

Typed properties and negative typed properties: Dealing with type observations and negative statements in the CIDOC CRM

Athanasios Velios^{a,*}, Carlo Meghini^b, Martin Doerr^c and Stephen Stead^d

^a *Ligatus, University of the Arts London, Chelsea College of Arts, 16 John Islip Street, London, SW1P 4JU, United Kingdom*

^b *Networked Multimedia Information Systems Laboratory, Institute for Information Science and Technologies “Alessandro Faedo” – ISTI, Italian National Research Council (CNR), Italy*

^c *Information Systems Laboratory, Institute of Computer Science, Foundation of Research and Technology Hellas, Greece*

^d *Paveprime Ltd., United Kingdom*

Editors: Mehwish Alam, FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany; Victor de Boer, Vrije Universiteit Amsterdam, The Netherlands; Enrico Daga, The Open University, United Kingdom; Marieke van Erp, KNAW Humanities Cluster, The Netherlands; Eero Hyvönen, University of Helsinki, Aalto University, Finland; Albert Meroño Peñuela, Vrije Universiteit Amsterdam, The Netherlands; Harald Sack, FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany

Solicited reviews: Enrico Daga, The Open University, United Kingdom; Luigi Asprino, University of Bologna, Italy; Stefano Borgo, Laboratory for Applied Ontology, Italy; one anonymous reviewer

Abstract. A typical case of producing records within the domain of conservation of cultural heritage is considered. During condition and collection surveys in memory organisations, surveyors observe types of multiple components of an object but without creating a record for each one. They also observe the absence of components. Such observations are significant to researchers and are documented in registration forms but they are not easy to implement using popular ontologies, such as the CIDOC CRM which primarily consider individuals. In this paper techniques for expressing such observations within the context of the CIDOC CRM in both OWL and RDFS are explored. OWL cardinality restrictions are considered and new special properties deriving from the CIDOC CRM are proposed, namely ‘typed properties’ and ‘negative typed properties’ which allow stating the types of multiple individuals and the absence of individuals. The nature of these properties is then explored in relation to their correspondence to longer property paths, their hierarchical arrangement and relevance to thesauri. An example from bookbinding history is used alongside a demonstration of the proposed solution with a dataset from the library collection of the Saint Catherine Monastery in Sinai, Egypt.

Keywords: Numerous individuals, non-existence, CIDOC CRM, bookbinding

1. Introduction

The problem addressed in this paper is observed often in research. It will be introduced within the cultural heritage domain using a representative example from bookbinding history and the Conceptual Reference Model (CIDOC

* Corresponding author. E-mail: a.velios@arts.ac.uk.

CRM) [9]. The CIDOC CRM is a popular standard for modelling records in memory organisations such as museums, libraries, galleries and archives. It can be used to model bookbinding records of observations of material evidence on books used for answering research questions in the field of bookbinding history [27].

Datasets integrated with the CIDOC CRM can be jointly queried in a knowledge base through the property relations offered by the CIDOC CRM. It is often the case that observations recorded in knowledge bases are contradictory, for example, as different understandings or views of a phenomenon. Such contradictions are interesting cases for further research, so identifying their corresponding statements in the knowledge base is important. In the field of bookbinding history these have a significant impact on decision making for professionals in memory organisations, primarily for the conservators treating and repairing books but also curators and scholars studying them for understanding the history of the dissemination of copies of the texts and their reception.

The maturity and robustness of the CIDOC CRM is demonstrated, among others, by the apparatus that describes the materiality of objects; this apparatus has been critically reviewed for many years. For example, a frequently used property of the CIDOC CRM is ‘P46 is composed of (forms part of)’, which can be used to express the link between a physical thing (e.g. a book) belonging to the CIDOC CRM class ‘E18 Physical Thing’ and its component part belonging to the same class. For example, a leaf marker is a small piece of material attached to a leaf for marking an important part of the text such as the beginning of a chapter.

An important construct within the CIDOC CRM is the use of the property ‘P2 has type’ with domain ‘E1 CRM Entity’ (i.e. the property is inherited by all CIDOC CRM classes) and range ‘E55 Type’. The property can be used to classify individuals based on terminological systems such as thesauri. The use of this property allows the CIDOC CRM to remain an ontology primarily of generic properties while being able to accommodate the granularity offered by domain experts through thesauri (see [9], on Minimality). ‘E55 Type’ can be used to model concepts in thesauri and the property ‘P2 has type’ can be considered as a way to extend the CIDOC CRM by using classes from thesauri as if they were individuals.

These properties are used regularly in descriptions of material aspects of books, including components making up a book structure. Books are complex objects which may include several hundred components and many observations can be recorded for each one of them. However, often due to limited resources it is not possible for all components to be recorded in a knowledge base. A survey of the manuscripts in the Saint Catherine Monastery in Sinai, Egypt [21], which was generally accepted as very detailed, was limited to observations which were needed for answering specific research questions. For example, leaf markers are recorded on page 1 of the Saint Catherine data registration form [20]. A book may have many leaf markers but recording each one of them is logistically not possible. In the Saint Catherine data form, only the materials and types of leaf markers are recorded without reference to individual ones. Therefore, there is a record of the type ‘leaf markers’ [11], but there is no record for any individual leaf marker.

Another important set of questions expressed in data registration forms is that of existence. For example, it is important to know if there are no leaf markers attached to a book. This information is important when planning conservation work or when studying leaf markers. If a book does not have any leaf markers, then there exists no individual to be described. Therefore the absence of individuals of a type (leaf marker type) needs to be recorded. This raises questions about the capacity of observation of material evidence and the set of real world constraints that are necessary to establish certainty of lack of existence. While the main focus of this work is how it is best to express lack of existence in a knowledge base, some discussion on the nature of observing non-existence is included in Section 4.3.

To summarise: the problems explored here are: a) how to record the type of things when these are too numerous to be documented individually (e.g. for a book with too many leaf markers) and b) how to produce records of things of a type for which individuals do not exist (e.g. for a book without leaf markers).

Solutions to these problems in both RDF Schema (RDFS) and OWL 2 DL (OWL) are discussed in this paper. OWL offers additional expressiveness in comparison to RDFS, but an RDFS solution is considered valuable because RDFS is popular in the CIDOC CRM community, especially for Linked Data implementations. It is noted that neither RDFS nor OWL can express the full semantics of the CIDOC CRM, in particular with respect to the concept of shortcuts as described by Meghini and Doerr [16] (p. 138), but here the aim is to provide a practical solution in either case.

1.1. Structure of the paper

Following the introduction, Section 2 presents the formalisation of the problem including a set of competency questions which are used to evaluate possible solutions. Section 3 presents related work which includes possible solutions and their evaluation based on these questions which shows that they have limitations. Section 4 provides recommended solutions including an analysis of the implications in the context of the CIDOC CRM and documentation practice. The recommended solutions are evaluated based on the same competency questions and this evaluation shows that the recommended solutions can answer the competency questions successfully. Section 5 presents an implementation of the proposed solutions for a sample dataset on book history from the library of the St. Catherine Monastery in Sinai. Section 6 summarises the conclusions and Section 7 points to future work.

2. Formalisation

When discussing solutions in OWL in this paper, the OWL 2 DL language in the functional notation is adopted alongside the notion of ontology as defined in that language.

The RDF language and the turtle notation are adopted when discussing solutions in RDFS. In both cases CIDOC CRM classes and properties are used, abbreviated by their identifiers. For example, ‘E55’ as opposed to ‘E55 Type’.

International Resource Identifiers (IRIs) standing for individuals are written as XML Q-names. For example:

```
ObjectPropertyAssertion( P46
  :book1 :leafMarker1 )
:book1 P46 :leafMarker1
```

The instances of class ‘E55 Type’ are also referred to as ‘types’ and object property assertions involving the property ‘P2 has type’ are referred to as ‘type assertions’. For example:

```
ClassAssertion( E55 :leafMarkerType )
ObjectPropertyAssertion( P2 :leafMarker1
  :leafMarkerType )
:leafMarker1 P2 :leafMarkerType
```

For each problem a set of ‘selection’ competency questions [24] is defined, i.e. those which return individuals as results. These are used to help with assessing possible solutions. These questions are also articulated in a simplified SPARQL notation for clarity.

2.1. Problem 1: Numerous things

In the first problem, an ontology O is considered containing several large sets of assertions of the form:

$$A(:s, :t) = \{ \text{ObjectPropertyAssertion}(\Pi :s :i), \text{ObjectPropertyAssertion}(P2 :i :t) \mid :i \in I(:s, :t) \}$$

where $:s$ and $:t$ are specific individuals, I is a large set of individuals and Π is any CIDOC CRM property different from $P2$. For example, $:s$ is the book $:book1$, $:t$ is the type $:leafMarkerType$, Π is the CIDOC CRM property ‘P46 is composed of’ and I contains all leaf markers on $:book1$.

The problem is how to compress O into O' so that the above assertion sets A are not included in O' , but it is still possible to extract from O' significant knowledge about individuals $:s$ and $:t$. For example, there is no need to

explicitly mention all individual leaf markers on a specific `:book1`, but it is useful to know that `:book1` does have components of type `:leafMarkerType`. `O` and `O'` are also referred to as the 'original' and 'compressed' graphs respectively.

The competency questions are selected based on the following criteria to ensure relevance:

- a) Questions involving `I` are not relevant.
- b) Questions involving `:s` and `:t` are relevant.
- c) Questions involving Π are relevant where Π is considered as a constant.

The first kind of assertion in `O` (`:s Π :i`) is considered for a given `:s`. This triple can be queried in SPARQL by triple patterns combining each member of the set $\{ :s, ?s, \exists s \}$ with each member of the set $\{ :i, ?i, \exists i \}$, with the convention that terms `?s`, `?i` are variables occurring in the result, whereas terms `$\exists s$` and `$\exists i$` are variables not occurring in the result:

- a) (`:s Π :i`): Is `:s` connected to `:i` through Π ? Or, does `:i` connect to `:s` through Π ?
- b) (`:s Π ?i`): Which individuals connect to `:s` through Π ?
- c) (`:s Π $\exists i$`): Is `:s` connected to any individual through Π ?
- d) (`?s Π :i`): Which individuals are connected to `:i` through Π ?
- e) (`?s Π ?i`): Which pairs of individuals are connected through Π ?
- f) (`?s Π $\exists i$`): Which individuals are connected to any individual through Π ?
- g) (`$\exists s$ Π :i`): Is there any individual connected to `:i` through Π ?
- h) (`$\exists s$ Π ?i`): Which individuals connect to any individual through Π ?
- i) (`$\exists s$ Π $\exists i$`): Is there any individual connected to any individual through Π ?

In addition, the classes to which `:s` and `:i` belong can be queried:

- j) (`:s a ?class`): To which classes does `:s` belong?
- k) (`:i a ?class`): To which classes does `:i` belong?

From the above list, questions a), b), d), e), g), h), k) are not relevant as they ask for or mention individual members of `I`. Questions c) and i) can be answered by question f), which, alongside j), are the only significant questions for the first kind of assertion.

If the exercise is repeated for the second kind of assertion (type assertion), (`:i $P2$:t`) as well as for the combination of the two (`:s Π :i`) (`:i $P2$:t`) it can be shown that the significant competency questions are:

Q1: (`?s Π $\exists i$`): Which individuals are connected to any individual through Π ? E.g. which books have components?

Q2: (`:s a ?class`): To which class does `:s` belong? E.g. is this item in the class 'E18 Physical Thing' (i.e. a book, whether or not it is not elaborated further in `O`)?

Q3: (`$\exists i$ $P2$?t`): Which individuals connect to any individual through $P2$? E.g. which individuals are types of some individual?

Q4: (`:t a ?class`): To which class does `:t` belong? E.g. is this item in the class 'E55 Type' (i.e. leaf marker type, whether or not it contributes to the description of a book in `O`)?

Q5: (`?s Π $\exists i$`) (`$\exists i$ $P2$?t`): Which individuals are connected through Π to some individual of which type? E.g. which books have components and what types are they?

Q5 can also be posed for individual `:t` and for individual `:s`:

- (`?s Π $\exists i$`) (`$\exists i$ $P2$:t`): Which individuals are connected through Π to some individual of type `:t`. E.g. which books have leaf marker components?
- (`:s Π $\exists i$`) (`$\exists i$ $P2$?t`): Which individuals are types of some individual that connects to `:s` through Π ? E.g. to which types do the components of book `:s` belong?

The answers to both these queries can be obtained from the answer to Q5, so they do not need to be considered separately.

Q1 and Q5 may also be stated over super-properties of Π which will be considered in Section 4. Q3 and Q5 might also be stated over super-properties of $P2$, but $P2$ does not have any super-properties in the CIDOC CRM, so this case is not considered.

It is also noted that questions Q1–Q5 may also occur as part of larger ones. The substitution strategy considered in Section 4 can be applied directly to Q1–Q5 and also to larger questions.

Therefore the problem considered here is replacing $A(:s, :t)$ with a set of assertions $A'(:s, :t)$ so that: a) A' is significantly smaller than A , and b) A' allows answering the competency questions Q1–Q5 where Π can also be any super-property of Π .

A similar problem has been described previously as ‘MISO-R’, ‘multiple indirectly specified objects through a relationship’ [26]: “There is a distinguished object that is related, by the same kind of relationship, to (possibly even one, but usually) multiple undistinguished objects of a certain type.” The issue addressed in MISO-R is how to express the multiplicity of these undistinguished objects beyond the statements of existence. Problem 1 is a special case of MISO-R for the CIDOC CRM property ‘P2 has type’, considering at least one undistinguished object but without aiming to quantify them.

2.2. Problem 2: Non-existing things

In the second problem, an ontology ON is considered containing a set of assertions of the form:

$$N(:s, :t) = \{ \text{NegativeObjectPropertyAssertion}(\Pi :s :i), \text{ObjectPropertyAssertion}(P2 :i :t) \mid :i \in I(:s, :t) \}$$

where: $:s$, Π , $P2$, I and $:t$ are defined as before. The first assertion `NegativeObjectPropertyAssertion($\Pi :s :i$)` is also called a ‘grounded negative statement’ [3]. ON needs to be compressed to ON’ so that all assertions in N are not included. For example, the fact that `:book1` does not have any of the leaf markers in the domain that ON represents needs to be asserted.

The second problem is often solved in databases using the Closed World Assumption (CWA) but given that the formulation of the problem is done using OWL and the CIDOC CRM, both of which adhere to the Open World Assumption (OWA), CWA databases are not considered here. However, in Section 4.3.2 it is shown how a reduced scope of the OWA can be used to reason about non-existing things.

The competency questions for problem 2 have the same structure as those for problem 1, but they are posed with negative polarity. For example, Q1’ articulated as: “Which individuals are not connected to an individual through Π ?” Q1’ is impossible to answer in OWA systems unless there is knowledge that allows to deduce that an individual satisfies the question condition. E.g. it is impossible to know whether a book has no components since assertions about non-existing components are made for specific types while components of other types may exist. Q2’ and Q4’ are not considered because knowing all classes that individuals in $:s$ and $:t$ do not belong is not relevant knowledge. Q3’ is also not considered because knowing all types which are not included is also not relevant knowledge.

Q5’: $\neg[(?s \Pi \exists i) (\exists i P2 :t)]$ which individuals are not connected through Π to any individual or, if they are, the individual is not of type $:t$? E.g. which books do not have any components or they only have components that are not leaf markers? The last part of the question is relevant but the first part is not in this case as it is similar to Q1’. This articulation of Q5 is more appropriate to the articulation with $?t$ as the specific type $:t$ forms part of the context of complete observation as explained in Section 4.3.2. This question also applies to sub-properties of Π and $P2$.

The domain expertise of the maintainer of ON is important for identifying the relevant negative statements to include. In considering negative statements with property Π , only knowledge that involves individuals in the domain of Π are relevant. The rest may be logically true but not relevant. For example, the domain of property ‘P46 is composed of’ is ‘E18 Physical Thing’. Only statements that negate ‘P46 is composed of’ for physical things are

relevant because they bring significant knowledge. Any statements that negate ‘P46 is composed of’ for individuals which do not belong to ‘E18 Physical Thing’, do not contribute to knowledge.

Problem 1 and problem 2 are related as the existence of one implies the existence of the other.

The competency questions for each problem will be used as the basis for evaluating existing solutions in Section 3 and also the recommended solutions in Section 4. As it will be shown none of the solutions in Section 3 can correctly answer the competency questions while the proposed solutions in Section 4 can.

3. Existing solutions

An obvious solution is to declare two disjoint classes: a) one for individuals for which the connection to a type through Π and $P2$ applies (e.g. book with leaf markers), and b) another for the individuals for which the connection to a type through Π and $P2$ does not apply (e.g. books without leaf markers). Membership of each class indicates whether the characteristic exists and membership of both indicates an observation contradiction. This solution is impractical because: a) it requires the definition of two classes for every type listed in a thesaurus, which often contains thousands of types or many more, posing questions around maintenance and b) there is no way to query for property Π or for type: t which are part of the competency questions.

The existing solutions evaluated in the following sections are summarised in Table 1.

Table 1
Summary of existing solutions based on related work

solution	short description	comments
Counting quantifiers (OWL, RDFS)	Pt property combining semantics of Π and $:t$	cannot query for Π or $:t$, Q1, Q3, Q5, Q5' cannot be answered
Existential restrictions (OWL)	ObjectSomeValuesFrom	equivalent to recommended solution in 4.1 but less readable
Existential restrictions (RDFS)	blank nodes	cannot express multiplicity of individuals
Placeholder individuals (OWL, RDFS)	individual $:I$ corresponds to many real individuals	cannot express multiplicity of individuals, Q5' cannot be answered
Shortcut properties (OWL, RDFS)	SP property connecting $:s$ and $:t$	cannot query for Π or $:t$, Q1, Q5, Q5' cannot be answered
Linguistic annotations (OWL, RDFS)	linguistic annotation to individual $:I$ to indicate multitude	semantics of annotations may be unknown
Reification (OWL, RDFS)	new class TypedStatement with properties to connect to $:s$, Π , $P2$ and $:t$	difficult to query for individuals in reified (range) and non-reified statements (domain)
Negated properties (RDFS)	properties negate by convention	limited reasoning
Property chains (OWL)	ObjectPropertyChain	Q5' cannot be answered

3.1. Counting quantifiers

Mirza et al. [18] describe a method for automatically introducing ‘counting quantifiers’ in a knowledge base with examples from wikipedia. Counting quantifiers are statements about the number of types of statements:

```
DataPropertyAssertion( Pt :s n )
```

where Pt is a property which combines the semantics of Π and $:t$, and n is the number of statements applicable between $:s$ and $:t$ via Π . For the example with leaf markers, the statement:

```
DataPropertyAssertion(
  hasNumberOfLeafMarkers :book1 5 )
```

would mean that `:book1` has 5 leaf markers. Although the exact number of individuals is not of concern in problem 1, this solution can be used in problem 2 to describe individuals with n equal to 0, e.g. for books without leaf markers. This solution fails to answer the competency questions as one cannot query for Π or $:t$ and it would be impractical to implement as the new properties depend on thesaurus types.

3.2. Existential restrictions

OWL existential restrictions can be applied to indicate that at least one individual exists, that connects to $:s$ through Π and is of type $:t$:

```
ClassAssertion( ObjectSomeValuesFrom(
  Π ObjectHasValue( P2 :t ) ) :s )
```

This allows answering questions Q1 to Q5. In order to answer Q5', it would be possible to use the complement of the class expression contained in the above assertion:

```
ObjectComplementOf( ObjectSomeValuesFrom
  ( Π ObjectHasValue( P2 :t ) ) )
```

However, the last class expression would also denote all individuals not connected to any type apart from $:t$ and while this is logically correct, it does not constitute significant knowledge (also see Section 4.3.2).

Alternatively it is possible to use the axiom:

```
SubClassOf( ObjectSomeValuesFrom( Π
  ObjectHasValue( P2 :t ) )
  owl:Nothing )
```

which asserts that there are no individuals that connect to $:s$ through Π and are of type $:t$. This can answer Q5' within the given context of $:t$. Another formulation of this solution in a more readable way is presented in Section 4.1.

Existential restrictions are sometimes implemented in RDFS using blank nodes, i.e. to indicate that one unknown individual exists. This reduces the capacity of the solution to express the possible (and likely) existence of many individuals. Using multiple blank nodes implies that a fixed number of individuals exist which, in the case considered here, is unknown and cannot be specified. Other limitations of blank nodes have been reported (e.g. [15] and [8]) regarding the inconsistency of software implementations processing blank nodes and the lack of understanding of blank nodes by people who work within a Linked Data context.

3.3. Placeholder individuals

One way of solving problem 1 is by defining one individual which represents all of the numerous things that need to be described. Svátek et al. [26] call these individuals 'some instances placeholder individuals'.

```
ObjectPropertyAssertion( Π :s :I )
ObjectPropertyAssertion( P2 :I :t )
```

where $:I$ is one individual representing all members of I . For example:

```
ObjectPropertyAssertion( P46 :book1
    :book1LeafMarkers )
ObjectPropertyAssertion( P2
    :book1LeafMarkers :leafMarkerType )
```

Semantically, this would be consistent if the identity criterion for $:I$, which belongs to class ‘E18 Physical Thing’, could be set as the group of all leaf markers on a book. This allows answering questions Q1 to Q5 but there is no simple way of indicating the multiplicity of this single individual. Problem 2 cannot be resolved by declaring a placeholder individual and question Q5’ cannot be answered.

3.4. Shortcut properties

Another potential solution is defining a new property SP connecting individual $:s$ and type $:t$ directly:

```
ObjectPropertyAssertion( SP :s :t )
NegativeObjectPropertyAssertion( SP
    :s :t )
```

For example a contradiction would be created when asserting the following for $:book1$:

```
ObjectPropertyAssertion( SP :book1
    :leafMarkerType )
NegativeObjectPropertyAssertion( SP
    :book1 :leafMarkerType )
```

Shortcut properties do not answer the competency questions of problems 1 and 2 since they do not refer to Π .

3.5. Linguistic annotations

Svátek et al. [26] consider existential restrictions and placeholder individuals as possible solutions for problem 1 and flag the problem of one individual potentially representing multiple individuals. To resolve this problem they propose a solution based on linguistic annotations of restrictions. They make an argument for wider adoption of annotations as part of computing processes but this relies on naming conventions which may be difficult to use for reasoning as the semantics of the labels used may not be known.

3.6. Reification

Another potential solution is declaring a new class TypedStatement to accommodate the following reification construct:

```
ClassAssertion ( TypedStatement :ts )
ObjectPropertyAssertion(
    typedIndividual :ts :s )
ObjectPropertyAssertion(
    typedProperty :ts  $\Pi$  )
ObjectPropertyAssertion(
    typedProperty :ts P2 )
ObjectPropertyAssertion(
    typedType :ts :t )
DataPropertyAssertion(
    typedNegative :ts xsd:boolean )
```

For example:

```
ClassAssertion( TypedStatement
  :statement1 )
ObjectPropertyAssertion(
  typedIndividual :statement1 :book1 )
ObjectPropertyAssertion(
  typedProperty :statement1 P46 )
ObjectPropertyAssertion(
  typedProperty :statement1 P2 )
ObjectPropertyAssertion(
  typedType :statement1
  :leafMarkerType )
DataPropertyAssertion(
  typedNegative :ts false )
```

which states that `:book1` has components of type `:leafMarkerType`.

This is a viable solution for both problems and the competency questions could be encoded and answered based on the reification structure, in particular in RDFS. A solution based on class reification is not recommended because when querying the knowledge base, explanations are required to define: a) which part of it contains direct statements with individuals as subjects and b) which part of it contains reified statements with individuals as objects. Reification methods based on properties which eliminate this problem are possible [7] and this is explored further in Section 4.2.

3.7. Negated properties

Negated properties are properties whose semantics are understood as negation. These are often used in wiki-data within an RDFS context. For example the property `P9660` (<https://www.wikidata.org/wiki/Property:P9660>) specifies resources which are not described in a relevant wikipedia page whereas property `P1343` (<https://www.wikidata.org/wiki/Property:P1343>) specifies resources described in a page, indicating a contradiction when used together based on semantics. While this solution can work for direct knowledge it does not provide any rules to assist with reasoning, but it is worth noting it as frequent practice.

3.8. Property chains

OWL property chains can be used to specify the path from `:s` to `:t` through properties Π , $P2$. For a property Π a property ωp is declared as:

```
SubObjectPropertyOf(
  ObjectPropertyChain(  $\Pi$  P2 ) : $\omega p$  )
```

This can then be used to assert statements:

```
ObjectPropertyAssertion( : $\omega p$  :s :t )
NegativeObjectPropertyAssertion( : $\omega p$  :s :t)
```

Asserting both for the same individual will identify a contradiction. For this solution to work it is necessary to specify manually the ωp of each CIDOC CRM property to query. The problem with this solution is that the negative property assertion negates the whole chain Π , $P2$ and therefore it is impossible to know if the negation applies to Π or $P2$. However, this solution can answer the competency questions Q1–Q5. It can also answer question Q5' if it is assumed that the negation applies to Π and not $P2$, for example, when the observation can only be done on Π .

4. Cardinality and typed properties

Two recommended solutions, one for OWL and one for RDFS are presented next.

4.1. OWL cardinality restrictions

OWL cardinality restrictions can be used to define unnamed classes based on cardinality of properties. The axioms defining these classes are called ‘Object Property Cardinality Restrictions’ in OWL 2 DL [19]. For problem 1 the range cardinality of Π is at least 1 and for problem 2 the range cardinality of Π is at most 0:

```
ClassAssertion( ObjectMinCardinality(
  1  $\Pi$  ObjectHasValue( P2 :t ) ) :s )
ClassAssertion( ObjectMaxCardinality(
  0  $\Pi$  ObjectHasValue( P2 :t ) ) :s )
```

For example, the following statements indicate a contradiction in the knowledge base:

```
ClassAssertion( ObjectMinCardinality(
  1 P46 ObjectHasValue( P2
    :leafMarkerType ) )
  :book1 )
ClassAssertion( ObjectMaxCardinality(
  0 P46 ObjectHasValue( P2
    :leafMarkerType ) )
  :book1 )
```

OWL cardinality restrictions are preferred to the solution discussed in Section 3.2 because they allow explicit articulation of the intended statements making them more readable.

In Section 5 it is shown that this solution allows answering the competency questions for both problems and a proof is exemplified for a test dataset.

4.2. RDFS typed properties

The solution in Section 4.1 is specific to OWL and this paper aims to offer a solution also in RDFS where cardinality restrictions for properties cannot be used.

The main limitation of a shortcut property SP , as described in Section 3.4, is the lack of capacity to query for Π since there is no way to connect the shortcut property SP with the properties that it stands for: Π , $P2$. Shortcut properties cannot be used for negative statements unless negation is embedded in their semantics in a fashion similar to what is described in Section 3.7. These limitations are overcome here by creating two new kinds of properties and providing axioms for them. The new kinds of properties are: a) typed properties (TP) and b) negative typed properties (NTP), which are both accompanied by additional reification statements capturing their semantics. For typed properties:

```
TP a rdf:property ;
  H1  $\Pi$  ;
  H2 P2 ;
  Hn false .
TP rdfs:domain  $\Pi$ d .
TP rdfs:range E55 .
:s TP :t
```

E55 is the range of the TP property given that it is the range of ‘P2 has type’. Πd is the domain of Π and also the domain of TP. Similarly, for negative typed properties:

```
NTP a rdf:property ;
    H1  $\Pi$  ;
    H2 P2 ;
    Hn true .
NTP rdfs:domain  $\Pi d$  .
NTP rdfs:range E55 .
:s NTP :t
```

For example, for multiple existing leaf markers the property is TP46:

```
TP46 a rdf:property ;
    H1 P46 ;
    H2 P2 ;
    Hn false .
:book1 TP46 :leafMarkerType
```

and for non-existence of leaf markers the property is NTP46:

```
NTP46 a rdf:property ;
    H1 P46 ;
    H2 P2 ;
    Hn true .
:book1 NTP46 :leafMarkerType
```

The reification statements apply to the property and therefore this solution is free from the class reification problems mentioned in Section 3.6, i.e. the individuals `:s` and `:t` are in the expected domain and range respectively. This reification allows connecting the new properties with the original CIDOC CRM properties for querying. Inconsistencies in observation recorded in the knowledge base do not depend on the interpretation of property semantics but can be done automatically based on the rules included in the reified statements.

Intuitively, a TP property stands for the composition (chain) of properties Π and P2. The newly introduced properties H1 and H2 connect the new property to its component properties, that is the CIDOC CRM properties of the first and second step of the chain respectively. Property Hn is `false` for TP properties and `true` for NTP properties.

Syntactically, the identifiers of the new properties can be produced automatically by inserting ‘T’ (typed) and ‘NT’ (negative typed) in front of the CIDOC CRM property identifier. The labels require human processing to ensure readability and generally fall into this pattern:

TP: “[CIDOC CRM property label] of type”

NTP: “[negation (e.g. “is not” or “does not”)] [CIDOC CRM property label] of type”.

The additional statements required to correctly answer the competency questions Q1–Q5 and Q5’ in the compressed graphs O’ and ON’ for TP properties and NTP properties respectively are examined next.

4.2.1. Additional statements for TP properties

In order to identify the impact of the TP properties in a knowledge base, it is appropriate to consider the way properties are axiomatised in RDFS. In particular, RDFS offers three properties to describe the semantics of a property (see Hayes and Patel-Schneider [6], Section 9.2.1):

- rdfs:domain
- rdfs:range
- rdfs:subPropertyOf

For each TP the domain is the same as the domain of Π . The range is always the range of ‘P2 has type’, i.e. ‘E55 Type’.

Therefore, assuming that Πd is the domain of Π , the following axiomatic triples capture the meaning of TP with respect to domain and range:

```
TP rdfs:domain  $\Pi d$  ;
   rdfs:range E55
```

For each super-property of Π a new TP’ property is required with similar definition:

```
TP’ a rdf:property ;
    H1  $\Pi'$  ;
    H2 P2 ;
    Hn false .
TP rdfs:subPropertyOf TP’
```

where Π' is a super-property of Π which is already defined in the CIDOC CRM specification.

The property ‘P2 has type’ does not have any super-properties but if that were the case additional properties TP’’ would be necessary for each super-property of P2.

4.2.2. Additional statements for NTP properties

A similar process is followed for NTP properties. In the following cases individuals $:i$ are considered belonging to \mathbb{I} which is a large set of individuals to be compressed as explained in Section 2. Statements with NTP properties can imply:

- a) that the individual $:i$ is not connected to $:s$ through Π , or
- b) that the individual $:i$ is not connected to $:t$ through P2

For both cases the domain and range of NTP are defined as follows:

```
NTP rdfs:domain  $\Pi d$  ;
    rdfs:range E55
```

Case b) is considered first. In case b), for every sub-property of P2 whose domain contains $:i$, a new property NTP’’ is required with similar definition:

```
NTP’’ a rdf:property ;
    H1  $\Pi$  ;
    H2 P2’ ;
    Hn true .
NTP’’ rdfs:subPropertyOf NTP
```

where P2’ is a sub-property of P2 with a valid domain, i.e. property ‘P137 exemplifies’. It is noted that ‘P177 assigned property type’ is also sub-property of ‘P2 has type’ but $:i$ does not belong to its domain ‘E13 Attribute Assignment’. Therefore although the statements:

```

:s NTP' :t .
NTP' a rdf:property ;
H1  $\Pi$  ;
H2 P177 ;
Hn true .
NTP' rdfs:subPropertyOf NTP

```

are logically true, they are not significant.

In case a), for every sub-property of Π , in whose domain $:i$ belongs, a new property NTP' is required with similar definition:

```

NTP' a rdf:property ;
    H1  $\Pi'$  ;
    H2 P2 ;
    Hn true .
NTP' rdfs:subPropertyOf NTP

```

where Π' is a sub-property of Π as already defined in the CIDOC CRM.

Adding these statements in the knowledge base for any of the three cases does not lead to contradictions in relation to the observations and it does not affect the capacity to answer Q5'.

In Section 5, it will be shown that the compressed graphs O' and ON' do allow to answer the competency questions, exemplifying the proof using a specific dataset for concreteness. In the rest of this section, the implications of the proposed solution are discussed in relation to documentation practice with the CIDOC CRM.

4.3. Observation, negation and categorical statements

The philosophical discourse around non-existence is often introduced with the concept of 'referential fallacy' (for example, see the discussion of the existence of Pegasus in [23]), i.e. the assumption that a referenced entity in a knowledge base exists in real life when it could be fictitious. Fictitious things are not considered in this paper. In heritage research and when producing documentation records the following are considered: a) a potentially real individual and b) the absence of any real individual.

4.3.1. A potentially real individual

A potentially real individual may be the result of interpreting references and other sources of evidence or the result of indirect observation. For example a publication referencing leaf markers existing on a specific book, or evidence of adhesive on the leaf at the location where a leaf marker would be expected, may indicate the existence of an individual leaf marker for a period of time. In these cases, the available knowledge only constitutes a finite set of constraints A_1, \dots, A_n on properties relating the potential real individual to individual $:s$ and type $:t$. The existence statement matches a real individual with the given constraints, in the example "there was a leaf marker at the specific location", given the evidence of adhesive. The non-existence statement is: "there exists no individual fulfilling the constraints A_1, \dots, A_n ". In the example: "there is no individual, instance of 'leaf marker', fulfilling the constraints of a bibliographical reference or evidence of adhesive at the expected location". The latter does not make a statement about the counterfactual individual (i.e. the leaf marker), but about the portion of reality covered by the constraints. Therefore it does not constitute a referential fallacy. The counterfactual instance should be instantiated only if the test on the applicable constraints is positive i.e. indicating a real individual.

Additional knowledge beyond these sets of constraints, such as direct observation, means that the individual is known to exist.

4.3.2. Absence of any real individual

Absence of any real individual is described by the same constraints. For example the question whether a book has leaf markers requires a complete observation of every leaf of the book which in itself is limited by constraints of the conditions of observation (for example, part of the book may be inaccessible). There may be cases when

previously unknown leaves of a book with leaf markers are reunited with the book thus creating a new boundary for complete observation (this is often illustrated in the field of biodiversity where previously thought extinct species have reappeared [25]). Therefore negative statements about the absence of any individuals presuppose complete observation within a set of constraints.

In the context of a knowledge base Razniewski and Nutt [22] have summarised the nature of partially-complete knowledge bases which follow neither the OWA, nor the CWA. In their work, knowledge base queries are characterised based on completeness to allow users to understand whether the results assume OWA or CWA. This characterisation can be done through providing contextual information about data completeness (i.e. similar to a set of constraints). Darari et al. [4] explore the Semantic Web as an Open World dataset with pockets of complete data under CWA. In a similar fashion they consider the certainty of answers as a metric to evaluate results of queries by comparing to a hypothetical complete dataset within a given context. In the example of leaf markers, completeness of observation is reflected by the material aspects of the book. The context of the limited Closed World for the NTP properties consists of: a) the domain of the property, i.e. the individual being completely observed, b) the range of the property, i.e. the type that it is observed for, and c) the original CIDOC CRM property Π included in the reification statements through H1, i.e. the kind of observation. In contrast, neither range instances of Π nor of P2 are observed completely.

4.3.3. Typed properties as categorical statements

The importance of categorical and cross-categorical knowledge in the CIDOC CRM has been discussed before [5]. Lin et al. [12] discuss issues around categorical knowledge using an example from the field of biodiversity: “The Kobra eats rodents and lives in India”. This statement is expressed as if the category of ‘Kobra snakes’ is an instance of a snake (instance of ‘E18 Physical Thing’) although in reality it is an instance of ‘E55 Type’. The example goes further mixing categories and individuals: “a specific snake of the type Kobra eats rodents”. This is in parallel to the example of a specific book carrying leaf markers. In order to accommodate such statements a proposal for the MetaCRM [1] was established where all domains and ranges of CIDOC CRM properties were replaced by ‘E55 Type’. These highlight the switch from statements about individuals to statements about types of things similar to TP properties. However, the TP properties additionally offer direct links through H1 to the original CIDOC CRM properties Π that they derive from and do not consider more uncertain modalities such as: “The Kobra typically eats rodents”.

4.4. Characteristics of typed properties

Characteristics of TP and NTP are considered next.

4.4.1. Typed properties as CIDOC CRM shortcuts

TP properties can be considered as shortcuts within the CIDOC CRM. For the example of P46, if the following statements are valid:

```
:s P46 :i .
:i P2 :t
```

then this is also valid:

```
:s TP46 :t
```

NTP properties are not CIDOC CRM shortcuts since the property of the chain for which the negation applies is unclear.

4.4.2. Existing CIDOC CRM typed properties

The scope note of the CIDOC CRM property ‘P125 used object of type’ reads: “This property associates an instance of E7 Activity to an instance of E55 Type, which defines the type of object used in an instance of E7 Activity, when the specific instance is either unknown or not of interest, such as use of ‘a hammer’.” Its sub-property ‘P32 used general technique’ can also be considered as typed property.

4.4.3. Hierarchy of typed properties

CIDOC CRM property inheritance applies to derived TP and NTP properties. This does not conflict with the additional statements as a result of the requirements for the RDFS entailment patterns.

4.4.4. Negative typed properties and thesauri

Thesauri used with the CIDOC CRM are often hierarchical using broader/narrower relationships provided by standards like ISO 25964-1:2011 [10] and SKOS [17]. For example, in the field of bookbinding history the Language of Bindings Thesaurus (LoB) [30] provides such relationships. The concept for ‘leaf markers’ has broader concept ‘bookmarks’. TP properties are consistent with broader relationships in thesauri, but NTP properties are not. For example, if:

```
:s NTP46 :leafMarkerType
```

it cannot be concluded that:

```
:s NTP46 :bookmarks
```

whereas the opposite stands. Therefore, the reasoning based on broader/narrower relationships in thesauri can be considered reversed when using NTP properties.

It is noted that the quality of the thesauri should be such that it allows such reasoning.

5. Application and correctness of the proposed solutions

5.1. Dataset

Data collected during the survey of the Library of the St. Catherine Monastery in Sinai, Egypt is used to demonstrate the two solutions. The data describes whether the manuscripts in the library feature leaf markers. In total there are 3,277 records [28]. Two of them are shown in Table 2 [29].

The records were encoded for the OWL solution as shown next:

```
ClassAssertion(crm:E22_Human-Made_Object
:fff7d74e-79f9-4805-8fc5-7395bc849fa0)
AnnotationAssertion(rdfs:label :fff7d74e-79f9-4805-8fc5-7395bc849fa0
"Arabica 0011"@en)
ClassAssertion(ObjectMinCardinality(1 crm:P46_is_composed_of
ObjectHasValue(crm:P2_has_type
lob:5423)) :fff7d74e-79f9-4805-8fc5-7395bc849fa0)
ClassAssertion(crm:E22_Human-Made_Object
:e009097f-d4d5-44c3-9e01-45c13a56f1a1)
AnnotationAssertion(rdfs:label :e009097f-d4d5-44c3-9e01-45c13a56f1a1
"Arabica 0002"@en)
ClassAssertion(ObjectMaxCardinality(0
crm:P46_is_composed_of ObjectHasValue(crm:P2_has_type
lob:5423)) :e009097f-d4d5-44c3-9e01-45c13a56f1a1)
```

Table 2
Example records of existence of leaf markers on manuscripts

shelfmark	uuid	leaf markers?
Arabica 0002	e009097f-d4d5-44c3-9e01-45c13a56f1a1	no
Arabica 0011	fff7d74e-79f9-4805-8fc5-7395bc849fa0	yes

The records were also encoded for the RDFS solution as shown next:

```
:e009097f-d4d5-44c3-9e01-45c13a56f1a1 a crm:E22_Human-Made_Object ;
rdfs:label "Arabica 0002"@en ;
crm:NTP46_137_is_not_composed_of_physical_thing_that_exemplifies lob:5423 ;
crm:NTP46_is_not_composed_of_physical_thing_of_type lob:5423 ;
crm:NTP56_does_not_bear_feature_of_type lob:5423 .
:fff7d74e-79f9-4805-8fc5-7395bc849fa0 a crm:E22_Human-Made_Object ;
rdfs:label "Arabica 0011"@en ;
crm:TP46_is_composed_of_physical_thing_of_type lob:5423 .
```

The URI lob:5423 corresponds to leaf marker type in the LoB thesaurus.

5.2. Competency questions and queries

Tables 3 and 4 show the queries for questions Q1–Q5 implemented in OWL DL query expressions and SPARQL for the OWL and RDFS solutions respectively where Q^f is the rewritten query Q for the proposed RDFS solution.

An example of a query involving super-properties of Π is included next:

```
SELECT ?s {
  ?s ?tp ?t .
  ?tp H1 ?p .
   $\Pi$  rdfs:subPropertyOf* ?p .
  ?tp Hn false
}
```

Q5' can be answered using OWL DL query expressions for the OWL solution in ON and ON'. It is noted that due to the OWA the assertions in ON cannot answer Q5' and the query can only be constructed involving individual members of I:

```
not (  $\Pi$  value :i )
```

In ON' the query can be formulated based on cardinality restrictions:

```
 $\Pi$  max 0 ( P2 value :t )
```

Q5' can be answered using SPARQL for the RDFS solution in ON' only. ON does not contain relevant RDFS statements. Negation in ON through SPARQL tools such as the MINUS operator (for example see [2]) does not

Table 3
OWL DL queries for questions Q1–Q5

Q	O, O'
Q1	Π some [instances]
Q2	:s [classes]
Q3	inverse P2 some [instances]
Q4	:t [classes]
Q5	Π some (P2 some)
	×
	inverse P2 some (inverse Π some)

Table 4
SPARQL queries for questions Q1–Q5

	Q on O	Q' on O'
Q1	<pre>SELECT ?s { ?s Π ?i }</pre>	<pre>SELECT ?s { ?s ?tp ?t . ?tp H1 Π . ?tp Hn false }</pre>
Q2	<pre>SELECT ?c { ?s a ?c }</pre>	<pre>SELECT ?c { ?s a ?c }</pre>
Q3	<pre>SELECT ?t { ?i P2 ?t }</pre>	<pre>SELECT ?t { ?s ?tp ?t . ?tp H2 P2 . ?tp Hn false }</pre>
Q4	<pre>SELECT ?c { ?t a ?c }</pre>	<pre>SELECT ?c { ?t a ?c }</pre>
Q5	<pre>SELECT ?s ?t { ?s Π ?i . ?i P2 ?t . }</pre>	<pre>SELECT ?s ?t { ?s ?tp ?t . ?tp H1 Π . ?tp H2 P2 . ?tp Hn false }</pre>

return relevant knowledge.

```
SELECT ?s ?t {
  ?s ?ntp ?t .
  ?ntp H1 Π .
  ?ntp H2 P2 .
  ?ntp Hn true
}
```

This query can also include sub-properties of Π:

```
SELECT ?s ?t {
  ?s ?ntp ?t .
  ?ntp H1 ?p .
  ?p rdfs:subPropertyOf* Π .
  ?ntp H2 P2 .
  ?ntp Hn true
}
```

The SPARQL query for Q5' can be articulated with ?t given it is matching a pattern in a triple store without concern about the negation context. As mentioned before, in OWL this query can only be articulated for an individual :t to respect the constrains of the observed Closed World.

5.3. Identifying contradictions

The identification of contradictory statements about the existence of individuals is important for scholarship as they indicate areas of further discussion. In OWL such contradictions are automatically identified. For example the following is inconsistent:

```
EquivalentClasses (
  :books_with_leafmarkers
  ObjectMinCardinality( 1 P46
    ObjectHasValue( P2 lob:5432 ) ) )
EquivalentClasses (
  :books_without_leafmarkers
  ObjectMaxCardinality( 0 P46
    ObjectHasValue( P2 lob:5432 ) ) )
ClassAssertion(
  :books_without_leafmarkers :book1 )
ObjectPropertyAssertion( P2 :leafMarker1
  lob:5432 )
ObjectPropertyAssertion( P46 :book1
  :leafMarker1 )
```

An assertion that the individual `:book1` has a component of type ‘leaf marker’ (`lob:5432`) contradicts the assertion of `:book1` belonging to the class of books without leaf markers.

In RDFS statements matching contradictory observations can be identified through SPARQL queries:

```
SELECT ?s {
  ?s ?ntp :t .
  ?ntp H1 Π .
  ?ntp H2 P2 .
  ?ntp Hn true .
  ?s ?tp :t .
  ?tp H1 Π .
  ?tp H2 P2 .
  ?tp Hn false .
}
```

This will identify the individuals `:s` which are connected to type `:t` with both a positive and a negative typed property derived from Π , indicating an inconsistency in observation.

5.4. Proof of correctness

It is now possible to show the correctness of the method by proving the following proposition using the notions and the notations introduced in [6]:

Proposition. *For every competency query Q , on graphs O and O' as defined in Section 2, the answer to Q on O , $\text{ans}(Q, O)$, is the same as the answer to Q^f on O' $\text{ans}(Q^f, O')$.*

Proof. The proof is carried out only for $Q1$ and the positive case, as the proofs for the other queries and the negative case are similar. For every RDFS model J of O , there exists a corresponding model H of O' such that $Q1^J = (Q1^f)^H$.

Using the notation introduced in Section 2, $Q1$ is $(?s \ \Pi \ \exists i)$ which asks for the individuals connected through Π to any individual. Intuitively, for every triple:

$$s \ \Pi \ i$$

in O the following triples, among others, are inserted in O' :

$$\begin{array}{l} TP \ H1 \ \Pi \ . \\ s \ TP \ t \end{array}$$

and it is easy to see that s shows up in the answer of both Q and Q' . Formally, let J be an RDFS model of O . By hypothesis, $s^J \in Q^J$ that is, $(s^J, i^J) \in \Pi^J$. Now, let H be the corresponding RDFS model of O' . H is the same as J except that it does not satisfy the removed triples and satisfies the added triples in O' . Then $(s^H, t^H) \in TP^H$ and $(TP^H, P^H) \in H1^H$, which implies that $s \in (Q')^H$. Since this applies to every RDFS model J of O , $\text{ans}(Q, O) = \text{ans}(Q', O')$. \square

6. Summary of conclusions

When documenting heritage, the following two problems often appear a) how can the typology of numerous individuals be recorded without including them in the knowledge base and b) how the non-existence of individuals can be recorded. These problems were summarised with a set of competency questions in comparison to the knowledge available when individuals are included. The competency questions were then filtered based on the significance of the knowledge for research.

Following a review of potential solutions, in OWL the use of cardinality restrictions is recommended as an optimal solution as it excludes statements about the numerous individuals $:i$ and allows queries for the significant individuals $:s, \Pi$ and $:t$. In RDFS new properties with reification statements are proposed to describe property chains for typed properties (TP) and negative typed properties (NTP). These reification statements link the significant individuals for answering the competency questions. RDFS entailment patterns were examined to identify additional required statements.

When describing the non-existence of individuals, the reified statements of NTP properties apply to both parts of the property chain when, in reality, it could be that only one of them is negated, but negating both does not have a negative impact on the results of the competency questions.

The use of NTP properties requires a context to define completeness of observation. In practice this means full capacity to observe the individual. The proposed NTP properties derive from the CIDOC CRM properties. Completeness of observation is described by the domain and range of the NTP property as well as the original CIDOC CRM property from which the NTP property is derived.

TP properties are shortcuts in the CIDOC CRM whereas NTP properties are not. The hierarchy of TP and NTP properties mirrors that of the CIDOC CRM property hierarchy. When discussing reasoning about broader/narrower concepts from thesauri, statements using NTP properties also apply to narrower terms of a thesaurus in contrast to statements using TP properties where this is not the case.

7. Future work

An implementation extension of the CIDOC CRM which will allow easy use of TP and NTP properties is in preparation. The development of that extension is undertaken as part of work for the Linked Conservation Data project [13]: a project which explores ways of sharing data produced by conservators with significant representation from book and paper conservators working with historic books. The progress of the development of the extension can be followed in the Linked Conservation Data GitHub repository [14].

Acknowledgements

The authors thank the CIDOC CRM special interest group. This work has been initiated by the Linked Conservation Data project and is partly funded by the Arts and Humanities Research Council in the UK. The problems were introduced by Prof. Nicholas Pickwood during the condition survey of the manuscripts of the library of the St. Catherine's Monastery in Sinai, Egypt.

References

- [1] 11th Meeting on FRBR/CRM Harmonization together with 16th CIDOC CRM SIG Meeting Germanisches Nationalmuseum, Nuremberg 4–7 December 2007, CIDOC CRM Conceptual Reference Model, 2007. <http://www.cidoc-crm.org/Meeting/16th-cidoc-crm-and-11th-frbr-crm> (accessed February 25, 2021).
- [2] R. Angles and C. Gutierrez, Negation in SPARQL, in: *The 10th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW 2016)*, Panama City, Panama, 2016, <http://ceur-ws.org/Vol-1644/paper11.pdf>.
- [3] H. Arnaout, S. Razniewski and G. Weikum, Enriching knowledge bases with interesting negative statements, in: *Automated Knowledge Base Construction (AKBC 2020)*, online, 2020. doi:10.24432/C5101K.
- [4] F. Darari, S. Razniewski and W. Nutt, Bridging the semantic gap between RDF and SPARQL using completeness statements, in: *The 13th International Semantic Web Conference Riva del Garda, Trentino, Italy, 2014*. arXiv:1408.6395 [Cs] (accessed February 12, 2021).
- [5] M. Doerr, Modelling learning subjects as relationships, in: *Intuitive Human Interfaces for Organizing and Accessing Intellectual Assets*, G. Grieser and Y. Tanaka, eds, Springer, Berlin, Heidelberg, 2005, pp. 201–214. doi:10.1007/978-3-540-32279-5_14.
- [6] P. Hayes and P.F. Patel-Schneider, RDF 1.1 Semantics, World Wide Web Consortium, Feb. 25, 2014. <https://www.w3.org/TR/rdf11-nt/#rdfs-entailment> (accessed Mar. 01, 2022).
- [7] D. Hernández, A. Hogan and M. Krötzsch, Reifying RDF: What works well with wikidata? in: *SSWS@ISWC*, Vol. 1457, 2015, pp. 32–47.
- [8] A. Hogan, M. Arenas, A. Mallea and A. Polleres, Everything you always wanted to know about blank nodes, *Journal of Web Semantics* **27–28** (2014), 42–69. doi:10.1016/j.websem.2014.06.004.
- [9] International Organization for Standardization, ISO 21127: Information and documentation – a reference ontology for the interchange of cultural heritage information, ISO, Geneva, 2006.
- [10] International Organization for Standardization, ISO 25964-2: Information and documentation – thesauri and interoperability with other vocabularies, ISO, Geneva, 2013.
- [11] leaf markers, Language of Bindings (n.d.). <https://w3id.org/lob/concept/5423> (accessed February 25, 2021).
- [12] C.-H. Lin, J.-S. Hong and M. Doerr, Issues in an inference platform for generating deductive knowledge: A case study in cultural heritage digital libraries using the CIDOC CRM, *International Journal of Digital Libraries* **8** (2008), 115–132. doi:10.1007/s00799-008-0034-0.
- [13] Linked Conservation Data, (n.d.) <https://www.ligatus.org.uk/lcd/> (accessed February 24, 2021).
- [14] Linked Conservation Data Consortium, linked-conservation-data/crmntp, 2021. <https://github.com/linked-conservation-data/crmntp> (accessed February 24, 2021).
- [15] A. Mallea, M. Arenas, A. Hogan and A. Polleres, On blank nodes, in: *The Semantic Web – ISWC 2011–10th International Semantic Web Conference*, Bonn, Germany, October 23–27, 2011, I.
- [16] C. Meghini and M. Doerr, A first-order logic expression of the CIDOC conceptual reference model, *International Journal of Metadata, Semantics and Ontologies* **13** (2018), 131–149. doi:10.1504/IJMSO.2018.098393.
- [17] A. Miles and S. Bechhofer, SKOS simple knowledge organization system reference, World Wide Web Consortium, 2009. <http://www.w3.org/TR/2009/REC-skos-reference-20090818/> (accessed December 30, 2015).
- [18] P. Mirza, S. Razniewski, F. Darari and G. Weikum, Enriching knowledge bases with counting quantifiers, in: *The Semantic Web – ISWC 2018*, 2018, pp. 179–197. doi:10.1007/978-3-030-00671-6_11.
- [19] B. Motik, P.F. Patel-Schneider and B. Parsia, OWL 2 web ontology language structural specification and functional-style syntax (Second Edition), World Wide Web Consortium (2012). <https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/> (accessed March 30, 2022).
- [20] N. Pickwood, *Assessment Manual*, University of the Arts London, London, 2005, <https://www.ligatus.org.uk/node/19>.
- [21] N. Pickwood, Recording medieval bindings – the role of the conservation survey, with reference to work currently under way in the library of the monastery of St Catherine on Mount Sinai, in: *La Reliure Médiévale*, G. Lanoë and G. Grand, eds, Brepols, Paris, 2008, pp. 47–59.
- [22] S. Razniewski and W. Nutt, Databases under the partial closed-world assumption: A survey, in: *Proceedings of the 26th GI-Workshop Grundlagen von Datenbanken, CEUR*, Bozen-Bolzano, Italy, 2014, p. 6.
- [23] M. Reicher, Nonexistent objects, in: *The Stanford Encyclopedia of Philosophy, Winter 2019*, E.N. Zalta, ed., Metaphysics Research Lab, Stanford University, 2019, <https://plato.stanford.edu/archives/win2019/entries/nonexistent-objects/> (accessed February 1, 2021).
- [24] Y. Ren, A. Parvizi, C. Mellish, J.Z. Pan, K. van Deemter and R. Stevens, Towards competency question-driven ontology authoring, in: *The Semantic Web: Trends and Challenges*, Cham, 2014, pp. 752–767. doi:10.1007/978-3-319-07443-6_50.
- [25] M. Sakashita, Sea Otter, Centre for Biological Diversity. (n.d.). https://www.biologicaldiversity.org/species/mammals/sea_otter/index.html (accessed February 24, 2021).

- [26] V. Svátek, J. Klůka, M. Vacura and M. Homola, Patterns for referring to multiple indirectly specified objects (MISO): Analysis and guidelines, in: *Advances in Pattern-Based Ontology Engineering*, Vol. 51, IOS Press, 2021, pp. 1–24, Available: <https://ebooks.iospress.nl/volume/advances-in-pattern-based-ontology-engineering>.
- [27] A. Velios, Bookbinding descriptions in a linked data world: How the CIDOC-CRM can improve research in bookbinding history, in: *Bookbindings, Bibliologia*, N. Golob and J. Vodopivec Tomažič, eds, Brepols, Turnhout, 2017, pp. 13–26.
- [28] A. Velios, TP and NTP full dataset, University of the Arts London, Mar. 31, 2022. doi:10.25441/arts.19487411.
- [29] A. Velios, TP and NTP small dataset, University of the Arts London, Mar. 31, 2022. doi:10.25441/arts.19487468.
- [30] A. Velios and N. Pickwood, The development of the language of bindings thesaurus, in: *Book Conservation and Digitization – the Challenges of Dialogue and Collaboration*, A. Campagnolo, ed., ARC Humanities Press, 2020, pp. 157–168. doi:10.2307/j.ctv13gvhxx.14.