Erlangen OWL and CRM RDF

Steps towards CRM OWL

CIDOC SIG, Lux 2022
December 07, 2022

Mark Fichtner

Germanisches Nationalmuseum Nürnberg

Erlangen CRM / OWL

Motivation

- The CIDOC CRM document is now very well structured
- Automatic processing on RDF layer proofs to be good
- No „electorial process" is necessary anymore for most of the things
- The RDF and the ECRM OWL are now much closer than ever before

Erlangen CRM / OWL

Aim

- (Semi-)automatically create an CIDOC CRM OWL

- Have this CIDOC CRM OWL with CIDOC and Erlangen namespaces for the classes for backward compatiblity (owl:sameAs or owl:equivalentClass)

- At best: Have one ruleset for RDF and OWL and only do different things where it is really necessary

- However there are currently some differences between RDF and OWL that are more or less cosmetic -> We need to decide which way to go

Erlangen CRM / OWL

- Multilingual Labels:
  - ECRM has only english labels, rdf has multiple languages

```
<rdfs:label xml:lang="en">E1 CRM Entity</rdfs:label>
```

```
<rdfs:Class rdf:about="E1_CRM_Entity">
  <rdfs:label xml:lang="en">CRM Entity</rdfs:label>
  <rdfs:label xml:lang="de">CRM Entität</rdfs:label>
  <rdfs:label xml:lang="el">Οντότητα CIDOC CRM</rdfs:label>
  <rdfs:label xml:lang="fr">Entité CRM</rdfs:label>
  <rdfs:label xml:lang="pt">Entidade CRM</rdfs:label>
  <rdfs:label xml:lang="ru">CRM Сущность</rdfs:label>
  <rdfs:label xml:lang="zh">CRM实体</rdfs:label>
```

- Structure of |

  - ECRM inclu...   not

```
<rdfs:label xml:lang="en">E1 CRM
```

```
ss rdf:about="E1_CRM_Entity">
label xml:lang="en">CRM Entity</rdfs:label>
label xml:lang="de">CRM Entität</rdfs:label>
label xml:lang="el">Οντότητα CIDOC CRM</rdfs:label>
label xml:lang="fr">Entité CRM</rdfs:label>
label xml:lang="pt">Entidade CRM</rdfs:label>
label xml:lang="ru">CRM Сущность</rdfs:label>
label xml:lang="zh">CRM实体</rdfs:label>
```

Erlangen CRM / OWL

- SKOS Notation:
  - ECRM has additionally a skos:notation element, rdf does not have that

```
<skos:notation rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
>E1</skos:notation>
```

Erlangen CRM / OWL

- Structure of comments:
  - RDF just has the scope note, ECRM has scope note, examples and FOL

```
    <rdfs:comment>This class comprises all things in the universe of discourse of the CIDOC Conceptual Reference Model.
It is an abstract concept providing for three general properties:
Identification by name or appellation, and in particular by a preferred identifier
Classification by type, allowing further refinement of the specific subclass an instance belongs to
Attachment of free text and other unstructured data for the expression of anything not captured by formal properties
All other classes within the CIDOC CRM are directly or indirectly specialisations of E1 CRM Entity.</rdfs:comment>
```

```
    <rdfs:comment xml:lang="en">This class comprises all things in the universe of discourse of the CIDOC Conceptual Reference Model.

It is an abstract concept providing for three general properties:
1. Identification by name or appellation, and in particular by a preferred identifier
2. Classification by type, allowing further refinement of the specific subclass an instance belongs to
3. Attachment of free text and other unstructured data for the expression of anything not captured by formal properties

All other classes within the CIDOC CRM are directly or indirectly specialisations of E1 CRM Entity.


Examples:
- the earthquake in Lisbon 1755 (E5) (Chester, 2001)


In First Order Logic:
E1(x)</rdfs:comment>
```

- Comments and where to place them:
  - ECRM and RDF place the comments only on Classes and forward-Properties with only some exceptions.
  - Decision: Keep it like that or also put it on inverses?

Erlangen CRM / OWL

- Support for .1 properties
  - RDF has a „special" support for .1 properties. This approach would also be possible for OWL.
- Transitive/SymmetricProperty where it fits
  - RDF does not have that, so it is not implemented. OWL should implement it
- Reflexive/Non-Reflexive
  - won't be implemented by both as it is not really possible yet.
- Multiple Inheritance for „primitives" like P1
  - No multiple inheritance -> ObjectProperty in OWL
- Restrictions?
- OWL 1 vs. OWL 2?

Thank you!