Mapping of the Dublin Core Metadata Element Set to the CIDOC CRM

Martin Doerr

Institute of Computer Science, Foundation for Research and Technology – Hellas
Science and Technology Park of Crete
P.O. Box 1385, GR 711 10, Heraklion, Crete, Greece
martin@ics.forth.gr

Technical Report 274, ICS-FORTH, July 2000

Abstract: The CIDOC CRM is the first ontology designed to mediate contents in the area of material cultural heritage and beyond, which has been submitted to ISO as proposal for an international standard. The Dublin Core Element Set can be regarded as the most important metadata standard of the library world and far beyond, to define basic finding aids for electronic resources by a minimal set of semantic fields or "access points". This report describes the semantic mapping of the current Dublin Core Element set to the CIDOC CRM and its latest extension. This work represents a proof of concept for the functionality the CIDOC CRM is designed for.

Keywords: Ontology, metadata, data structure mapping, Dublin Core, CIDOC CRM

1. Introduction

The CIDOC Conceptual Reference Model (see [CRM], [Doer99]), in the following only referred as "CRM", is outcome of an effort of the Documentation Standards Group of the CIDOC Committee (see "http:/www.cidoc.icom.org") of ICOM, the International Council of Museums, beginning in 1996. It is a domain ontology formulated in the form of an object-oriented semantic model aiming to solve the problem of semantic interoperability between museum data of various kinds and their relations to archive and library material.

After the model has found a stable form in 1998, and was accepted by CIDOC on the CIDOC Conference in Melbourne in this year, the Group decided that the model has to undergo a series of practice test in order to verify its utility and completeness in a so-called "Correlation Test Project". Particular importance was seen in the demonstration that and how the CRM is able the mediate between current relevant standards of the domain, that are regarded relevant to the scope set for the CRM. This is done via sets of mapping rules, that define how to transform descriptions given in the structures of one standard into an equivalent description in the CRM with the same intended meaning.

This process is expected to produce a set of extensions and improvements over the current version 2.2 of the CRM. The latter has been submitted to ISO TC46, SC4 for ballot in June 2000, (see http://www.niso.org/sc4ballt.html) and results from the Correlation Test Project are intended to support the discussion and finalizing processes foreseen by ISO. So far, CIDOC members have been creating mappings to the AMICO (http://www.amico.org/docs/dataspec.html] data model), the Dublin Core Elements, EAD [EAD] and the SPECTRUM [SPEC] data fields. In a meeting

between representatives of the CIDOC Documentation Standards Group and other stakeholders in the domain in Agios Pavlos, Crete, in June 2000, experiences from these mappings have been discussed and a set of extensions for the CRM have been proposed (the “Agios Pavlos Extensions", see http://www.ics.forth.gr/proj/isst/Activities/CIS/cidoc/docs/).

This report presents the first of those mappings, the mapping from the Dublin Core Element Set to the CRM carried out at ICS-FORTH, using CRM version 2.2 and the Agios Pavlos Extensions. The Dublin Core Element Set can be regarded as the most important metadata standard of the library world and far beyond, to define basic finding aids for electronic resources by a minimal set of semantic fields or “access points”. This mapping can be regarded as one of the necessary proofs of concept for the functionality the CRM is designed for.

2. Comparison between CRM and DC

The purpose of these mappings is to show the use of the CRM as a data transfer mechanism, and not as a metadata format. The distinction we wish to make here is that the CRM offers a mediation between alternative representations whereas a metadata format makes recommendations about which contents should be described for a kind of object. As such, the CRM often has redundant and sometimes over-detailed elements, and it declares properties in both directions. It does not make propositions about encoding details, identifier conventions and other implementation issues. In other respects the CRM and metadata standards are similar and can use the same formalism, e.g. RDFS/RDF. The CRM may be used to “inspire” “good” metadata.

The mappings presented here are not a reformatting mechanism. They are not meant to be reversible in a formal sense. Rather they describe how to transform descriptions in one structure into an equivalent description in the CRM structure with the same meaning, to the degree the contents under investigation fall under the scope of the CRM. More formally, we regard one DC record to refer to a real world domain of entities and one relevant state of affairs therein. The mapping mechanism will create a CRM instance (or “model” for the CRM), which approximates **the same state of affairs** with respect to the part of the domain it is designed for. In some cases, the CRM may commit to a slightly different conceptualization of the domain, and may use domain knowledge to “fill gaps”. (See also [Guar98])

The intention is to render optimally an information equivalent in the expert’s sense. We restrict the presentation to the semantically interesting equivalences which **need expert** knowledge, and leave questions like identifier creation, identity of intermediate or partially identified nodes, information packaging etc. to implementers or further studies (an example of a rigid mapping mechanism can be found in [Chri99]). We have deliberately used a format, we thought to be more comprehensive to the domain expert than the computer scientist, if successfully comments will show.

Dublin Core in particular has acquired a remarkable importance as a convention to “extract” a set of core data from a resource and as a finding aid. Its value is in the simplicity and in the breadth of the attempted scope. This seems to be defined in a pragmatic or experiment spirit: “let’s see

how far we can extend the semantic interpretation of our initial field definitions". To some extent, this approach encourages ambiguity: the looser the definition, the more flexible the interpretation and the wider its application. This may often yield surprising results when comparing different types of resources. E.g. by extending "resource" to include "parties", such as agents and people these latter may inherit a "creator" property. While this appears reasonable when applied to a group or legal body, it may sound strange or ambiguous when used for an individual human being. This does not mean of course, that no reasonable interpretation can be found, rather it means that several interpretations may seems equally plausible.

By contrast, the CRM has different intentions and adopts a different approach. It is based on a fundamental data model of basic intellectual constructs, SIS-TELOS [Mylo90], [Anal96, 98], which can be extended in a regular manner, following a methodology of semantic models similar to RDFS, to any level of granularity. The Simple Class level of SIS-TELOS used by the CRM in particular is nearly identical with RDFS [RDFS], and the CRM can be easily transformed into an equivalent RDFS form. Such models have recently become known in computer science as "ontologies". The CRM in particular is a top-level and domain ontology in the sense of Nicola Guarino [Guar98].

Extensions of a data model like SIS-TELOS or RDFS imply specialization:
1. The declaration of new subclasses under existing classes
2. The declaration of new properties
3. The declaration of new subproperties under existing properties

They further imply true extension:

4. The declaration of new root classes

For the CRM we propose a more "constraint extension", the formal properties of which have to be investigated, in order to preserve basic reasoning features between different extension. The motivation is that all new possible data paths should have well defined abstractions in the original model. Therefore mechanism 2 and 4 above should be disallowed. Instead we propose:

- The declaration of new subclasses under existing entities of the CRM,
- the declaration of new subproperties under existing properties of the CRM, and
- the indirection of existing properties of the CRM with intermediate entities being themselves subclasses of existing entities.

In other words, whereas the DC acquires its genericity by using "underspecified" notions, the CRM acquires genericity through constraint extensibility. Furthermore, whereas the DC makes proactive recommendations for developing finding aids, the CRM tries to interpret formats in a reactive manner. The definition of the relationship between CRM and DC has three benefits:

1. Given a mapping from the CRM to the Dublin Core, DC can exploit the finer granularity and more rigid definitions of the CRM. Provided that a given resource falls within the scope of the CRM, or a constraint extension to it, mapping of the schema should automatically entail

mapping to the Dublin Core. In other words, the CRM provides a language and a common intermediate format to provide unique definitions for creating Dublin Core records from richer formats. That such a need exists has turned out in the AQUARELLE project, where multiple groups mapped their richer data structures to CIMI access points independently, creating initially incompatible interpretations.

2. Given DC metadata about some resource, the mapping to the CRM provides a prescription for how to insert information encoded in such a record into a finer grained data structure. The CRM encodes domain knowledge, e.g. differences of physical and intellectual products, the relation of events, agents, dates and places etc., hence it can, to some extent, allow for the "recovery" of information which has been "flattened out" in a DC record.
3. The symmetry of the CRM with respect to all relations and involved entities (it is not "about one resource") allows information to be merged into descriptions of entities other than those initially intended. E.g. a set of DC records and others about paintings from Seurrat could be turned into a description of the person, and eventually even into a DC record about him.

Specific structural elements of the CRM can help resolve questions of grouping between DC elements. The CRM *events* between Actors (Agents) and Objects, for example, allow multiple dates and agents information in a DC record to be grouped correctly. Other relevant structures include composition relations (part-whole) of a set of entities, states and state transitions, use and intentions, and the distinction between physical embodiment and immaterial form.

Due to the width of scope of the Dublin Core, the mapping below has motivated some regular extensions to the CRM, and confirmed another set of extensions that has been already derived from mapping the CRM to the AMICO model. We hope that mappings of EAD, DC, FRBR [FRBR], and SPECTRUM will close any remaining gaps in the general coverage of the CRM. The mapping exercises so far completed have confirmed the validity of the approach and the stability of the entities and properties in the current proposal. We are therefore convinced that it is feasible to produce a solid proposal for standardization; one that will satisfy the basic interoperability needs of the respective cultural institutions - museums, archives and libraries - and that will also respect the diversity of the field of cultural heritage information. We believe that the CRM accurately represents certain fundamental concepts used in the cultural heritage domain. Given experts of the same domain and the same formalism, other models should, and in fact recently tend to do, produce comparable or even compatible structures, albeit with different names for entities and properties, more or less direct relations, more or fewer subclasses. (See also [Guar98] on intended models and conceptualizations).

3. Formalism

The CRM is a formal ontology in the form of a semantic model.
In a structured record,

- No distinction is made between the nature of an attribute value and kind of relation expressed by the attribute between the record as a whole and the attribute value.
- Referred discrete entities are not explicitly identified, but rather referred to by some representative values.
- Information is arranged (sequence, nesting etc.).

A semantic model,

- declares each attribute value as an a priori independent entity in the universe of discourse and connects such values symmetrically by directed links/arcs/properties .
- The entities acquire identity independent from the composition of descriptional elements in some data structure.
- The structure of a formal ontology is supposed to reflect the organization of the universe of discourse as we perceive it, rather than that of the document we create (e.g., if not given graphically, the presentation in a document will have to serialize a multiply connected net in some way). It has an "ontological commitment" to the "underlying conceptualization" of its creators [Guar98].

Let's regard an example of a structured record like:

```
struct painting {
        integer record_id;
        string title;
        string kind_of_title;
        struct painter {
                string artist_name; /* ULAN identifier*/
                string nationality;
                string contribution; /* role in the creation process */
                } artist;
        time creation_date_begin;
        time creation_date_end;
        string creation_date_comment;
        string creation_place; /* TGN identifier */
        string last_exhibition;
         } PAINTING;
```

In the following, CRM entities are referred by the unique id and name , eventually followed by a comment in parenthesis, like:

E7 Activity (of type "exhibition")

If a record with *record_id* = 5034 is to be transferred to the CRM, an instance of E22 has to be created with a unique identifier, e.g. "Object5034". But any other unique for the object combination of values can be used.

CRM links are referred by the unique id of the **applicable** entity in the given context, followed by the unique id of the entity from which the link is **inherited**, the name of the link and the name of the referred entity, like:

E22 (E7) was used for: Activity

We use as name of the link only the part associated with the direction applicable for the given context. E.g. We present the link "E11 Modification. has produced (was produced by)" as

E22 (E11) was produced by: Modification

if I want to read it from the product rather than from the event.

We have therefore chosen a mapping format, where for each PAINTING field we declare the correspondence of the **value** to a CRM entity, like:

last_exhibition = E7 Activity (of type "exhibition")

This means that for a transfer from PAINTING to CRM, the contents of the "*last_exhibition"* field are used to create a respective unique identifier of an E7 entity instance. The contents may also be used to derive additional attributes of the E7 entity instance, if they can be parsed, e.g. "Documenta 1998, Kassel" may be used to find date, title and place. The E7 instance is also assigned the type "exhibition" from a thesaurus.

In the sequence we declare the correspondence of the **relation** expressed by the field to a CRM link, like:

painting -> last_exhibition = E22 (E7) was used for: Activity

This means for a transfer from PAINTING to CRM that a property instance of type "was used for" is created, that connects the instance of E22 "Object5034" representing our example painting to the above created instance of E7 "Documenta1998, Kassel".

For a transfer from CRM to PAINTING, the instance of E7, that has type "exhibition" or a narrower term of it, is connected by "was used for" to the respective E22 entity, and that has the latest date associated, will be used to fill the field *"last_exhibition"* of record 5034.

If a substructure corresponds to a CRM entity, I declare the correspondence of the **relation** expressed by one of its field with respect to the substructure name, rather than the catalog record, like:

painter -> nationality = E21 (E1) has type: Type

If the relation expressed by an PAINTING field corresponds to a path with intermediate entities in the CRM, I have marked it as "(JOIN)", like:

painting -> painter = E22 (E11) was produced by – E12 (E7) carried out by: Actor
(JOIN)

This means, that for a transfer from PAINTING to CRM, an **intermediate** instance of E12 has to be created, with an identifier uniquely defined by the circumstances, like "Object5034-

production". In the sequence, it is inserted appropriately between the node "Object5034" and the Actor instance, e.g. "Hans Blumenbaum Jr". This intermediate node may be bare of other attributes. However in this example, missing attributes can be found in the *painting* record, like *creation_date_begin.* Joins may be double or triple, requiring multiple intermediate nodes.

In the inverse transformation, i.e. from CRM to PAINTING, respective attributes of the E12 instance "Object5034-production" would be distributed to the *painting* record, but the identity of the event itself would be lost. The idea of an ontology like the CRM is, to connect properties directly to the entity they are causally related with, and not to the description target (here a painting).

This is the basic mapping mechanism. In the transformation from structured records to CRM, first a set of unique entity identifiers is created. These allow later to merge the compiled knowledge, and to retrieve connections not visible in single documents. Eventually, the same event may be registered multiply, if not enough evidence for its identity is found. Heuristics may be run to deal with the merging of such cases. Then the entity instances are connected by property instances, and the equivalent CRM instance is ready. This in turn can be transformed into many other forms of records without further ambiguities.

The "has note" link can be used for any information about an entity not formally described by another entity or property in the CRM, as e.g. currently artist biography. It can also be used to capture all comments in the records as e.g. *creation_date_comment* . For fine-grain distinction of the semantic of such comments, the "has note" link has a type attribute.

Complete PAINTING-CRM Mapping:

title = E35 Title
painting -> title = E22 (E19) has title: Title

kind_of_title = E55 Type (Title Type)
painting-> kind_of_title = E22 (E19) has title - E35 (E1) has type: Type (JOIN)

painter corresponds to E21Person, a specialization of E39 Actor.
painting -> painter = E22 (E11) was produced by – E12 (E7) carried out by: Actor (JOIN)
painter-> artist_name can be used to create the unique identifier for the E39 instance. This would be an ideal convention to create E39 instances.
nationality = E55 Type (Person Type)
painter -> nationality = E21 (E1) has type: Type

contribution = E55 Type (Role Type)
painter-> contribution =E22 (E11) was produced by – E12 (E7) carried out by: (in the role of: Type)
This example of a link on a link, i. e. a role specifying the "carried out by" can be modeled in systems without this

feature by a relation entity for the “carried out by” link. In RDFS e.g. I would declare a class “carrying out” with property “in the role of”.

creation_date_begin = E61 Time Primitive, lower value
creation_date_end = E61 Time Primitive, upper value

painting->creation_date_begin /
painting->creation_date_end =
E22 (E11) was produced by – E12 (E2) has time-span - E52 (E1) at most within: Time Primitive (Double JOIN).

creation_date_comment= E62 String (Text attached to the time-span)
painting->creation_date_comment =
E22 (E11) was produced by – E12 (E2) has time-span - E52 (E1) has note: String (Double JOIN).
creation_place = E53 Place (TGN identifiers are good to create instances of E53)
painting->creation_place= E22 (E11) was produced by – E12 (E4) took place at: Place (JOIN).

For the following, the reader is kindly advised to use the respective documents about the CRM and the Dublin Core Element Set parallel to the reading of this text. We apologize for this inconvenience, but we could not find a suitable way to repeat the respective information of these documents in this text.

2. The Dublin Core Metadata Element set.

2.1 Methodological considerations

We present here a mapping of the DC Version 1.1: Reference Description (http://purl.org/DC/documents/rec-dces-19990702.htm) , taking into account qualifier proposals published on the DC Website until May 2000. Purpose of this mapping is to demonstrate the mechanism of data transfer from a DC record into a CRM representation preserving its semantics. It is further a demonstration how to provide fine-granularity semantics to DC records, to the degree DC definitions allow for a more definite interpretation. In the case such semantics can be defined, the CRM form can facilitate additional reasoning and merging aids with information in different formats.

Most DC elements are what linguists call “underspecified”, i.e. elements acquire a specific sense from the context, like the English word “begin”. The CRM tries to provide definitions with a unique sense, allowing for context-independent conclusions. As such, its philosophy is opposite to DC. Only well-defined notions are added, and the model is thought to be **extended** by users in a well-defined way, rather than leaving the user to **interpret** underspecified fields for his case.

This is not a critique on Dublin Core, but a clarification of different approaches for different tasks.

Each DC record corresponds to a resource being described. A resource in DC is defined as “anything that has identity”. This hardly being a philosophically well-founded definition, but more likely a pragmatic approach not to prematurely exclude anything that may turn out to be useful later, we take it as synonymous to anyone of the Dublin Core Types. The latter actually match with CRM notions.

We therefore distinguish different mappings according to different resource types. Our understanding of DC is, that elements are not only optional, but also for some types not applicable. Be e.g. a record about a person, let’s say my daughter, I would regard DC.source, DC.coverage, DC.subject, DC.publisher, as not applicable, and some people may regard DC.creator = "The Creator" for all persons. In all such cases, we have avoided to provide a mapping, not because of the inability of the CRM, but in expectance of a reasonable common interpretation of DC to come for such types. Cases where we feel CRM does not provide a mapping are explicitly marked.

2.2 Format conventions

In the following we refer to the resource described by a DC record as “DC”, and the restriction to a certain type as “DC [<type>]”, e.g. “DC [DCT1 party]”. We shall use either DCT1 types or CRM types, as more convenient.

By “type1 **OR** type2” we mean the type corresponding to the logical union of the sets of items correctly classified by type1 and type2.

For cases, where a CRM Entity is restricted to a certain value to achieve a mapping, we refer to the restricting value as above: <CRM Entity> [<value>], e.g. Type [Description].

3 Mapping

The Dublin Core metadata elements are:

- Title
- Creator
- Subject
- Description
- Publisher
- Contributor
- Date
- Type
- Format

- Identifier
- Source
- Language
- Relation
- Coverage
- Rights

The following chapters follow basically this sequence.

3.1 Mapping the described resource

Below we list for each DCT1 type either the equivalent CRM entity, or the lowest CRM superclass covering the respective DCT1 type:

DCT1 collection isA E22 Man-Made Object.
DCT1 dataset isA E73 Information Object

Following the examples, which all involve active parties, and from linguistic feeling, we conclude
DCT1 event = E7 Activity
In the sense of organized events (parties, conferences) rather than earthquakes, a bomb falling, a blossom opening.

From the definition follows rather DCT1 event = E4 Period, but this seems underspecified for the purpose.

DCT1 image = E38 Image **OR** E23 Iconographic Object

DCT1 interactive resource isA E73 Information Object
DCT1 model isA E29 Design or Procedure
DCT1 party = E39 Actor
DCT1 physical object = E19 Physical Object

We interprete the term "geographic area" in the sense of a resource as the material in this area, and not as the immaterial determination of a region independent of what changes it undergoes over time. Hence:
DCT1 place = E22 Site

Even if a service is some installed robot, we have argued in the CRM, that it follows the will and instruction of some people, and is not distinct from tools in general. One may argue, that in case of a robot no human contact takes place. Again, this may be regarded as analogous to sending letters. If the actual reasoning turns out to make a difference between people and their robots, this mapping should be extended.
DCT1 service isA E39 Actor

DCT1 software isA E73 Information Object
DCT1 sound isA E73 Information Object
DCT1 text isA E33 Linguistic Object

In the sequence, the resource a DC record describes is mapped as an instance of the respective CRM entity, e.g.:

If DC.Type = "sound", then
DC resource = E73 Information Object (of type “sound")

Or more formally:
DC [DCT1 sound] = E73 Information Object

The parenthesis "of type "sound"" means the instantiation of the property "has type" of the entity E28. And finally, if no DC.Type is given:

DC = E1 CIDOC Entity

3.2 DC.Title

Unconstraint:
DC.Title = E41 Appellation
DC -> DC.Title = E1 is identified by: Appellation
where E1 : CIDOC Entity.

In any case, an Appellation may be simultaneously a Title, can be translated as such, and have any other attribute of a Title. This holds for all Types, for which the CRM does not foresee an explicit property “has title”. It is unclear from the Dublin Core Documentation, if DC.Title actually has such a specific meaning. In the below, we interpret Title in the specific meaning for human creations in the narrower sense, and we interpret Title as Appellation in all other cases.

DC [E71 Man-Made Stuff]:
DC.Title = E35 Title
DC->DC.Title = E71 has title: Title

The following are the qualifiers of “DC Title Working Group Proposal for Title Qualifier”, Date: 2000-02-10”, applicable also to the general case above:

DC->DC.Title.Alternative = E71 has title: Title
(has type : Type["Alternative"])
In words, the property “has type” of the property “has title” in instantiated with the value “Alternative” etc.

DC.Title.LANG = E56 Language
DC->DC.Title.LANG = E71 has title – E35 (E33) has language : Language (JOIN)
where E33 Linguistic Object.

Translations of a title are better referred as such in the CRM, rather than as "alternative": E71 has title – E35 (E33) has translation : Linguistic Object

Actually Foundations, which do not fall under "Man-Made Stuff", tend to have a real, translatable title. However, the Agent Qualifier Working Group does not foresee a Qualifier "Title", rather a "Name", for the analogous case (see 3.3)!

3.3 Agents

In any case, Agent in the sense of DC is identical with Actor in the CRM:

DC.Creator = E39 Actor
DC.Contributor = E39 Actor
DC.Publisher = E39 Actor

The following are the qualifiers of "DC Agent Qualifiers, DC Working Draft – 10 December 1999", applicable to all Agents.

Agent Type = E55 Type (Actor Type)
Agent -> Type = E39 (E1) has type : Type

Agent Name = E41 Appellation
Agent -> Name = E39 (E1) is identified by: Appellation
where E1 : CIDOC Entity.

Note, that the CRM is a semantic net, not a data structure. Therefore the repetition of identical properties at different positions in the net is not only tolerable, but wanted, as it allows to identify certain semantics at different points, as here e.g., an Agent Name eventually appearing again as Title of a DC record about the same person. The specific nature of an Appellation being a name is encoded in the "has type" property of Appellation.

Agent Affiliation = E40 Legal Body
Agent -> Affiliation = E39 (E74) was member of : Group
where E74 Group, superclass of E40 Legal Body

Temporary roles of validity with respect to an event, action, or activity are dealt with in the CRM as property of the property "carried out by", connecting the Agent with an Activity:

Agent Role = E55 Type ("in the role of" – Type)
Agent -> Role = E7 carried out by : Actor
(in the role of : Type)

where E7 Activity.

Roles associated in a relevant way with the life of an Agent are regarded as Actor Type in the CRM.

Identifiers may appear as the value of the entity instance itself. In that case, they must be unique at least within the given CRM application. URI's are quite appropriate for that purpose. Then no mapping is needed. If identifiers of some DC application are not compatible with a respective CRM application, Identifier is a case of Appellation. In particular, the "has type" property of Appellation can be instantiated with "URI" etc.

Agent Identifier = E41 Appellation
Agent -> Identifier = E39 (E1) is identified by: Appellation
where E1 CIDOC Entity

Other qualifiers in use like "jurisdiction" can be mapped to the "has note" property of Actor, if no global reasoning about it is required. Else, specific mappings are needed, e.g. "member of " could relate an Agent to the social group representing the jurisdiction etc.

3.4 DC.Creator, DC.Contributor

The specific fields "creator", "contributor", "publisher" are in the CRM event-mediated. Contributors are not fundamentally distinct from creators. The "in the role of" property allows to make this and any finer distinction. Physical creations are regarded as distinct in their associations from conceptual ones. Those include material flow for physical creations, in contrast to broad and often vague intellectual influences in conceptual creations. Of course, physical creations can have conceptual aspects. Respectively, three cases are distinguished:

DC [E24 Physical Man-Made Stuff]:

Modification is the most general entity of physical intervention on an object. It was felt that "creation" is ambiguous in the many cases, where relevant pre-existing parts are used or re-arranged. Else, Modification is specialized into Production for essentially "new" items made of more or less "raw materials".

DC->DC.Creator = E24 (E11) was produced by – E11 (E7) carried out by : Actor
(in the role of : Type)
where E11 Modification, E7 Activity, (JOIN).

DC [E28 Conceptual Object]:

In contrast, conceptual creations do not leave physical traces of previous materials, only similar or identical parts.

DC->DC.Creator = E28 (E11) was created by – E65 (E7) carried out by : Actor

(in the role of : Type)
where E65 Conceptual Creation, E7 Activity, (JOIN).

DC [E74 Group]:

Groups are formed, if at all outcome of a willing act. This implies foundations, governments etc. The resulting group should be more regarded as success of the act than as a product, as no strict relation of its actions to the founder can be concluded. We do not regard physical persons as created, nor that any analogous semantics exist.

DC->DC.Creator = E74 (E66) was formed by – E66 (E7) carried out by : Actor
(in the role of : Type)
where E66 Formation, E7 Activity, (JOIN).

For all DCT1 Types not covered by the above cases, we assume that DC.Creator is either not applicable or still has to be reasonably interpreted.

The above holds identically for DC->DC.Contributor instead of DC->DC.Creator, where "in the role of: Type" either equals "Contributor" or a respective narrower term.

3.5 DC.Subject

DC.Subject = E55 Type (Concept Type)

DC [E24 Physical Man-Made Stuff]:

DC-> Subject = E24 depicts concept : Type

DC [E28 Conceptual Object]:

DC-> Subject = E28 refers to concept : Type

We feel that other resource types do not have a subject in the proper sense.

The following are the qualifiers of "DC Working Draft – 14 November 1999", about the consensus of the SubDesc Working Group (http://www.mailbase.ac.uk/lists/dc-subdesc/files/wd-subdesc-qual.htm) for Subject. The element qualifiers are Types of subject types (restriction of the Type-Type to the respective value or a narrower term):

DC.Subject.Descriptor = E55 Type (Type Type [Descriptor])
DC -> DC.Subject.Descriptor = E24 depicts concept : Type
where E55 (E1) *has type* : Type[Descriptor]

DC.Subject.Classification = E55 Type (Type Type [Classification])
DC -> DC.Subject.Descriptor = E24 depicts concept : Type
where E55 (E1) *has type* : Type[Classification]

DC.Subject.Keywords = E55 Type (Type Type [Keyword])
DC -> DC.Subject.Descriptor = E24 depicts concept : Type
where E55 (E1) *has type* : Type[Keyword]

Finally value qualifiers appear as type of E32 Authority Document:

E55 (E32) is part of : Authority Document

Besides subject characterization as concepts, (as appear in classification systems, library subject headings), the CRM foresees direct reference to instances of the real world, in particular as element of a related data structure.

3.6 DC.Description

As in DC, in CRM everything has a description:

DC.Description = E62 String
DC -> Description = E1 has note : String
(has type : Type)
where E1 CIDOC Entity, Type a "Note Type".

"Note Types", referred by the property "has type" of the "has note" property account for Qualifiers "Abstract, Note, TableOfContents, Audience, Release etc. (In RDFS one would develop the property "has note" into a subproperty for each "Note Type", like "has abstract" etc., in order to simulate properties of properties).

Language can be mapped as type, but better as part of the encoding format of the string itself.
The latter is out of the scope of the CRM.

3.7 DC.Publisher

Following [FRBR], only a "Manifestation" can be published. In the following, there are three interpretations for publishing:

1) The resource itself is a manifestation. Publishing is seen as the act of offering copies on the market, eventually registering an ISBN or something equivalent, or putting a copy on an Internet site for public access.

2) The resource is an expression/ work. Publishing is then seen as the various possible actions of creating published manifestations.
3) The resource is anything else. Then there is no publication possible. Rather, a document describing it has been created and published. Typically, one would register the **first publication**, e.g. in archeology, an object is regarded as "published" after the first publication of a description of it.

The following mapping uses the general properties of Activity. It may be beneficial, to create a specific "Publication" entity, given more specific properties are identified.

Case 1 (DC [E73 Information Object]):

DC->DC.Publisher = E73 (E7) was taken into account by– E7 carried out by : Actor
where E7 Activity, (JOIN), and E7 has type : Type [Publication]

Case 2 (DC [E28 Conceptual Object]):

DC->DC.Publisher = E28 (E7) was taken into account by –
E65 (E7) carried out by : Actor

where E65 Conceptual Creation, E7 Activity, (JOIN),
and E65 has type : Type [Publication Creation]

Of course, the creation of a manifestation may be modeled separate, if the knowledge is available in the DC record or somewhere else. Further, if a composite "publication creation" event is used, different roles of Actors can be distinguished by the "in the role of" property of the "carried out by" property.

Case 3 (DC [DCT1 physical object, party, event]):

DC->DC.Publisher = E1 is documented in - E31 was created by -
E65 (E7) carried out by : Actor
where E1 Document, E65 Conceptual Creation, E7 Activity, (JOIN),
and E65 has type : Type [Publication Creation]

This case is an interesting example for the utility of a more fine-grain language to mediate between different data structures. The above path over three properties may actually be composed from information found in more than one document.

3.7 DC.Date

The following uses the qualifiers of "DC Date Qualifiers, DC (Final) Working Draft – 14 December 1999". In any case:

DC.Date = E52 Time-Span
DC.Date.start = E61 Time Primitive, lower value
DC.Date.end = E61 Time Primitive, upper value

There are two interpretations for start/ end: Either uncertainty, an interval within which something has happened, or a duration over which a process lasted. The uncertainty interval is given by:

DC.Date->start /
DC.Date->end = E52 (E1) at most within: Time Primitive

And the duration interval is given by:

DC.Date->start /
DC.Date->end = E52 (E1) at least covering: Time Primitive

For a certain duration, both have to be given with identical values.

The relation of a Date to a resource is highly underspecified. Without a qualifier, the only reasonable conclusion is, that the coming into existence is a date before or equal to the given one, and the end of the existence of the resource is equal or after the date. We introduce a completely unspecified event:

DC [E70 Stuff]:

DC ->Date = E70 was present at - E5 (E2) has time-span : Time-Span
where E5 Event, E2 Temporal Entity, (JOIN)

For the qualifier “Created”, this splits into:

DC [E24 Physical Man-Made Stuff]:
DC ->Date.Created = E24 (E11) was produced by
– E12(E2) has time-span : Time-Span
where E12 Production, E2 Temporal Entity, (JOIN)

DC [E28 Conceptual Object]:
DC ->Date.Created = E28 (E11) was created by
– E65 (E2) has time-span : Time-Span
where E65 Conceptual Creation, E2 Temporal Entity, (JOIN).

Analogous to the Creator. Here the JOIN over the event has to use the same instance of the respective Production or Creation event introduced to map the Creator or Contributor. Curious enough, the Date qualifiers distinguish Creation from Modification, but not the Agent fields.

Analogous “Issued” relates to the “Publication” events

DC [E39 Actor]:

DC ->Date = E39 participated in : E5 (E2) has time-span : Time-Span
where E39 Actor, E5 Event, E2 Temporal Entity, (JOIN)

DC [E5 Event]:

DC ->Date = E5 (E2) has time-span : Time-Span
where E5 Event, E2 Temporal Entity

For the qualifier “Created”, we distinguish the cases:

DC [E24 Physical Man-Made Stuff]:
DC ->Date.Created = E24 (E11) was produced by
– E12(E2) has time-span : Time-Span
where E12 Production, E2 Temporal Entity, (JOIN)

DC [E28 Conceptual Object]:
DC ->Date.Created = E28 (E11) was created by
– E65 (E2) has time-span : Time-Span
where E65 Conceptual Creation, E2 Temporal Entity, (JOIN).

Analogous to the Creator. Here the JOIN over the event has to use the same instance of a Production or Creation event. Curious enough, the Date qualifiers distinguish Creation from Modification, but not the Agent fields.
This mapping can be continued to the forming of a group etc.

The “Modified” Qualifier maps to a modification event for Physical Man-Made Stuff. For Conceptual Object, we propose either Conceptual Creation with a suitable Type in the case of version creation. Else one may put it into a “has note” field, not being relevant for resource discovery. We leave these mappings to the reader. Alternatively, one could introduce a Modification event for Conceptual Objects as regular extension of Activity.

Analogous “Issued” relates to the “Publication” events referred above. “Available” can be mapped as an Activity of Type “Publication period” with respective begin/end. We leave these mappings to the reader.

"Valid" is mapped to “has note”, or can be modeled as a regular extension of Temporal Entity relating to E31 Document.

3.8 DC.Type

In any case:

DC.Type = E55 Type
DC -> DC.Type = E1 has type : Type
where E1 CIDOC Entity

In the sequence, the resource "DC" is mapped to the lowest CRM entity, that comprises the DC Type, as analyzed in 3.1.

The only qualifier proposed in "DC Type Working Group Proposal for Qualifier Usage Date: 1999-09-07" is "Note":

DC.Type.Note = E62 String
DC.Type->Note = E55 (E1) has note : String
where E55 Type

Being withdrawn later, “DC Type Qualifiers, Working Draft – 10 December 1999” proposes DCT1. This is mapped as a value (DCT1) for the type of Type:

DC.Type.DCT1 = E55 Type [DCT1] (Type Type)
DC.Type->DCT1 = E55 (E1) has type : Type [DCT1]
where E55 Type

3.9 DC.Format

The first obvious interpretation of DC.Format is a type (e.g. the proposed MIME types). As such:

DC.Type = E55 Type
DC -> DC.Type = E1 has type : Type
where E1 CIDOC Entity

The MIME types correspond however more specifically to types of E73 Information Object, which is included in the above.

The Qualifiers from “DC Format Working Group, Proposed Format Qualifiers, Date:1999-12-21” Medium and IMT can be mapped as type types:

DC.Type.Medium = E55 Type [Medium] (Type Type)
DC.Type->Medium = E55 (E1) *has type* : Type [Medium]
where E55 Type
DC.Type.IMT = E55 Type [IMT] (Type Type)
DC.Type->IMT = E55 (E1) *has type* : Type [IMT]
where E55 Type

The interpretation of “Extent” depends on the resource type.. The CRM foresees dimensions for physical entities:

DC [E18 Physical Entity]:

DC.Format.Extent = E54 Dimension

DC-> Format.Extent	= E18 has dimension : Dimension where E18 Physical Entity

Sizes of files etc. map to the "has note" property. Else, Dimension can be regularly extended to other resource types.

3.10 DC.Identifier

An identifier unique throughout the scope of the application would be used to create the node (entity instance) representing the resource (here "DC"). All further identifiers would be Appellations (see Title). From the experience of museums, it seems to be good practice to register all previous identifiers. The CRM foresees a specific property for Physical Objects, that are handed around through may be thousands of years:

DC [E19 Physical Object]:

DC.Identifier	= E42 Object Identifier
DC-> Identifier	= E19 is identified by : Object Identifier
	= E19 preferred identifier is : Object Identifier where E19 Physical Object

3.11 DC.Source

Source will be dealt with in “Relation”.

3.12 DC.Language

Language holds only for Linguistic Objects. Note that the CRM foresees multiple instantiation. Anything can be simultaneously a Linguistic Object, e.g. a Maya vase with Maya characters on it. For Language, only the Qualifier RFC 1766 has been proposed:

DC [E33 Linguistic Object]:

DC.Language	= E56 Language
DC-> Language	= E33 has language : Language
DC.Language.RFC1766	= E55 Type [RFC1766] (Language Code Type)
DC.Language->RFC1766	= E33 (E1) *has type* : Type [RFC1766] where E33 Linguistic Object

3.13 DC.Relation

The CRM in general tries to make hidden events explicit. Therefore a series of relations, in particular causal ones, should be expressed through events. We do not try to map an unqualified

relation to the CRM. Of course, the CRM could be trivially extended that way, if regarded necessary.

“HasPart”/"IsPartOf" maps to the different “is composed of” properties of Physical, Conceptual Objects, Place, Period, Group. The reason not to introduce one property in the CRM is that those objects cannot mix. A document cannot have a physical part.

DC [E19 Physical Object]:
DC->Relation.HasPart = E19 is composed of : Physical Object
where E19 Physical Object
DC->Relation.IsPartOf = E19 forms part of : Physical Object
where E19 Physical Object
DC [E28 Conceptual Object]:
DC->Relation.HasPart = E28 is composed of : Conceptual Object
where E28 Conceptual Object
DC->Relation.IsPartOf = E28 forms part of : Conceptual Object
where E28 Conceptual Object
DC [E4 Period]:
DC->Relation.HasPart = E4 is composed of : Period
where E4 Period
DC->Relation.IsPartOf = E4 forms part of : Period
where E4 Period
DC [E53 Place]:
DC->Relation.HasPart = E53 is composed of : Place
where E53 Place
DC->Relation.IsPartOf = E53 forms part of : Place
where E53 Place
DC [E74 Group]:
DC->Relation.HasPart = E74 had members : Actor
where E74 Group
DC [E39 Actor]:
DC->Relation.IsPartOf = E39 was member of : Group
where E39 Actor

"IsReferenceOf"/"HasReferencedBy" map to three different cases: The reference in the sense of a valid information about something (documentation), the intellectual relation made in some conceptual object, and the depictions on, or by the shape of physical objects.

DC [E1 CIDOC Entity]:
DC->Relation. HasReferencedBy = E1 (E31) is documented in : Document
where E1 CIDOC Entity, E31 Document
= E1 (E28) is referred to by : Conceptual Object
where E1 CIDOC Entity, E28 Conceptual Object

DC [E31 Document]:

DC->Relation. IsReferenceOf = E31 documents : CIDOC Entity
where E31 Document

DC [E28 Conceptual Object]:
DC->Relation. IsReferenceOf = E28 refers to : CIDOC Entity
where E28 Conceptual Object

DC [E24 Physical Man-Made Stuff]:
DC->Relation. IsReferenceOf = E24 depicts concept: Type
= E24 depicts object: Physical Stuff
= E24 depicts event: Event
where E28 Conceptual Object

DC [E55 Type]:
DC->Relation. HasReferencedBy = E55 is depicted by : Physical Man-Made Stuff
where E55 Type

DC [E18 Physical Stuff]:
DC->Relation. HasReferencedBy = E18 is depicted by : Physical Man-Made Stuff
where E18 Physical Entity

DC [E5 Event]:
DC->Relation. HasReferencedBy = E5 is depicted by : Physical Man-Made Stuff
where E5 Event

"IsVersionOf"/"HasVersion", "IsFormatOf"/"HasFormatand" "IsBasedOn"/"IsBasisFor" are event mediated. We see these relations as immediately tied to the creation of the respective resource. This provides the semantically correct hook for the dates associated. We interpret "IsVersionOf","HasFormat" as a special case of "IsBasedOn":

DC [E28 Conceptual Object]:
DC->Relation.IsBasedOn = E28 was created by -
E65 (E7) took into account : Conceptual Object
where E65 Conceptual Creation, E7 Activity (JOIN)
= E28 was created by -
E65 (E7) used object : Physical Object
where E65 Conceptual Creation, E7 Activity (JOIN)

DC->Relation.IsVersionOf = E28 was created by -
E65 (E7) took into account : Conceptual Object
where E65 Conceptual Creation, E7 Activity (JOIN)
and E65 (E1) has type : Type [Versioning]

DC->Relation.IsFormatOf = …….. has type : Type [Reformatting]

DC->Relation.IsBasisFor = E28 was taken into account by -
E65 (E7) has created : Conceptual Object
where E65 Conceptual Creation, E7 Activity (JOIN)

DC->Relation.HasVersion = E28 was taken into account by -

E65 (E7) has created: Conceptual Object
where E65 Conceptual Creation, E7 Activity (JOIN)
and E65 (E1) has type : Type [Versioning]
DC->Relation.HasFormat = …….. has type : Type [Reformatting]

We interpret "IsRequiredBy" as a more special case of "was intended for", where the type of the requiring resource matches the type of things the resource was made for.:

DC [E71 Man-Made Stuff]:

DC->Relation.IsRequiredBy = E22 was intended for -
E55 (E1) is type of : CIDOC Entity
where E22 Man-Made Object, E55 Type,
E1 CIDOC Entity Activity (JOIN)

Further semantics can be produced by adding a suitable attribute to the "E29 Design or Procedure", which allows to symmetrically represent the total of utensils needed for a specific function. If two items are "made for each other", they are part of a larger whole.

3.14 DC.Coverage

Coverage maps in general as:

DC [E28 Conceptual Object]:
DC->Relation. IsReferenceOf = E28 refers to: CIDOC Entity
where E28 Conceptual Object

3.15 DC.Rights

DC [E72 Legal Object]:

DC.Rights = E30 Right
DC -> DC.Rights = E72 is subject to : Right
= E72 right held by : Actor
where E72 Legal Object

May be the two links in the CRM are not ideal. A “rights” statement can be associated with the “has note” of the E30 Right entity.

4. Conclusions

The mapping of the Dublin Core Elements above, together with experience from mapping the AMICO data dictionary was the major motivation for the "Agios Pavlos Extensions", which have added to the CRM version 2.2 basically notions of Information Objects, and some "underspecified" properties. The latter are interpreted in the CRM as abstractions over specific "subproperties", and of course hooks for further extensions. In contrast to the DC, these properties were entered in the CRM to have a clear role in reasoning. E.g., physical participation in an event creates a fix point on the life-line of a person, which allows to deduce termini post-quem and antequem.

Of course, both, these mappings and the extensions to the CRM may undergo further consolidation. In particular we have not consulted any members of the DC development for more interpretation, so everything is based on our subjective impression. But this is not so important, because we wanted to demonstrate here the method and the power of the CRM approach, rather than to have reached a satisfactory end stage (we may never do). In this spirit, we have presented the mapping from the "impossible side", from the unspecific DC to the more precise CRM, which gives raise to the above presented series of case distinctions. Thereby we have shown, how the domain knowledge built-in in the CRM can actually recover details that are not formally entailed by DC.

The opposite mapping, which of the possible CRM paths map to which DC Elements, is more tedious due to the richness of CRM paths, on the other side it is semantically more trivial, as all paths are contracted to simple fields. [Chri99] presents a mechanism to completely formalize the viewing of a semantic net like a CRM implementation as a flat set of access points like the DC via Description Logics. The latter is not a data transformation mechanism, but allows even to query dynamically the underlying database.

We deliberately have left completely out issues of identifier generation and other implementation issues. Implementation and experimentation is the ultimate measure for the above theory, and we hope both soon to take place. Whereas implementation can be done by IT engineers, mappings as those presented here have a deeply interdisciplinary character, and we see a strong need more people to engage in such work, in order to bring forward the information society. We intend to continue this work with mapping a series of related standards and formats to the CRM, to create a broad conceptual base for data interchange in the cultural area and museum world.

5. References

[Anal96] A. Analyti and P. Constantopoulos: Specialization by Restriction and Schema Derivations. Technical Report FORTH-ICS/TR-176, October 1996.

[Anal98] A. Analyti, N. Spyratos, P. Constantopoulos and M. Doerr, "Inheritance under Participation Constraints and Disjointness" Proc. of the 8th European-Japanese Conference on Information Modeling and Knowledge Bases, pp. 269-287, 1998. Also, to be published in the book: H. Jaakkola, H. Kangassalo, E. Kawaguchi (eds.), Information Modeling and Knowledge Bases X, 1999

[Chri99] Y. Velegrakis, V. Christophides and P. Constantopoulos "Declarative Specification of Z39.50 Wrappers", In Proc. of the ECDL'99, pages 381-402, Paris, France, September 1999.

[CRM] "Definition of the CIDOC object-oriented Conceptual Reference Model", produced by the ICOM/CIDOC Documentation Standards Group, editors: Nick Crofts, Ifigenia Dionissiadou, Martin Doerr, Matthew Stiff, September 1999, See locations:
http://www.ics.forth.gr/proj/isst/Activities/CIS/cidoc/docs/crm_definition_15_6_00.rtf
http://www.geneva-city.ch:80/musinfo/cidoc/oomodel/CRMdefinition_040999.rtf
http://www.geneva-city.ch:80/musinfo/cidoc/oomodel

[Doer99] Martin Doerr and Nicholas Crofts "Electronic Esperanto: The Role of the Object Oriented CIDOC Reference Model", Proc. of the ICHIM'99, Washington, DC, September 22-26, 1999.

[EAD] "EAD Tag Library for Version 1.0, Encoded Archival Description (EAD)Document Type Definition (DTD)", Version 1.0, Technical Document No. 2, June 1998. Published by the Society of American Archivists and theLibrary of Congress (http://lcweb.loc.gov/ead/tglib/tlhome.html)

[FRBR] "Functional Requirements for Bibliographic Records", Final Report,UBCIM Publications - New Series Vol 19, by: IFLA Study Group on the Functional Requirements for Bibliographic Records, K . G. Saur München 1998.
(http://www.ifla.org/VII/s13/frbr/frbr.htm)

[Guar98] Guarino N. Formal Ontology and Information Systems. In N. Guarino (ed.), "Formal Ontology in Information Systems". Proc. of the 1st International Conference, Trento, Italy, 6-8 June 1998. IOS Press

[Mylo90] J.Mylopoulos, A.Borgida, M.Jarke, M.Koubarakis, "Telos: Representing Knowledge about Information Systems", ACM Transactions on Information Systems, Vol.8, No.4, October 1990, pp.325-362.

[RDFS] "Resource Description Framework (RDF) Schema Specification 1.0", W3C Candidate Recommendation 27 March 2000, editors: Dan Brickley, R.V. Guha, by W3C., http://www.w3.org/TR/2000/CR-rdf-schema-20000327

[SPEC] "SPECTRUM, The UK Museum Documentation Standard", compiled and edited by Alice Grant, Museum Documentation Association, Cambridge 1994.