of DCMI practice are not followed, models and conventions much like them will need to be invented. Solutions may differ, but many of the problems will surely be recognizably the same.

## References


[1] "About 'info' URIs — Frequently Asked Questions", http://www2.elsevier.co.uk/~tony/info/info.html.

[2] Thomas Baker, "A Grammar for Dublin Core", D-Lib Magazine (October 2000), http://www.dlib.org/dlib/october00/baker/10baker.html.

[3] Thomas Baker, Makx Dekkers, "Identifying Metadata Elements with URIs: The CORES Resolution", D-Lib Magazine (July 2003), http://www.dlib.org/dlib/july03/baker/07baker.html.

[4] Tim Berners-Lee, "Web architecture from 50,000 feet", http://www.w3.org/DesignIssues/Architecture.html.

[5] "Dublin Core Application Profile Guidelines" [CEN Workshop Agreement CWA 14855], European Committee for Standardization, http://www.cenorm.be/cenorm/businessdomains/businessdomains/informationsocietystandardizationsystem/published+cwas/cwa14855.pdf.

[6] "The Dublin Core Metadata Element Set" [ISO 15836:2003(E)], http://www.niso.org/international/SC4/n515.pdf.

[7] Dublin Core Metadata Initiative, http://dublincore.org/.

[8] DCMI Registry Working Group, http://dublincore.org/groups/registry/.

[9] Dublin Core Metadata Registry, http://dublincore.org/dcregistry/.

[10] "Dublin Core Qualifiers", http://dublincore.org/documents/2000/07/11/dcmes-qualifiers/.

[11] "DCMI Grammatical Principles", http://dublincore.org/usage/documents/principles/.

[12] "DCMI Metadata Terms", http://dublincore.org/documents/dcmi-terms/.

[13] "DCMI Namespace Policy", http://dublincore.org/documents/dcmi-namespace/.

[14] "DCMI Term Declarations Represented in RDF Schema Language", http://dublincore.org/ schemas/rdfs/.

[15] DCMI Usage Board, http://dublincore.org/usage/.

[16] "DCMI Usage Board decisions", http://dublincore.org/usage/decisions.

[17] DELOS Working Group on Registries, "Principles of metadata registries", http://delos-noe.iei.pi.cnr.it/activities/standardizationforum/Registries.pdf.

[18] Diane Hillmann, Stuart Sutton, "DCMI Usage Board Administrative Processes", http://dublincore.org/usage/documents/process/.

[19] "info" Registry, http://info-uri.info/registry/.

[20] Traugott Koch, "Guidelines for Vocabulary and Encoding Scheme Qualifiers", http://dublincore.org/usage/documents/vocabulary-guidelines/.

[21] Stefan Kokkelink and Roland Schwänzl, "Expressing Qualified Dublin Core in RDF/XML", http://dublincore.org/documents/2002/05/15/dcq-rdf-xml/.

[22] Carl Lagoze, "The Warwick Framework: a container architecture for diverse sets of metadata", http://www.dlib.org/dlib/july96/lagoze/07lagoze.html.

[23] Andy Powell, Mikael Nilsson, Ambjörn Naeve, Pete Johnston, "DCMI Abstract Model" [DCMI Working Draft], http://www.ukoln.ac.uk/metadata/dcmi/abstract-model/2004-01-16/.

[24] "Resource Directory Description Language (RDDL)", http://www.tbray.org/tag/rddl4.html.

[25] "Resource Description Framework (RDF) Schemas" [W3C Working Draft 9 April 1998], http://www.w3.org/TR/1998/WD-rdf-schema-19980409/.

[26] Shigeo Sugimoto, Thomas Baker, Mitsuharu Nagamori, Tetsuo Sakaguchi, Koichi Tabata, "Versioning the Dublin Core across Multiple Languages and Over Time", *Proceedings of SAINT2001 Workshops*, San Diego: IEEE Computer Society, pp. 151–156.

[27] "Vocabulary Scheme Registration", http://wip.dublincore.org/schemes/index.html.

[28] Workshop on Dublin Core metadata, European Committee for Standardization (CEN), http://www.cenorm.be/sh/mmi-dc/.