Mapping between the Dublin Core Abstract Model DCAM and the TMDM

Lutz Maicher

University of Leipzig, Johannisgasse 26,04103 Leipzig, Germany
maicher@informatik.uni-leipzig.de

Dublin Core is a widely used metadata vocabulary. There is ongoing standardisation work for the usage of DC-vocabularies in Topic Maps. Although the DC-vocabularies are already used in RDF, XML and HTML, they are defined independently of any particular encoding syntax. The DCMI Abstract Model (DCAM) is the metamodel for the DC vocabulary. It should facilitate the development of better mappings and interoperability. In a first step, this paper describes the relationships between the DCAM and the Topic Maps metamodel (TMDM). Afterwards, the directed DCAM→TMDM and TMDM→DCAM mappings are defined. The DCAM→TMDM mapping, combined with a serialisation specification for a Topic Maps notation, is an encoding guideline for this Topic Maps notation. For the ongoing standardisation of the usage of DC-vocabularies we propose a two layers approach. The first is the mapping defined here, which assures the interoperability between DC-metadata in Topic Maps and documented in other representation formats. The second layer provides authoring guidelines for Topic Maps authors describing DC-metadata. Strictly adhering to these authoring guidelines will assure that the created topic maps can be mapped to the DCAM and, much more important, become always be mergeable (irrespective of the DCAM in the background).

Introduction

Due to the subject-centricness of metadata, their representation in Topic Maps is obvious. One of the most widespread metadata vocabularies is developed by the Dublin Core Initiative (DCMI). Recently the standardisation of the *usage* of the DC-vocabulary in Topic Maps is urged on. This paper introduces a formal mapping between the metamodels of Topic Maps and the DC-vocabulary. We argue that the standardisation of the usage of DC terms in Topic Maps will be much more accurate, concrete and convenient on top of the mapping defined here.

Generally, the vocabulary specified by the DCMI is intended to be independent from the used representation methods, like HTML, RDF, or Topic Maps. This approach should assure the interoperability of metadata descriptions. For supporting this universality the DCMI has developed a metamodel for the DC-vocabulary, the Dublin Core Abstract Model (DCAM) [PNN+07].

As figure 1 illustrates, for the usage of the DC vocabulary (and all of its application specific extensions, the *application profiles*) in Topic Maps the relationship between the TMDM and the DCAM is important. According to the DCAM, all metadata about a *resource* is represented as a *description*. A bunch of related descriptions is called *description set*. Using standardised *encoding guidelines* these description sets can be serialised to RDF, XML, HTML or Topic Maps.

To publish a description set as topic map, it must be transformed into a valid TMDM instance. This transformation relies on a DCAM→TMDM mapping as defined in this paper. Afterwards, the produced TMDM instance can be serialised using a Topic Maps notation, like XTM, LTM or CTM. As figure 1 illustrates, each bundle of a DCAM→TMDM mapping and a serialization specifications for a Topic Maps notation is an *encoding guideline* for this Topic Maps notation.

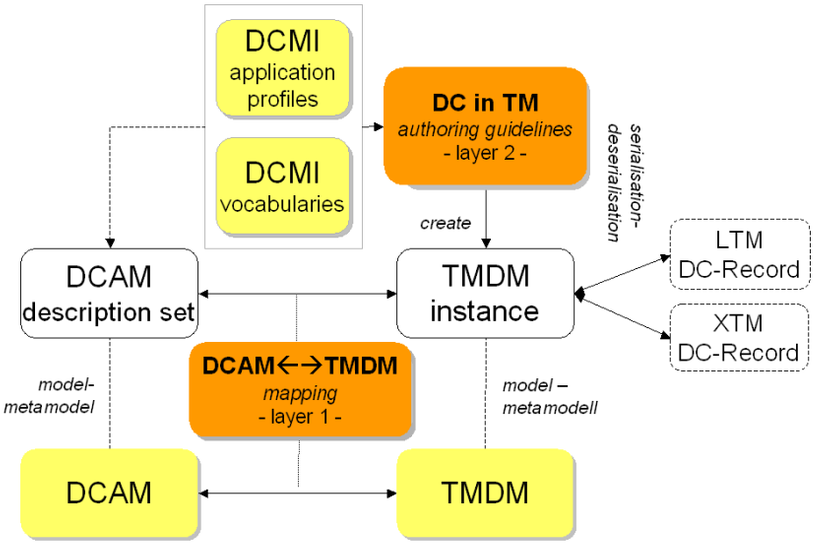

Figure 1 Standardising the usage of DC-vocabularies in Topic Maps

But for empowering the full interoperability, the mapping must be both directed. In this case, even a topic map representing DC-metadata can be transformed into a description set. Unfortunately, the TMDM and the DCAM have a different terminological complexity. In example the TMDM provides typed and scoped names, which is completely unknown in the DCAM. Due to these differences, an isomorphic mapping between TMDM and DCAM is not possible. Therefore this paper introduced a two directed mappings: DCAM→TMDM and TMDM→DCAM.

The recent standardisation efforts are about the **usage** of DC-terms in Topic Maps. It is obvious, that when authoring a topic map these terms should be used in a way that always valid description sets can be produced by applying TMDM→DCAM. Only in this case, the *interoperability* intended by the DCMI can be assured.

Summarised, when standardising the usage of DC-vocabularies we foresee the need of a two layer approach. The *first layer* is a DCAM←→TMDM mapping as

defined here, the *second layer* defines the authoring guidelines for creating topic maps representing DC-metadata. The mapping defined in the first layer assures the interoperability of DC metadata expressed in Topic Maps and metadata expressed in other representation formats. The need for the second layer is twofold. On the one hand, the application of the authoring guidelines will assure that such created topic maps will always be interoperable with metadata represented in other representation formats. On the other hand, the application of the authoring guidelines will assures the *mergeability* of the created topic maps (irrespective of the DCAM in the background). This mergeability is due to the fact, that the usage guidelines will exactly define how an observation, i.e. the information about the creator of a resource, has to be documented in a topic map. If all authors adhere to these guidelines, the same observations will be documented equally, with the result of the mergeability of the created topic maps [MB06]. The approach of this paper will assure both in parallel, *interoperability* and *mergeability*.

The reminder of this paper is as follows:

- *DCMI Meta Terms and the DCMT-Topic Map* provide a short overview about the terms defined by the DCMI, including a reference to the DCMT-topic map,
- *Dublin Core Abstract Model* provide a deep introduction into the three layers of the DCAM, including a comprehensive investigation into the relationships to the TMDM,
- *The identity crisis in the DCAM* discusses the differences in the identity disclosure mechanisms in DCAM and TMDM, including the implications for the defined mappings
- *DMCA→TMDM* and *TMDM→DCAM* properly define the mappings, followed by a short *Example*, and
- *Towards authoring guidelines for DC in Topic* Maps describes in short detail the foreseen authoring guidelines as second layer of the standardisation.

DCMI Meta Terms and the DCMT-Topic Map

The DC-vocabulary is separated into the following five categories. All terms are not bound to any syntax, but their intended usage is defined in [DCMI]:[1]

- *Elements*[2]*:* contributor, coverage, creator, date, description, format, identifier, language, publisher, relation, rights, source, subject, title, type

1 All terms labelled with (*) have the status „conforming". The necessity of these terms is proved by a specific user community. These terms are conforming to the DCAM, but do not belong to the core vocabulary.

2 The set *elements* are the 15 basic terms of the DC vocabulary. These terms are additionally standardised as ISO 15836:2003 [ISO15836]. According to [DCMI] the identifier of an element is the namespace *http://purl.org/dc/elements/1.1/* suffixed by the according term. In example, the subject identifier for *coverage* is *http://purl.org/dc/elements/1.1/coverage*.

- *Other Elements*[3]*:* accrualMethode*, accrualPeriodicity*, accrualPolicy*, audience, instructionalMethod*, provenance*, rightsHolder*
- *Element Refinements*[4] abstract, accessRights*, alternative , available, bibliographicCitation*, conformsTo, created, dateAccepted*, dateCopyrighted*, dateSubmitted*, educationLevel*, extent, hasFormat, hasPart, hasVersion, isFormatOf, isPartOf, isReferencedBy, isReplacedBy, isRequiredBy, isVersionOf, issued, license, mediator, medium, modified, references, replaces, requires, spatial, tableOfContents, temporal, valid
- *DCMI Type Vocabulary*[5]: Collection, Dataset, Event, Image, InteractiveResource, MovingImage, PhysicalObject, Service, Software, Sound, StillImage, Text
- *Vocabulary and Encoding Schemes*[6]*:* Box, DCMIType, DDC, IMT, ISO3166, ISO639-2, LCC, LCSH, MESH, Period, Point, RFC1766, RFC3066, TGN, UDC, URI, W3CDTF

All information about the DC-vocabulary given in [DCMI] is published in the „Dublin Core Metadata Terms"-topic map (DCMT-topic map) [Ma07].[7] Due to the DCMT-topic map authoring metadata in a topic map becomes more convenient, because it becomes always sufficient to only refer to any used term of the DC-vocabulary using the defined subject identifiers. Through merging in or requesting the DCMT-topic map (i.e. with TMRAP) all information about the terms, like labels, dependencies or definitions, can easily be acquired when needed.

---

3 *Other elements* are elements, which do not belong to the 15 core terms of the DC vocabulary. According to [DCMI] the identifier of an other element is the namespace *http://purl.org/dc/terms/* suffixed by the according term.

4 *Element Refinements* are terms which specify Elements in more detail. According to [DCMI] the identifier of an element refinement is the namespace *http://purl.org/dc/terms/* suffixed by the according term.

5 The *DCMI type vocabulary* is a generic, domain independent vocabulary for typing resources. According to the identifier of a type term is the namespace *http://purl.org/dc/dcmitype/* suffixed by the according term.

6 At the moment, the DCMT-topic map does not contain any information about vocabulary and enconding schemes. This will change in future.

7 Besides the officially defined identifiers in [DCMI] the DCMT-topic map additionally defines identifiers for each term of the DC-vocabulary in *one* namespace *http://psi.semports.org/dc/*. The reason for introducing these additional identifiers is convenience for the Topic Maps authors, because all terms are within one namespace. This avoids numerous confusions for the authors. As shown in [Ma07a], the synonymous use of different subject identifiers is unproblematic, if the synonymy is public disclosed as it is done by the DCMT-topic map.

Dublin Core Abstract Model (DCAM)

The Dublin Core Abstract Model (DCAM) [PNN+07] is the metamodel for the DC vocabulary. The DCAM has three models: the DCMI resource model, the DCMI Description set model, and the DCMI vocabulary model. In the following these models will be introduced and the relationships to the TMDM as metamodel for topic maps will be investigated in detail.

DCMI resource model

The *DCMI resource model* specifies how the "real" world is composed when looking through DCAM glasses. According to the DCMI resource model, the whole world is a set or *resources*. A resource is "anything that has identity". The DCMI resource interpretation is equivalent to resource definition in RDF [RDF]. Pursuing this chain and taking the RDF←→TMDM [PPG+06] discussion into account it becomes obvious, that a *resource* in DCAM is the same as a *subject* in Topic Maps.

According to the resource model, a resource is composed of *property/value pairs*. A *property* is a specific aspect, characteristic, attribute, or relation used to describe the resources. To each property a *value* is assigned, which is by definition a resource. In Topic Maps, not any assumptions about the composition of the subjects in the "real world" are defined anywhere.

Each resource becomes a *described resource* when a proxy is created and information is documented about it. The DCMI description set model in the next section defines the structure of these proxies, which are called descriptions.

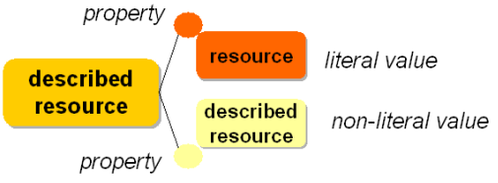

Figure 2 The DCMI Resource Model

Furthermore, the DCMI resource model separates between *literal values* and *non-literal values* of properties. The DCAM describes this difference as follows: "Each *non-literal value* may be the *described resource* in a separate *description* within the same description set – for example, a separate *description* may provide metadata about the person that is the creator of the *described resource*. A *literal value* can not be the *described resource* in a separate *description*." [PNN+07] Summarised, the resource which is a *non-literal value* is represented by a proxy in the models and the resource which is a *literal value* is only represented as a literal in the models. As Figure 2 illustrates, only the resource of a non-literal values becomes a described resource by its own.

DCMI description set model

The *DCMI description set model* specifies how information about resources – which are sets of property/value pairs – will be represented. Generally, the description set model defines the modelling constructs of the DCAM. To most extend, the DCAM ←→TMDM mapping is a mapping between the DCMI description model and the TMDM.

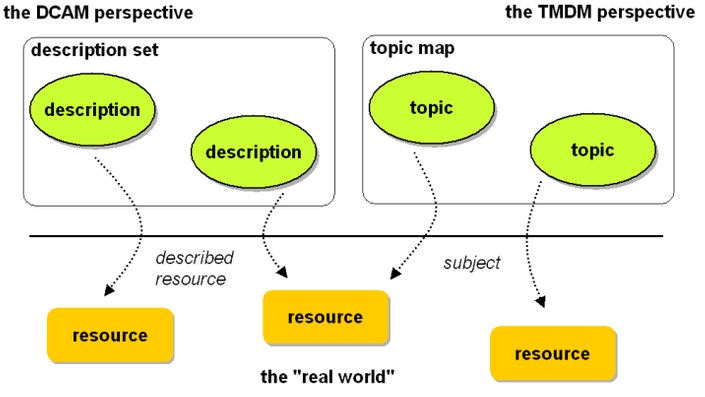

Figure 3 Relationship between descriptions and topics

From the DCAM perspective, the proxy of a described resource is a *description*. The counterpart in the TMDM is a topic, which is a proxy for a subject. Associated descriptions can be combined into *description sets*. The counterpart in the TMDM are topic maps, seen as TMDM instances. Figure 3 summarise these similarities. The same resource of the "real world" plays from the DCAM perspective the role of "described resource" for its proxy (which is a *description* there) and plays from the TMDM perspective the role "subject" for its proxy (which is a *topic* there).

According to the DCMI description model, each description identifies the described resource by zero or **one** *described resource URI*. In contrast, in the TMDM the subject is described by a set of subject identifiers or subject locators. As discussed in the section "Identity crisis in the DCAM" this asymmetry avoids an isomorphic DCAM←→TMDM mapping.

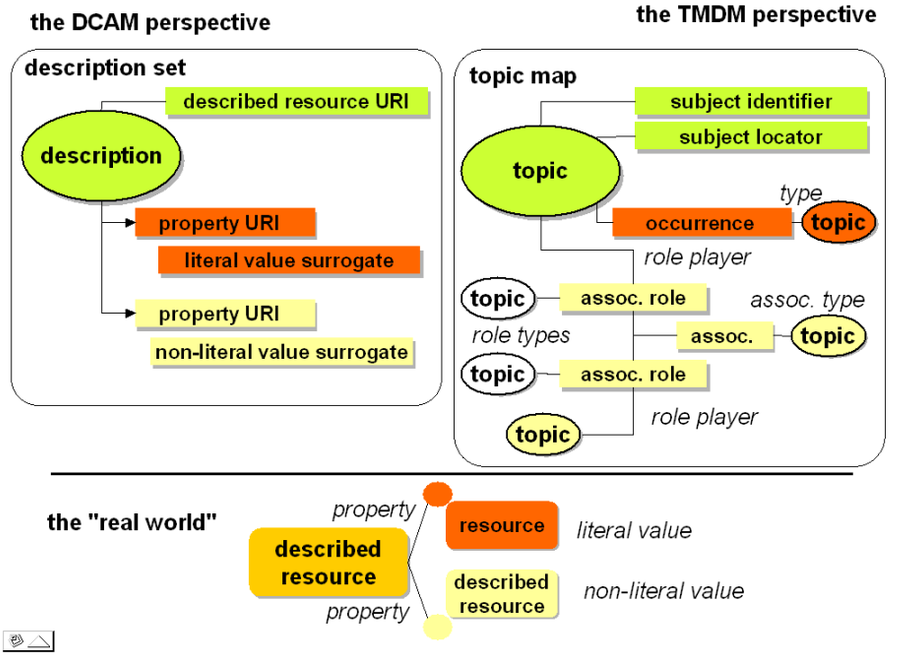

Figure 4 DCAM and TMDM relationship

Furthermore, the basis modelling components of descriptions are *statements*. Generally, each statement is a proxy for a property/value pair of a described resource. In the TMDM, topics are even used to make statements about subjects. In contrast to the very generic nature of any statements in the DCAM, the TMDM differs between topic names, variant names, occurrences, and associations. This even avoids an isomorphic DCAM←→TMDM mapping.

In the DCAM, each statement is a composition of a *property URI* and a *value surrogate*. The property URI is a URI which identifies the property. According to the resource model *each* value is a resource, so the value surrogate is always a proxy for the value.

Following the separation of literal and non-literal values in the resource model, the description set model differentiates between *literal value surrogates* and *non-literal value surrogates*. Each literal value surrogate is one *value string* (which is defined in detail below). Mostly a non-literal value surrogate is a value URI. This URI is a reference[8] to the description which represents the non-literal value. Furthermore, an *encoding scheme*[9] *URI* can be assigned to a non-literal value surrogate. This URI is

8 There is a subtle difference to the TMDM. In the DCAM the value URI is a reference. If there exists another description which uses this value URI as described resource URI, a valid reference is established. Otherwise, in the TMDM all values of properties which refer to other information items *are* these items, and not only references.

9 "Vocabulary Encoding Schemes indicate that the value is a term from a controlled vocabulary, such as the value "China - History" from the Library of Congress Subject Headings."

used to identify the vocabulary the used term (the value URI) is from. Alternatively to a value URI, the resource which is the non-literal value can also be described by a set of *value strings*.

From the TMDM perspective a property/value-pair having a literal value should be represented by an occurrence. Each occurrence is composed of a value (plus its datatype), which is the proxy for the literal value, and its type (which is another topic), which is the proxy for the property.

From the TMDM perspective a property/value-pair having a non-literal should be represented by an association. Each association is composed by a set of roles and a type. The topic which is the type of the association is the proxy for the property. In the TMDM each role is composed by a type and a role player. In the property/value relationship between a resource and non-literal value, the topic for the resource plays one role and the topic for the non-literal value plays the opposite role. The description set model is agnostic to the role types to use, they will be defined in the DCAM→ TMDM mapping. Figure 4 summarises the similarities of the DCMI description set model and the TMDM.

Generally, there exists a further distinction between the DCAM and the TMDM. The TMDM defines for all information items equality and merging rules. In consequence, two topic items representing the same subject (according to the defined rules) will be merged (according to the defined rules). This approach for using the identity of the proxies is not applied by the DCAM.

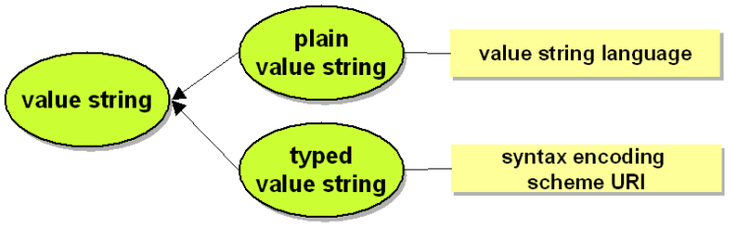

Figure 5 The composition of a value string

As last part the description set model defines value strings. A value string can be either a *plain value string* or a *typed value string*. A plain value string is intended to be human readable and may be tagged with the language used in this string. For tagging, the ISO language tags (like en-GB) should be used. A typed value string is tagged by a *syntax encoding scheme URI*. This URI identifies the syntax encoding scheme[10] under which the string should be interpreted.

A serialised description set is called *record*. There are encoding guidelines defined for diverse representation formats like RDF, XHTML and XML. The DCAM←→ TMDM mapping in the following bundled with the serialisation specification of a Topic Maps notation is an encoding guideline for this notation.

10 "Syntax Encoding Schemes indicate that the value is a string formatted in accordance with a formal notation, such as "2000-01-01" as the standard expression of a date."

The DCMI vocabulary model

In a last step, the DCMI vocabulary model is defined, which is an abstract model of the vocabularies *used* in the descriptions. According to this model, each vocabulary is a set of terms, which can be properties, classes, syntax encoding schemes, or vocabulary encoding schemes. Between these terms sub-property, sub-classes and type-instance relationships can be defined. Because the description set model is used to represent the relationships between the terms, it is rather in the scope of the second layer of the standardisation - the authoring guidelines - and not in the focus of this paper.

The identity crisis in the DCAM

As highlighted by Pepper and Schwab [PS03], using URIs for identifying resources might result in an identity crisis. The reason for such a crisis is due to the non-existence of the distinction between addressable and non-addressable subjects (or resources). This problem is well known from RDF (the related discussion is sketched in [PS03]), where one URI can be used to identify the retrievable information resource itself (to make statements about this information resource, like metadata) or this URI can be used to identify the subject which is described by the retrievable information resource (to make statements about the subject which is represented by the content of this information resource). Using the same URI for both homonymous interpretations directly leads to confusions and merging errors.

The DCAM does not introduce a mechanism for the disambiguation of URI semantics, whereas the TMDM introduces such a mechanism through the separation of subject locators and subject identifiers. It is obvious that this has direct consequences for the DCAM←→TMDM mapping.

When mapping a description to a topic map, a decision must be done whether the *described resource URI* should be used as subject identifier or as subject locator in the created topic item. In some cases the usage of a subject identifier might be appropriate (i.e. when information about a person is represented by the description), in some cases the usage of a subject locator might be appropriate (i.e. when information about a retrievable information resource is represented by the description).

Obviously, this decision is application depended, it depends on the used vocabulary. For example the value of the property *dc:creator* is mostly a person, so the topic representing this non-literal value surrogate should use the described resource URI as subject identifier. For other terms, other rules might apply. For this reason, the DCAM←→TMDM mapping will not make any appointments about it, the definition of all such rules are delegated to the standardisation of the authoring guidelines.

The inverse mapping direction from a topic item to a description yields additional problems. A topic item might have more than one *subject identifiers* or *subject locators*. (Due to the free mergeability of topic maps the authors will hardly be able to assure that a topic item has always only one sole identifier). For this reason, when

mapping a topic item into a description, one of these identifiers might randomly be chosen to be used as described resource URI in the description. This might yield a loss of information.

To avoid such a deranging forfeiture of information, it should be recommended to document the URI which desired as described resource URI as occurrence typed by *dc:identifier*. But it should not be an error when such an occurrence is not assigned to a topic item. In this case, the mapping algorithm will chose one subject identifier or subject locator randomly.

The DCAM→TMDM mapping

The purpose of the DCAM→TMDM mapping defined in the following is the transformation of a description set into a topic map. Each description D within a description set requires a topic item *r* in the topic map (which is seen as TMDM instance). This topic *r* will not be typed as "described resource proxy" because each topic should always be an agnostic representative of a subject, allowing any kinds of statements about it. The DC-relatedness of each statement is always derivable from the vocabulary used in these statements (see the TMDM→DCAM section).

> **Note:** When stated in the following "create an [topic map construct] of type [URI]" means, that to the value of the property *c.[type]* of the topic map construct a topic item *t* will be set, which has the URI as value of its property *t.[subject identifier]*.

When mapping a description D to a topic map, in the first step the topic item *r* representing the described resource must be created.

(a) **Create the resource topic.** Create a new topic item *r* which will be the proxy of the resource in the topic map.

(b) **Assign identity to the resource topic.** If assigned to D, the *described resource URI* will be set to *r.[subject locators]* (if this URI should be used as subject locator) or *r.[subject identifiers]* (if this URI should be used as subject identifier). According to the discussion in the previous section, the decision about the use of the *described resource URI* as subject locator or subject identifier is delegated to the authoring guidelines. As only constraint defined here, the decision must be compliant to the TMDM semantics.

(c) **Assign identity of the resource topic as occurrence.** Create an occurrence item *o* of type *dc:identifier*, assign the *described resource UR* to *o.[value]* and add *o* to *r.[occurrences]*.

In the next step each statement S of the description D must be mapped into the topic map. If a statement S has a non-literal value surrogate an association must be added to the topic map. In this case, the following has to be done:

(a) **Create the value topic.** Create a new topic item *v* which will be the proxy of the non-literal value in the topic map.

(b) **Assign identity to the value topic.** If available, assign the *value URI* of the non-literal value surrogate to *v.[subject locator]* (if this URI should be used as subject locator) or to *v.[subject identifier]* (if this URI should be used as subject identifier). The decision about the usage of the value URI as subject locator or subject identifier is not be specified here, but the decision must be compliant to the TMDM semantics.

(c) **Assign identity of the value topic as occurrence.** If *a value URI* is available, create an occurrence item *o* of type *dc:identifier*, assign the value URI of the non-literal value surrogate to *o.[value]* and add *o* to *v.[occurrences]*.

(d) **Assign vocabulary encoding scheme URI as occurrence.** If a *vocabulary encoding scheme URI* is available, create an occurrence item *o* of type *principles:vocabulary-encoding-scheme*, assign the vocabulary encoding scheme URI to *o.[value]* and add o to *v.[occurrences]*.

(e) **Assign value strings as occurrences.** If available, for each value string which is part of the non-literal value surrogate an occurrence item *o* of type *iso29111:valuestring* must be created according the guidelines for literal value surrogates below. Each o must be set to *v.[occurrences]*.

(f) **Create the typed association.** Create a topic item *at* and assign the *property URI* of the statement S to *at.[subject identifiers]*. Create an association item *a* in the topic map and assign the typing topic item *at* to *a.[type]*.

(g) **Create the resource role.** Create an association role item *ar1* of type *iso29111:resource*[11], assign *t* to *ar1.[player]* and assign *ar1* to *a.[roles]*.

(h) **Create the value role.** Create an association role item *ar2* of type *iso29111:value*, assign *v* to *ar2.[player]* and assign *ar2* to *a.[roles]*.

Note: The definition of the role types *iso29111:resource* and *iso29111:value* is necessary due to the directed nature of the statements in DC. For example, in an association of type *dc:creator* it is necessary to know which topic is the starting point (the created resource) and which topic is the endpoint (the creator of the resource). The authors of topic maps are completely free to create and use different role types such as *creator* or *publisher*. To be complaint to this document, these types must be subtypes (according to section 7 of [TMDM]) of *iso29111:resource* or *iso29111:value*.

If a statement S has a literal value surrogate a typed occurrence must be added to the topic map. In this case, the following has to be done:

(a) **Create and add the typed occurrence item.** Create a new occurrence item *o* which will be the proxy of the literal value in the topic map and add *o* to *r.[occurrences]*. The type of *o* is defined through the *property URI* of the Statement S.

11 The PSI for the role resource as well as the PSI for the role type value are proposed by Pepper [Pe07].

(b) **Add value.** Add the *value string* of the Statement S to *o.[value]*.

(c) **Add syntax encoding scheme URI.** If available, add the *syntax encoding scheme URI* to *o.[datatype]*.

(d) **Add value string language as scope.** If a value string language is available, a topic item *os* must be created. The value of os.*[subject identifiers]* is set to the value string language as described below and *os* is set to *o.[scope]*.

Note: According to the DCAM the language should be indicated using a "ISO language tag". Such a language tag is a language abbreviation according ISO 639. For creating subject identifiers, the namespace *lang* should be suffixed by these acronyms.

Finally, two further points should be outlined.

(a) **Naming the resource topic.** For a better readability of the created topic maps, the value of a property which has naming characteristics, like *title*, can be *additionally* assigned as unscoped and untyped name of *r*. But the value of such a property *must* additionally be represented as defined above, its usage as topic name is only informative.

(b) **Typing the resource topic.** For a better readability of the created topic maps, the type of a resource can be *additionally* represented as type-instance relationship, where *r* is playing the role of an instance. But in general all typing properties *must* always be represented as defined above, its further representation in such a type-instance relationships is only informative.

TMDM→DCAM mapping

The purpose of the TMDM→DCAM mapping defined in the following is the transformation of topic maps (which represent metadata using the DC vocabulary) into description sets. Due to the different terminological diversity of TMDM (which is multifaceted) and DCAM (which is more focused) such a transformation might always imply a loss of information (in most cases subject identifiers will be lost).

Furthermore, the mapping is error prone. Having the DCAM→TMDM mapping above, it will always be possible to create a valid topic map from any valid description set. But having a valid topic map, it is not possible to always create a valid description set out of this. For this reason, this mapping defines constraints which must be fulfilled by topic maps statements to assure that descriptions can be created out of them. If the constraints are not fulfilled, errors occur and statements of description can not be created.

The TMDM→DCAM mapping does only transform such information into the description set which is documented in the topic map using the DC vocabulary (or an approved application profile). All other information which is documented in the topic maps by using terms from other vocabularies (or terms of the DC vocabulary which are not used in the specified way) are *ignored* by the mapping defined here.

In a first step, for each topic item *r*, which represents a described resource according to the DCAM, a new description D must be created. Because topic items are not typed as "described resource proxy", eligibility is determined by the following rules:

(a) **Occurrence item.** The topic item *r* has at least one occurrence item *o* in its property *r.[occurrences]* which is typed by a term from the DC vocabulary (see above) or any approved application profile.

(b) **Role Player.** The topic item *r* plays at least in one association a role of type *iso2911:resource.* Furthermore this association must be typed by a term from the DC vocabulary (see above) or any approved application profile.

For each eligible topic item *r* a description D in the description set will be created. In a first step the identity must be assigned to the description:

(a) **Using the dc:identifier occurrence.** If available, the value of an occurrence item *o* in *r.[occurrences]* which is of type *dc:identifier* will become the *described resource URI* of the description D.

(b) **Using the subject identifiers or locators.** If such an occurrence is not available, one value from *r.[subject identifiers]* or *r.[subject locators]* can be used as *described resource URI* of the description D. If one of these values is a term from the DC vocabulary (or approved application profiles) this should be taken preferably, otherwise one value will be chosen randomly.

(c) **Using item identifiers.** If even subject identifiers or subject locators are not available, one randomly chosen value of *r.[item identifiers]* should be used as *described resource URI* of the description D.

In the next step all statements of D have to be created. The therefore needed information is either documented in typed occurrences or in typed associations.

If an occurrence item *o* from *r.[occurrences]* is typed by a term from the DC vocabulary (or an approved application profile) a new statement S with a literal value surrogate will be added to D as follows:

(a) **Add property URI.** One subject identifier of the topic item *ot* in *o.[type]* which is element of a DC vocabulary (or an approved application profile) will be used as *property URI* in S. If such a subject identifier is not available, an error occurs and the statement S can not be created.

(b) **Add value string.** The value string of the statement S is the value of *o.[value]*.

(c) **Add syntax encoding scheme URI.** If the value of *o.[datatype]* is not *XMLSchema:string* the *syntax encoding scheme URI* of the value string in statement S is set to this value. In this case, the *literal value surrogate* is a *typed value string*. Otherwise, the literal value surrogate is a *plain value string*.

(d) **Add value string language.** If the literal value surrogate is a plain value string and *o.[scope]* contains a topic item which has a subject identifier in the namespace *lang*, the *value string language* of the value string in statement S is set to the part of this identifier after the namespace *lang* (i.e. en-GB).

If an association item *a* is typed by a term from the DC vocabulary (or an approved application profile) and *r* is playing the role of type *iso2911:resource* in this association (whereby the topic item *v* is playing the role of type *iso2911:value*) a new statement S with a non-literal value surrogate will be added to D as follows:

(a) **Add property URI.** One subject identifier of the topic item *at* in *a.[type]* which is element of a DC vocabulary (or an approved application profile) will be used as property URI in S. If such a subject identifier is not available, an error occurs and the statement S can not be created.

(b) **Add value URI by occurrence.** If *v* has an occurrence *o* typed by *dc:identifier*, the value of *o.[value]* is used as *value URI* of the statement S.

(c) **Add value URI by identifiers.** Otherwise, if *v* has values in the properties *v.[subject identifier]* or *v.[subject locators]*, the *value URI* of the statement S is set to one of these values. If one of these values is a term from the DC vocabulary (or approved application profiles) this should be taken preferably, otherwise one value will be chosen randomly. If none value is available, none value URI is assigned to the non-literal value surrogate of the statement S.

(d) **Add vocabulary encoding scheme URI.** If *v* has an occurrence item *o* which is typed by *principles:vocabulary-encoding-scheme* the *vocabulary encoding scheme URI* of the non-literal value surrogate is set to the value of *o.[value]*. If more then one occurrence items of this type are assigned to *v*, one of them has to be chosen randomly.

(e) **Add value strings.** If *v* has occurrence items *o* which are typed by *iso29111:valuestring* for each of these items a *value string* of the non-literal value surrogate is created according to the specification for literal value surrogates above.

(f) **Error checking.** If neither a *value URI*, a *vocabulary encoding scheme URI* nor a *value string* can be assigned to the statement, an error occurs and the statement S can not be created.

Example

In this section a short example illustrates the mapping between DC descriptions and Topic Maps. The DC description provides metadata about a book. It is composed of one statement with a literal value surrogate and two statements with non-literal value surrogates. (For saving space, the necessary descriptions of the both creators of the book are not given in this example).

described resource URI	**doi:10.1007/11676904**
property URI	http://purl.org/dc/elements/1.1/type
value URI	http://purl.org/dc/dcmitype/Text
vocabulary enc. scheme URI	http://purl.org/dc/dcmitype/

property URI	http://purl.org/dc/elements/1.1/title
plain value string	Charting the Topic Maps Research and Applications Landscape
value string language	en-GB
property URI	http://purl.org/dc/elements/1.1/creator
value URI	mailto:maicher@informatik.uni-leipzig.de
property URI	http://purl.org/dc/elements/1.1/creator
value URI	mailto:jack.park@sri.com

Now the same information is presented as topic map, serialised in LTM 1.3. It should be underlined, that some information in a TMDM instance (like a datatype of an occurrence item) can not be represented using LTM. To get more information about the used terms from the DC-vocabularies the DCMT-topic map [Ma07b] can be merged in or requested.

```
#PREFIX dc @"http://purl.org/dc/elements/1.1/"
#PREFIX dctype @"http://purl.org/dc/dcmitype/"
#PREFIX lang @"http://www.topicmaps.org/xtm/1.0/language.xtm#"
#PREFIX iso29111 @"http://psi.topicmaps.org/iso29111/"
[id1 : dctype:Text = "Charting the Topic Maps …" @"doi:10.1007/11676904"]
   {id1 , dc:title, [[Charting the Topic Maps Research …]]} /lang:en-GB
   {id1 , dc:identifier, [[doi:10.1007/11676904]]}
dc:type(id1 : iso29111:resource, dctype:Text : iso29111:value)
[id2 @"mailto:maicher@informatik.uni-leipzig.de"]
   {id2 , dc:identifier, [[mailto:maicher@informatik.uni-leipzig.de]]}
[id3 @"mailto:jack.park@sri.com"]
   {id3 , dc:identifier , [[mailto:jack.park@sri.com]]}
dc:creator(id1 : iso29111:resource , id2 : iso29111:value)
dc:creator(id1 : iso29111:resource , id3 : iso29111:value)
```

Towards authoring guidelines for DC in Topic Maps

As already discussed in the introduction, for the standardisation of the DC/TM interoperability we foresee a two layers approach. The first layer defines the DCAM ←→TMDM mapping, as it is realised in this paper here. This mapping assures the *interoperability* of DC metadata expressed in Topic Maps and metadata expressed in other representation formats. The second layer is the definition of authoring guidelines for all terms of the DC-vocabularies for the creation of topic maps. The need for this layer is twofold. On the one hand, the application of these authoring guidelines will assure that such a created topic map will be interoperable with metadata represented in other representation formats. On the other hand, the application of the authoring guidelines will assure the *mergeability* of the created topic maps (irrespective of the DCAM in the background).

The standardised authoring guidelines should look like as follows:

First, it *must* be defined how a described resource and statements with literal-value and statements with non-literal values have to be represented when authoring a topic map. This standardisation must be strictly compatible to the TMDM→DCAM mapping defined here.

Second, for each term of the DC-vocabularies it *must* be decided, in which cases it should used as a property for a non-literal value and when it should be used as a property for a literal-value. No further specifications are mandatory for any term.

Third, guidelines for the representation of the described resources which are non-literal values *might* be defined. For example, best practice for choosing identifiers of persons, countries, dates, etc. can be defined. The more specific these specifications are, the better the mergeability of the resulting topic maps.

Fourth, it must be defined how relationships between terms of the DC-vocabulary (i.e. sub-property relationships, etc.) should be represented in Topic Maps to be compliant to the DCMI vocabulary model

Conclusion and further work

This paper has introduced a comparison and a mapping of the metamodel of the Dublin Core metadata vocabulary, the Dublin Core Abstract Model, and the metamodel of Topic Maps, the Topic Maps Data Model. Due to the different terminological expressivity, an isomorphic mapping between DCAM and TMDM is not possible. As consequence, two directed mappings has been introduced here. Especially the TMDM→DCAM mapping might imply a loss of information.

The purpose of the defined mappings is the assurance of the interoperability between DC metadata expressed in Topic Maps and DC metadata expressed in other representation formats.

On top of these mappings the authoring guidelines for DC-vocabularies in Topic Maps should be standardised. Defining such modelling methodologies assures, that *(1)* the created topic maps are always interoperable with the DCAM and *(2)* all created topic map are mergeable (irrespective of the DCAM in the background).

On the short term we foresee two further work packages. *First,* the TMDM→ DCAM mapping allows the specification of a *schema* which decides, whether a topic map represents DC-metadata correctly. Once TMCL is standardised, such a schema should be defined. Second, the specification of a DC metadata *filter* (view), which is a set of TMQL queries, is very close to this approach. Applying these queries will extract exactly the DC metadata represented in a topic map.

References

[DCMI] DCMI Usage Board (2006): *DCMI Metadata Terms*. Verfügbar unter: http://dublincore.org/documents/dcmi-terms/

[Ma07a] Maicher, L.: *The impact of semantic handshakes.* In: Maicher, L.; Sigel, A.; Garshol, L.M.: *Leveraging the semantics of Topic Maps.* LNAI 4438, Springer:Berlin (2006).

[Ma07b] Maicher, L.: *Dublin Core Meta Terms.* Topic map available at: http://informatik.uni-leipzig.de/~maicher/topicmaps/DCMT.ltm

[MB06] Maicher, L.; Böttcher, M. (2006): *Closing the Semantic Gap in Topic Maps and OWL Ontologies with Modelling Workflow Patterns.* In: Journal of Universal Computer Science, 12(Special Issue I-Know'06):261-269.

[PNN+07] Powell, A.; Nilsson, M.; Naeve, A. et al.: *DCMI Abstract Model.* Available at: http://dublincore.org/documents/abstract-model/

[PPG+06] Pepper, S.; Presutti, V.; Garshol, L.M. et al. (eds.): *Guidelines for RDF/Topic Maps Interoperability.* W3C Editor's Draft. Available at: http://www.w3.org/2001/sw/BestPractices/RDFTM/guidelines-20060630.html

[PS03] Pepper, S.; Schwab, S.: *Curing the Web's Identity Crisis. Subject Indicators for RDF.* In Proceedings of: XML Europe 2003.

[RDF] Hayes,P. (ed.): *RDF semantics. W3C recommendation.* Available at: http://www.w3.org/TR/rdf-mt/

Webreferences

[1] http://dublincore.org/
[2] http://dublincore.org/documents/abstract-model/
[3] http://dublincore.org/documents/dcmi-type-vocabulary/
[4] http://dublincore.org/resources/expressions/
[5] http://dublincore.org/usage/documents/principles/
[6] http://dublincore.org/usage/documents/process/#conforming

Namespaces

iso29111	*http://psi.topicmaps.org/iso29111/*
lang	*http://www.topicmaps.org/xtm/1.0/language.xtm#*
principles	*http://dublincore.org/usage/documents/principles/#*
XMLSchema	*http://www.w3.org/2001/XMLSchema#*