

# Expressing Dublin Core in Topic Maps

Steve Pepper  
Ontopedia

Oslo, Norway  
pepper.steve@gmail.com

**Abstract.** The Dublin Core Metadata Initiative is an open organization engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models. Its most important standard is the Dublin Core Metadata Element Set [1], a vocabulary of fifteen properties for use in resource description, which was approved as ISO 15836:2003 [2]. This and other vocabularies developed by Dublin Core are defined as abstract models which may be expressed in any number of different syntaxes. This paper presents a proposal for expressing such metadata using Topic Maps.

**Keywords:** metadata, Dublin Core.

## 1 Introduction

The Dublin Core Metadata Initiative (DCMI) is an “open organization engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models.” It has developed a number of metadata specifications, the most important and widely used of which is the *Dublin Core Metadata Element Set* [1], which was approved as an ISO standard in 2003 [2].

DCMES and three other vocabularies are documented in [3]. DCMES itself consists of 15 elements: contributor, coverage, creator, date, description, format, identifier, language, publisher, relation, rights, source, subject, title, and type. These are supplemented by 40 additional *Elements and element refinements*. There is also a vocabulary of 18 *Encoding schemes*, one of which (the *DCMI Type Vocabulary*) is also a Dublin Core (DC) vocabulary consisting of 12 terms.

In the DCMI abstract model [4], “resources” are “described” by metadata using “property-value” pairs. ‘Resource’ is defined as “anything that might be identified” and, although the concept of “being identified” is not defined, it is clear from the examples that ‘resource’ is essentially the same as ‘subject’ in Topic Maps. The resources to which Dublin Core metadata is assigned are almost always *information resources*, but this need not be the case; they might, for example, be “human beings, corporations, concepts [or] bound books in a library.”

In terms of the abstract model, the first two vocabularies mentioned above define *properties*; the encoding schemes identify controlled vocabularies such as ISO 3166, TGN and DCMI Box, that provide *values* for properties such as **language**, **coverage**, etc.; and the DCMI Type Vocabulary defines a set of possible *values* (such as Text,

## 2 Steve Pepper

Ontopedia

Service, Image, etc.) for the property **type**. Thus, two vocabularies define properties, one defines schemes that are the source of values, and the fourth consists of actual values.

DC vocabularies are defined as abstract structures; no particular syntactic representation is mandated. Instead, metadata can be expressed in any syntax that is convenient to the user. Recommendations already exist for expressing Dublin Core in HTML [5], XML [6], and HTML [7], and a new work item was recently approved by ISO for a Type 3 Technical Report on *Expressing Dublin Core Metadata Using Topic Maps* [8].

There are a number of reasons why it would be useful to be able to express Dublin Core in Topic Maps: (1) in order to add metadata to individual topic maps; (2) in order to use Topic Maps to collate DC metadata; and (3) in order to exploit the merging capabilities of Topic Maps to combine DC metadata with other non-DC data. The advantages derived from each of these applications increases greatly if the same conventions are used in every case.

This paper presents an overview of previous work in this area, discusses the issues raised, and proposes a unified approach suitable for standardization. Feedback from the community is welcomed.<sup>1</sup>

## 2 Review of previous work

A number of proposals have been put forward over the years regarding how to use DC with Topic Maps. This section summarizes and compares them.

Algermissen 2001 [9] was the first published (but unfinished) attempt to use Dublin Core in a Topic Maps context. It is described as “a processing model for HTML <meta> tags that make use of the Dublin Core element set” and consists of a set of PSIs in the form of a topic map modeled on `psi1.xtm`<sup>2</sup> within the framework of the Topic Maps Processing Model (TMPM4).

The topic map defines TMPM4 association templates that model 11 of the 15 core DC metadata elements. The element **creator** is typical: topics represent the resource and the value, and they are related via an association whose type corresponds to the metadata element (‘creator’). Association types and role types are assigned PSIs in the <http://www.topicmapping.com/psi/dc/dc.html#> namespace; role types also have the DC identifiers as additional subject identifiers.

The following code shows how a **creator** property would be expressed:<sup>3</sup>

```
dcc:at-resource-creator(  
  dcc:role-resource : the-document,  
  dcc:role-creator  : the-creator )
```

---

<sup>1</sup> Note: The goal of the current work is to standardize the usage of all four vocabularies documented in [3].

<sup>2</sup> `psi1.xtm` was the set of XTM 1.0 Published Subject Indicators (PSIs) published in December 2000 by TopicMaps.Org but subsequently withdrawn. It can be retrieved via the WayBack Machine at [web.archive.org](http://web.archive.org) as <http://www.topicmaps.org/xtm/1.0/psi1.xtm>.

<sup>3</sup> The notation used for examples is the Compact Topic Maps syntax, as defined in the draft of 2007-09-09 [10]. The syntax is not yet stable and changes are likely.

Nine other elements are treated in essentially the same way, the exception being **relation**, which is modeled as an “untyped relationship between resources”. The remaining four elements are modeled as follows:

- **title**: an untyped base name is assigned to the topic that represents the resource
- **description**: an occurrence of type ‘description’ is assigned to a topic that represents the resource
- **subject**: an untyped occurrence whose value is the resource is assigned to a topic that represents the subject
- **identifier**: [this element is marked as “to be done” with the comment “not sure if and how to handle this”]

The model requires 23 published subject identifiers in order to cater for 12 of the 15 core elements (**title** does not require additional PSIs since it is modeled as a base name, and the treatment of **subject** and **identifier** is incomplete).

To summarize: the Algermissen proposal uses names, associations and occurrences to handle 14 of the 15 core elements; associations are used in two different ways (typed, as in **creator**, and untyped, as in **relation**); occurrences are also used in two different ways (the resource “owns” the occurrence in the case of **description**, whereas the value “owns” it in the case of **subject**).

Pepper 2003 [11], reproduced in Appendix A, is an example of an approach pioneered by Ontopia and distributed as the file `dc.tmm` with the *Omnigator* browser. It can be summarized as follows:

- **title** is represented as an untyped topic name
- **description**, **date** and **rights** are represented as occurrences
- all other core elements (except **identifier**) are represented as associations
- there is no example for **identifier** so its intended use is unclear
- the DC identifiers (e.g. <http://purl.org/dc/elements/1.1/creator>) are used as subject identifiers for occurrence types and association types
- PSIs are defined for the concepts ‘resource’ and ‘value’, respectively (in the namespace <http://psi.ontopia.net/metadata/#>) for use as role types.

Maicher 2006 [12] represents a similar approach, also published in the form of a topic map. It has the advantage of covering all four sets of terms documented in [3], but since it lacks documentation and complete examples, there is no way of knowing exactly how it is intended to work in practice. The two short examples in the topic map (showing metadata assigned to the topic map itself, and to the topic representing the subject “Der Schrei”) include the following mappings:

- associations: **language**, **type**, **source**
- occurrences: **description**, **publisher**, **references**, **title**

The examples given for **creator** and **created** (the latter an “element refinement” of **date**) indicate that they can be treated as *either* occurrences *or* associations. Based on this one can assume that the intention is for users to be able to choose whether to use an occurrence or an association in any given instance.

#### 4 Steve Pepper

Ontopedia

Maicher follows Pepper in using the DC identifiers as subject identifiers, but also defines a parallel set (in the *http://psi.semports.org/dc/* namespace). He also follows Pepper in defining two role types to be used across the board whenever a metadata element is represented as an association; these are called ‘start’ and ‘end’. No indication is given of how to handle the **identifier** metadata element.

### 3 Discussion

Assigning metadata to resources is equivalent to making statements about topics. It is therefore natural to represent the assignment of property-value pairs as statements of various kinds. In the case of the two DC vocabularies that define metadata elements (and element refinements) the key issue is to decide whether to represent a given property as a name, association, or occurrence.<sup>4</sup> This section discusses the issues involved before outlining a proposal for standardization.

#### 3.1 Simplicity vs. naturalness

The simplest possible solution would be to represent all DC metadata property-value pairs as occurrences, the value of which would be either a string or a URI. While this is eminently feasible, it would not produce results in keeping with the Topic Maps way of thinking.

One of the causes of the findability problem (which Topic Maps, of course, exists to solve) is the fact that much existing metadata consists simply of string values, which are subject to the well-known problems of homonymy and synonymy. In order to facilitate the aggregation of information, the Topic Maps paradigm encourages explicit reference to subjects via topics, rather than implicit reference via the strings that are used to name them: in other words, the approach is *subject-centric*. Values of properties should therefore be represented as topics wherever possible, and this means using associations rather than occurrences to represent property-value pairs.

Certain properties, however, are such that it would be counter-intuitive to represent them as associations.<sup>5</sup> An obvious example is **description**, which in Dublin Core typically consists of a sentence or two of text. It would engender unnecessary overhead if this were to be represented as a topic; a more natural solution is to use an occurrence. Another example is **title**, which is clearly a naming property, and is therefore most naturally represented as a topic name.

The only advantage of a “simpler” scheme based solely on occurrences is that it is easier to explain; its disadvantage is that it is less natural and leads to less useful results, and it is therefore to be rejected. The current proposal adopts the policy of using names, associations and occurrences as appropriate.

---

<sup>4</sup> Handling the two vocabularies that define values and schemes for encoding values is trivial since terms belonging to these must clearly be mapped to topics.

<sup>5</sup> At least using the Topic Maps Data Model. It would be less counter-intuitive at the level of the Topic Maps Reference Model, which lacks the explicit semantics of the TMDM.

### 3.2 A more flexible scheme

A second issue is whether to mandate the use of one (and only one) kind of statement for any given metadata element, or whether to allow some degree of flexibility, perhaps coupled with certain admonitions concerning best practice. Algermissen and Pepper adopt the former approach, and Maicher the latter.

The most obvious example of this question arising is with elements which should ideally be represented as associations, but where it might sometimes be inconvenient to represent the value of the property as a topic. Many elements carry comments stating that “recommended best practice is to use a controlled vocabulary” (in which case an association, with the property value represented as a topic, would clearly be most appropriate). However this advice may sometimes not be followed and the value of the property in question might simply be entered as a string. In such cases one might question the utility of creating a topic about which nothing is known except its name: the situation is scarcely better than using a metadata property whose value is a string.

In our opinion, the importance of encouraging best practice outweighs this. By requiring the use of associations in such cases, a point is made about the “right way” to do things, and this will hopefully lead to the creation of more maximally useful topic maps. The current proposal is therefore prescribes exactly which kind of statement to use for each element, and mandates the use of associations in all but exceptional cases.

### 3.3 Role types

Given the fact that associations are recommended for most elements (9 out of 15 in the basic element set), it becomes necessary to define appropriate role types. While it would be possible to define a different pair of role types for each association type, there seems to be no good reason to do so.<sup>6</sup> Following the principle of Occam’s razor,<sup>7</sup> the current proposal recommends using the same pair of role types for every association type. These obviously need to be very generic; following Pepper (2003), the subjects ‘resource’ and ‘value’ are proposed, since these correspond directly to Dublin Core terminology:

```
%prefix dctm http://psi.topicmaps.org/iso29111/
dctm:resource - "Resource" .
dctm:value    - "Value" .
```

### 3.4 Subject identifiers

As we have seen, both Algermissen (2001) and Maicher (2006) create new PSIs for every DC element (in addition to reusing the DC identifier), whereas Pepper (2003)

---

<sup>6</sup> There are some who regard the reuse of role types across association types as bad practice. They are invited to state their objections.

<sup>7</sup> *Entia non sunt multiplicanda praeter necessitatem* (entities should not be multiplied unnecessarily).

simply reuses the existing identifiers. In the absence of any stated reason for the duplication it seems sensible to again apply Occam's razor and avoid creating new subjects unnecessarily. The current proposal therefore reuses the DC identifiers without creating new ones (except for the two role types described above).

### 3.5 The *identifier* element

All previous proposals have been unclear on the handling of the **identifier** element, whose DC definition is "an unambiguous reference to the resource within a given context". The accompanying comment states that "recommended best practice is to identify the resource by means of a string conforming to a formal identification system." Most identifiers tend to be URIs, but it is important to note that they need not be.

Now, since the value of an occurrence can be a string, the use of occurrences to represent **identifier** elements would provide a general solution. However, since Topic Maps has two additional constructs that carry identification semantics – subject identifiers and subject locators – it is necessary to investigate the possibility of using these instead.

Both of the latter are defined as 'locators' in the Topic Maps standard, and locators are simply strings that conform to some locator notation (note that they need not be URIs). This means that, from the point of view of their syntax and data type, either subject identifiers or subject locators could be used to represent **identifier** elements. The key question is whether this would convey the correct semantics. To answer this question, it is necessary to consider the following definitions:

- **subject identifier**: a locator that refers to [an information resource] in an attempt to unambiguously identify [a subject]
- **subject locator**: a locator that refers to the information resource that *is* the subject of a topic

A subject identifier involves *indirect* identification: it locates an information resource that is *different from* the subject that it identifies. A subject locator involves *direct* identification: it both locates and identifies the same information resource.

The question to be answered is whether one or other of these conditions can be assumed to exist *in every case* in which **identifier** is used. The answer is clearly no: The resource to which DC metadata is assigned is very often an information resource and its URL is often the value of its **identifier** property; in such cases it would be appropriate to use subject locators. However, a resource *need not* be network retrievable (it could, for example, be a human being), and in such cases subject locators would be totally inappropriate.

We are therefore faced with the following choice:

1. EITHER define a conditional mapping rule ("use a subject locator when the value of **identifier** is a locator which directly identifies the resource, and use subject identifier when it is a locator which indirectly identifies the resource"). Such a rule would be complicated to explain, subject to incorrect usage, and not easily amenable to automated processing.

2. OR define a simple rule (“use an occurrence”), which does not take full advantage of the semantics available in the Topic Maps model, but can be expected to be applied with great consistency.

In our opinion, consistency is the more important factor and so the current proposal mandates the second alternative.

### 3.6 Conclusion and proposed synthesis

Based on the preceding discussion of issues raised by earlier work, the approach to expressing DC metadata in Topic Maps proposed in this paper is as follows:

1. Regard Dublin Core resources as subjects.
2. Use statements to assign DC properties.
3. For each property, specify which kind of statement to use.
4. Recommend the use of associations for all elements *except* —
  - elements that are naming properties and thus more naturally represented as topic names (e.g. **title**); and
  - elements whose values consist of descriptive text or structured data types and are thus more naturally represented as occurrences (e.g. **description, date**).
    - For each property, specify which kind of statement to use.
5. Reuse existing identifiers.

The next section contains a complete set of declarations for all 85 Dublin Core terms in all four term sets, and also declarations for role types.

## 4 Declarations and examples

### 4.1 Role types

```
%prefix dctm http://psi.topicmaps.org/iso29111/
resource dc:resource - "Resource" .
value dc:value - "Value" .
```

### 4.2 DCMES

```
%prefix dc http://purl.org/dc/elements/1.1/
# name types
title dc:title - "Title"

# occurrence types
date dc:date - "Date"
description dc:description - "Description"
identifier dc:identifier - "Identifier"
rights dc:rights - "Rights"

# association types
contributor dc:contributor - "Contributor"
```

## 8 Steve Pepper

### Ontopedia

```
coverage dc:coverage - "Coverage"
creator dc:creator - "Creator"
format dc:format - "Format"
language dc:language - "Language"
publisher dc:publisher - "Publisher"
relation dc:relation - "Relation"
source dc:source - "Source"
subject dc:subject - "Subject"
type dc:type - "Type"
```

### 4.3 Elements and element refinements

The term set *Other Elements and Element Refinements* defines additional properties, many of which are refinements of the core 15 elements of the DCMES. Element refinements are represented in the same way as the element that they refine. The remainder are represented according to the principle given in the fourth bullet of section 3.6, above:

```
%prefix dct http://purl.org/dc/terms/

# name types
alternative dct:alternative - "Alternative"

# occurrence types
abstract dct:abstract - "Abstract"
accessRights dct:accessRights - "Access Rights"
available dct:available - "Available"
bibliographicCitation dct:bibliographicCitation - "Bibliographic
Citation"
created dct:created - "Created"
dateAccepted dct:dateAccepted - "Date Accepted"
dateCopyrighted dct:dateCopyrighted - "Date Copyrighted"
dateSubmitted dct:dateSubmitted - "Date Submitted"
educationLevel dct:educationLevel - "Audience Education Level"
extent dct:extent - "Extent"
instructionalMethod dct:instructionalMethod - "Instructional Method"
issued dct:issued - "Issued"
license dct:license - "License"
medium dct:medium - "Medium"
modified dct:modified - "Modified"
provenance dct:provenance - "Provenance"
tableOfContents dct:tableOfContents - "Table Of Contents"
valid dct:valid - "Valid"

# association types
accrualMethod dct:accrualMethod - "Accrual Method"
accrualPeriodicity dct:accrualPeriodicity - "Accrual Periodicity"
accrualPolicy dct:accrualPolicy - "Accrual Policy"
audience dct:audience - "Audience"
conformsTo dct:conformsTo - "Conforms To"
hasFormat dct:hasFormat - "Has Format"
hasPart dct:hasPart - "Has Part"
hasVersion dct:hasVersion - "Has Version"
isFormatOf dct:isFormatOf - "Is Format Of"
isPartOf dct:isPartOf - "Is Part Of"
isReferencedBy dct:isReferencedBy - "Is Referenced By"
isReplacedBy dct:isReplacedBy - "Is Replaced By"
isRequiredBy dct:isRequiredBy - "Is Required By"
isVersionOf dct:isVersionOf - "Is Version Of"
mediator dct:mediator - "Mediator"
references dct:references - "References"
replaces dct:replaces - "Replaces"
```

```

requires dct:requires - "Requires"
rightsHolder dct:rightsHolder - "Rights Holder"
spatial dct:spatial - "Spatial"
temporal dct:temporal - "Temporal"

```

#### 4.4 Encoding schemes

Encoding schemes (including DCMI Type) are always represented as topics.

```

%prefix dct http://purl.org/dc/terms/

Box dct:Box - "DCMI Box"

Dctype dct:Dctype - "DCMI Type Vocabulary"
DDC dct:DDC - "DDC"
IMT dct:IMT - "IMT"
ISO3166 dct:ISO3166 - "ISO 3166"
ISO639-2 dct:ISO639-2 - "ISO 639-2"
LCC dct:LCC - "LCC"
LCSH dct:LCSH - "LCSH"
MESH dct:MESH - "MeSH"
NLM dct:NLM - "NLM"
Period dct:Period - "DCMI Period"
Point dct:Point - "DCMI Point"
RFC1766 dct:RFC1766 - "RFC 1766"
RFC3066 dct:RFC3066 - "RFC 3066"
TGN dct:TGN - "TGN"
UDC dct:UDC - "UDC"
URI dct:URI - "URI"
W3CDTF dct:W3CDTF - "W3C-DTF"

```

When an encoding scheme is used to annotate the value of a property, the corresponding statement should be scoped accordingly, thus:

```

dc:subject ( dctm:resource : topicmap, dctm:value : Opera )
@dct:LCSH

```

This association represents the `dc:subject` property whose value (represented by the topic with the ID `Opera`) is a term from the Library of Congress Subject Headings encoding scheme.

#### 4.5 DCMI type vocabulary

The terms in the DCMI Type Vocabulary are represented as topics that play the role of ‘value’ in associations of type **dc:type**:

```

%prefix dctype http://purl.org/dc/dctype/

Collection dctype:Collection - "Collection"
Dataset dctype:Dataset - "Dataset"
Event dctype:Event - "Event"
Image dctype:Image - "Image"
InteractiveResource dctype:InteractiveResource - "Interactive
Resource"
MovingImage dctype:MovingImage - "Moving Image"
PhysicalObject dctype:PhysicalObject - "Physical Object"
Service dctype:Service - "Service"
Software dctype:Software - "Software"
Sound dctype:Sound - "Sound"

```

## 10 Steve Pepper

### Ontopedia

```
StillImage dctype:StillImage - "Still Image"
Text dctype:Text - "Text"
```

## 4.6 Example

The following example shows how to apply the preceding declarations in order to assign metadata to a topic map. We assume they are collected in a separate file called `dc-declarations.ctm` which is included in our topic map. The topic map is reified and statements are made about the reifying topic using the Dublin Core vocabulary:

```
%include dc-declarations.ctm
%prefix o http://psi.ontopedia.net/

# reify the topic map
~ topicmap

topicmap
- title : "DCMI Metadata Terms"
  date : 2007-07-04
  description : "A Topic Maps representation of DCMI Metadata Terms"
  source : http://dublincore.org/documents/dcmi-terms/
  isReplacedBy : "Not applicable"

creator( resource : topicmap, value : o:Steve_Pepper )
contributor(resource : topicmap, value : o:Patrick_Durusau )
contributor(resource : topicmap, value : o:Sam_Oh )
format(resource : topicmap, value : o:LTM )
```

## 5 The subject-centric advantage

Having clarified the basic approach and defined the necessary vocabulary we now return to the issue raised in sections 3.1 and 3.2: the extent to which occurrences should be used rather than (or in addition to) associations. The following two examples demonstrate the two approaches:<sup>8</sup>

### 2-legs example (not recommended)

```
*e21 isa: entry
- title: "Dublin Core in Topic Maps"
  date: 2007-08-07
  subject: "Subject-centric computing"
  subject: "Dublin Core"
  creator: "Steve Pepper"
  contributor: "Dmitry Bogachev"
  description: "Topic Maps and Dublin Core complement each other"
```

### 4-legs example (recommended)

```
*e41 isa: entry
- title: "About Dublin Core in Topic Maps"
  date: 2007-08-07
```

---

<sup>8</sup> The code examples have been simplified for reasons of clarity. In particular, template declarations are assumed but not shown in the 4-legs example.

```

subject: Subject-centric_computing
subject: Dublin_Core
creator: Steve_Pepper
contributor: Dmitry_Bogachev
description: "Topic Maps and Dublin Core complement each other"

```

The difference between these two examples may seem subtle but it can be extremely significant. In the 2-legs example, the values of the subject, creator and contributor properties are strings; in the 4-legs example, they are topics. For a single resource description viewed in isolation, the latter provides little extra value and involves some overhead. But for a description set consisting of multiple descriptions, the advantages of the subject-centric approach soon become apparent. The subjects “Subject-centric computing”, “Dublin Core”, “Steve Pepper” and “Dmitry Bogachev” represented by the topics in the 4-legs example become additional points of collocation and thus improve findability. Each of them can potentially be associated with any number of other subjects in multiple ways and thus new navigation paths become available. And each of these subjects offers a new potential for linking this topic map with information in other, related topic maps that may not necessarily represent Dublin Core metadata.

## 6 The DCMI Topic Map

In order to support the analysis performed in this paper, an easily navigable, richly hyperlinked, and queryable representation of the document *DCMI Metadata Terms* [3] was generated – as a topic map [13]. It includes the following:

- topics for each metadata term
- the properties of each term and its recommended form of representation in Topic Maps
- associations linking each term to the set to which it belongs.
- associations linking elements to their refinements
- associations linking encoding schemes to the properties that they qualify

## 7 Conclusions

This paper has discussed previous work and the issues involved in representing Dublin Core metadata in Topic Maps. It has proposed a unified approach suitable for standardization by ISO and pointed out some of the advantages of the subject-centric approach.

## References

- [1] *Dublin Core Metadata Element Set, Version 1.1*. DCMI Recommendation (2006-12-18). <http://www.dublincore.org/documents/dces/>.

## 12 Steve Pepper

### Ontopedia

- [2] ISO 15836-2003. *Information and documentation — The Dublin Core metadata element set*. ISO, Geneva (2003).
- [3] *DCMI Metadata Terms*. DCMI Recommendation (2006-12-18). <http://www.dublincore.org/documents/dcmi-terms/>.
- [4] *Dublin Core Abstract Model*. DCMI Recommendation (2007-06-04). <http://www.dublincore.org/documents/abstract-model/>.
- [5] *Expressing Dublin Core in HTML/XHTML meta and link elements*. DCMI Recommendation (2003-11-30). <http://dublincore.org/documents/dcq-html/>.
- [6] *Guidelines for implementing Dublin Core in XML*. DCMI Recommendation (2003-04-02). <http://dublincore.org/documents/dc-xml-guidelines/>.
- [7] *Expressing Simple Dublin Core in RDF/XML*. DCMI Recommendation (2002-07-31). <http://dublincore.org/documents/demes-xml/>.
- [8] Pepper, S.: NP for Technical Report - Information Technology - Topic Maps - Expressing Dublin Core Metadata using Topic Maps. ISO/IEC JTC 1/SC 34 N0758 (2006). <http://www.jtc1sc34.org/repository/0758.htm>.
- [9] Algermissen, J.: *A Processing Model for HTML using Dublin Core* (2001-11-19). <http://www.topicmapping.com/psi/dc/dc.html>
- [10] Heuer, L.; Hopmans, G.; Oh, S.; Pepper, S.: *ISO/IEC CD 13250-6 Information Technology - Topic Maps - Compact Syntax (CTM)*. ISO/IEC JTC 1/SC 34 N0905 (2007-09-09). <http://www.jtc1sc34.org/repository/0905.htm>.
- [11] Pepper, S.: *LTM example for Dublin Core metadata*. v 1.4 dc-example.ltm (2003-09-18)
- [12] Maicher, L.: *Dublin Core Metadata Terms*. Topic Map (2006). <http://www.informatik.uni-leipzig.de/~maicher/topicmaps/DCMT.ltm>
- [13] Pepper, S.: *DCMI Metadata Terms Topic Map*. [http://www.ontopedia.net/omnigator/models/topicmap\\_nontopoly.jsp?tm=DublinCore.ltm](http://www.ontopedia.net/omnigator/models/topicmap_nontopoly.jsp?tm=DublinCore.ltm).

## Appendix A

```
/* LTM example for Dublin Core metadata
-----
   $Id: dc-example.ltm,v 1.4 2003/09/18 12:55:46 pepper Exp $ */

#INCLUDE "dc.ltm"
#TOPICMAP this-tm

[tm-topic = "LTM Dublin Core Example" @"#this-tm"]

/* DC properties modelled as occurrences: */
{tm-topic, Description, [[This topic map provides an example in LTM
  of how to add Dublin Core metadata to a topic map.]]}
{tm-topic, Date, [[Date: 2003/09/18 12:55:46 $]]}
{tm-topic, Rights, [[Copyright (c) 2003, Ontopia. This topic map may
  be freely distributed in its current form provided this copyright
  notice is left intact. Portions of this topic map may be copied,
  modified, and reused without attribution for the purpose of adding
  metadata to other topic maps.]]}

/* DC properties modelled as associations: */
Creator(tm-topic : resource, pepper : value)
Subject(tm-topic : resource, dc : value)
Subject(tm-topic : resource, metadata : value)
Subject(tm-topic : resource, topicmaps : value)
Publisher(tm-topic : resource, ontopia : value)
Contributor(tm-topic : resource, larsga : value)
Type(tm-topic : resource, topicmap-type : value)
Format(tm-topic : resource, LTM-format : value)
```

```
Source(tm-topic : resource, na : value)
Language(tm-topic : resource, en : value)
Relation(tm-topic : resource, na : value)
Coverage(tm-topic : resource, na : value)

/* Topic used for non-applicable DC properties */
[na = "N/A"]

/* Topics used as values of real metadata for this topic map */
[en      = "English"
 @http://www.topicmaps.org/xtm/1.0/language.xtm#en]
[ontopia = "Ontopia" @http://psi.ontopia.net/ontopia/#ontopia]
[pepper  = "Steve Pepper"
 @http://psi.ontopia.net/ontopia/#pepper]
[larsga  = "Lars Marius Garshol"
 @http://psi.ontopia.net/ontopia/#larsga]
[dc      = "Dublin Core" @http://psi.ontopia.net/metadata/#dc]
[metadata = "Metadata" @http://psi.ontopia.net/metadata/#metadata]
[topicmaps = "Topic Maps"
 @http://psi.ontopia.net/iso13250/#topicmaps]
```