An Implementation of the CIDOC CRM in OWL-DL

Guenther Goerz, Bernhard Schiemann, Martin Oischinger
University of Erlangen-Nuremberg
Computer Science Department

© Doerr
The CIDOC Conceptual Reference Model

- A **reference ontology** with a particular focus on cultural heritage information and documentation ... and more

- Authoritative reference: 
  *Definition of the CIDOC Conceptual Reference Model*, ver. 4.2.4 (Jan. 2008)
  - 87 **classes** (hierarchy)
  - 148 **properties** (and inverses)

... "to be explained"

G. Görz, FAU, Inf. 8
Knowledge Modelling

• “Formal Ontology“: Theory of a domain of discourse (rational reconstruction), first of all: Normalization of terminology
  - Concepts / classes (abstraction): “is“
  - Properties / relations: “has“
  - Constraints and rules
  - Individuals: Object descriptions
Formal Domain and Reference Ontologies

- Domain ontologies define the terminological system of a domain of discourse (theory based)
- Reference ontologies define
  - Generic, not domain specific concepts (e.g. person, event, place, time, ...)
  - Fundamental logical/mathematical terms (class, relation, number; mereology)
CIDOC CRM
Top Level Classes

Types

refine

Actors

Conceptual Objects

Physical Entities

Temporal Entities

Appellations

identify/name

participate in

affect

within

occur at

location

Time-Spans

© T. Gill
Interpreting the CRM Document

• Definition in terms of „scope notes“ and examples
  - Scope notes often consist of definition and best practice recommendations;
  - are in many places intentionally underspecified.
  - Occasionally problematic choice of terms
  - Continuous improvement of the text (ver.4.2.4. !), but ...
Interpreting the CRM Document

• As any text, it requires interpretation: The description of intricate semantic problems in common language is not only error-prone, but also in danger of vagueness and a certain degree of ambiguity.

• Clarification by translation into a logic-based language;  
  - offers opportunities to uncover methodological problems.

G. Görz, FAU, Inf. 8
Why an (OWL-DL) Implementation?

• Study feasibility of CRM for implementation
  - which (formal) linguistic features are required?

• Make CRM available for automatic processing
  ⇒ practical application(s)

• Support of interoperability and data integration
  - preprocessing (data transformation)
  - at access time (inference)

• DL (Logic): Efficient reasoning services
Why an (OWL-DL) Implementation?

• Check consistency of the CRM definition
  - With ~ 90 concepts and ~ 150 properties hard to see whether there are no contradictions
• Check for redundancies; study underspecification
• Processing of complex queries requiring inference
• Check consistency and coherence of CRM extensions
From Data Model to Semantics

Representation of the

- **meaning** of content words („concepts“) is relational: Network of relations;
- **reference** by “external” grounding.

Logical framework

- Logical composition of expressions based on discourse rules / validity criteria
- Reasoning by sound and complete inference rules
- Transition from **data model to semantics**!
Knowledge and Reasoning

Why don't we just employ standard “First Order” Logic?

- It is too powerful:
  The problem of deciding whether a formula is logically implied by a theory is undecidable, i.e., there is no algorithm that solves the problem in a finite number of steps (a problem property!).

- It is too poor:
  A lot of language constructs convenient for knowledge representation are missing.
A Solution: **Description Logics**

- A family of logic languages, taking advantage of both worlds, logic and knowledge representation languages
- Explore the “most” interesting expressive decidable logics with “classical” semantics, equipped with “good” reasoning procedures
  - **Sound**: no wrong inferences are drawn
  - **Complete**: all the correct inferences are drawn
- **OWL** (“Web Ontology Language”) is a very expressive description logic language.

G. Görz, FAU, Inf. 8
DL Semantics

given by standard model theory:

Interpretation function $\mathcal{I}$

- **Individuals** $i^\mathcal{I} \in \Delta^\mathcal{I}$
  - John
  - Mary

- **Concepts** $C^\mathcal{I} \subseteq \Delta^\mathcal{I}$
  - Lawyer
  - Doctor
  - Vehicle

- **Roles** $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$
  - hasChild
  - owns

Example: 

$\text{Lawyer} \sqcap \text{Doctor}$

© Horrocks

G. Görz, FAU, Inf. 8
DL Knowledge Bases

Separation into

- **T-Box**: conceptual (Terminological) knowledge
  - Concepts (classes), properties, constraints
  - Inheritance hierarchy

- **A-Box**: knowledge about individuals, i.e. concept instances (Assertional)
Inferences

- Concept satisfiability, satisfiability of the whole knowledge base (consistency checking)
- Subsumption: Automatic Classification of concept and instance descriptions
- Proper instantiation
- Realization and retrieval (answering complex queries, ...)

G. Görz, FAU, Inf. 8
„Semantic Web“: Language Layers

- Query: SPARQL
- Ontology: OWL
- Data interchange: RDF
- URI/IRI
- Rule: RIF
- RDFS
- XML
- Proof
- Unifying Logic
- Trust

⇐ OWL-DL
XML: Data Representation

XML: eXtended Markup Language
- Metalanguage
- Grammar / Schema
  \(\sim\) special Markup Language

© Lagoze

G. Görz, FAU, Inf. 8
OWL as RDF(S) extension (1/2)

- class-def
- subclass-of
- slot-def
- subslot-of
- domain
- range

- class-expressions
  - AND, OR, NOT
- slot-constraints
  - has-value, value-type
  - cardinality
- slot-properties
  - trans, symm

© v. Harmelen
OWL as RDF(S) extension (2/2)

```xml
<rdfs:Class rdf:ID="herbivore">
  <rdf:type>
    rdf:resource="http://www.ontoknowledge.org/#DefinedClass"/>
  <rdfs:subClassOf rdf:resource="#animal"/>
  <rdfs:subClassOf>
    <owl:NOT>
      <owl:hasOperand rdf:resource="#carnivore"/>
    </owl:NOT>
  </rdfs:subClassOf>
</rdfs:Class>
```

© v. Harmelen
OWL as RDF(S) extension (2/2)

```xml
<rdfs:Class rdf:ID="herbivore">
  <rdfs:subClassOf rdf:resource="#animal"/>
  <rdfs:subClassOf>
    <oil:NOT rdf:resource="#carnivore"/>
  </rdfs:subClassOf>
</rdfs:Class>
```

© v. Harmelen
## OWL DL: Concept Expressions

<table>
<thead>
<tr>
<th>Constructor</th>
<th>DL Syntax</th>
<th>Example</th>
<th>FOL Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>intersectionOf</td>
<td>$C_1 \cap \ldots \cap C_n$</td>
<td>Human $\cap$ Male Doctor $\sqcup$ Lawyer $\neg$ Male</td>
<td>$C_1(x) \land \ldots \land C_n(x)$ $C_1(x) \lor \ldots \lor C_n(x)$ $\neg C(x)$</td>
</tr>
<tr>
<td>unionOf</td>
<td>$C_1 \cup \ldots \cup C_n$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>complementOf</td>
<td>$\neg C$</td>
<td></td>
<td>$\neg C(x)$</td>
</tr>
<tr>
<td>oneOf</td>
<td>${x_1} \cup \ldots \cup {x_n}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>allValuesFrom</td>
<td>$\forall P.C$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>someValuesFrom</td>
<td>$\exists P.C$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>maxCardinality</td>
<td>$\leq nP$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>minCardinality</td>
<td>$\geq nP$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $C$ is a concept (class); $P$ is a role (property); $x$ is an individual name
- **XMLS data types** and also classes in $\forall P.C$ and $\exists P.C$
  - e.g. $\exists$hasAge.$\text{nonNegativeInteger}$
- Unlimited **nesting of constructors**
  - e.g., $\text{Person} \sqcap \forall \text{hasChild.}(\text{Doctor} \sqcup \exists \text{hasChild.Doctor})$

_G. Görz, FAU, Inf. 8_
## OWL DL: Axioms

<table>
<thead>
<tr>
<th>Axiom</th>
<th>DL Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>subClassOf</td>
<td>$C_1 \sqsubseteq C_2$</td>
<td>Human $\sqsubseteq$ Animal $\sqcap$ Biped</td>
</tr>
<tr>
<td>sameClassAs</td>
<td>$C_1 \equiv C_2$</td>
<td>Man $\equiv$ Human $\sqcap$ Male</td>
</tr>
<tr>
<td>disjointWith</td>
<td>$C_1 \sqsubseteq \neg C_2$</td>
<td>Male $\sqsubseteq \neg$ Female</td>
</tr>
<tr>
<td>sameIndividualAs</td>
<td>${x_1} \equiv {x_2}$</td>
<td>${\text{President}<em>\text{Bush}} \equiv {\text{G}</em>\text{W}_\text{Bush}}$</td>
</tr>
<tr>
<td>differentIndividualFrom</td>
<td>${x_1} \sqsubseteq \neg {x_2}$</td>
<td>${\text{john}} \sqsubseteq \neg {\text{peter}}$</td>
</tr>
<tr>
<td>subPropertyOf</td>
<td>$P_1 \sqsubseteq P_2$</td>
<td>hasDaughter $\sqsubseteq$ hasChild</td>
</tr>
<tr>
<td>samePropertyAs</td>
<td>$P_1 \equiv P_2$</td>
<td>cost $\equiv$ price</td>
</tr>
<tr>
<td>inverseOf</td>
<td>$P_1 \equiv P_2^-$</td>
<td>hasChild $\equiv$ hasParent$^-$</td>
</tr>
<tr>
<td>transitiveProperty</td>
<td>$P^+ \sqsubseteq P$</td>
<td>ancestor$^+ \sqsubseteq$ ancestor</td>
</tr>
<tr>
<td>uniqueProperty</td>
<td>$\top \sqsubseteq \leq 1P$</td>
<td>$\top \sqsubseteq \leq 1$ hasMother</td>
</tr>
<tr>
<td>unambiguousProperty</td>
<td>$\top \sqsubseteq \leq 1P^-$</td>
<td>$\top \sqsubseteq \leq 1$ hasSSN$^-$</td>
</tr>
</tbody>
</table>

\[ \mathcal{I} \text{ satisfies } C_1 \sqsubseteq C_2 \text{ iff } C_1^\mathcal{I} \subseteq C_2^\mathcal{I} \; ; \text{ satisfies } P_1 \sqsubseteq P_2 \text{ iff } P_1^\mathcal{I} \subseteq P_2^\mathcal{I} \]

\[ \mathcal{I} \text{ satisfies ontology } \mathcal{O} \text{ (is a model of } \mathcal{O} \text{) iff satisfies every axiom in } \mathcal{O} \]

G. Görz, FAU, Inf. 8
History of our CRM 4.2.4 Implementation in OWL-DL

• Earlier (incomplete) implementations available, e.g.
  - in RDF
  - in OWL-DL: ver. 3.4.9 by Aldo Gangemi (Trento)
• Started in 2007 as a student project
  - to be used in in-house projects
  - Status: research
• 2008: First external users
  - e.g. Marco Neumann (MetMA New York)
  - Continuous improvement

G. Görz, FAU, Inf. 8
Principles of the Implementation

- The CRM document should serve as the primary reference for the implementation.
- Stay as close as possible to the specifications in the CRM document.
- Whatever is underspecified or unspecified in the CRM document has been left open in the implementation as well.
- There are some features which could not be implemented as described or have not been implemented for certain reasons.
Tools

• **Ontology editor Protégé** (Stanford U.)
  - equipped for OWL-DL
  - Various plugins, e.g. for visualization

• **DL Inference engine Racer** (TU Hamburg-Harburg)

• Both cooperating in client-server architecture
Ontology Editor Protégé: CRM in OWL-DL
Ontology Editor Protégé: CRM in OWL-DL
Specification of E77.Persistent_Item

- Subclass of E1.CRM_Entity

- Scope Note: This class comprises items that have a persistent identity, sometimes known as endurants in philosophy. They can be repeatedly recognized within the duration of their existence by identity criteria rather than by continuity or observation. Persistent Items can be either physical entities, such as people, animals or things, or conceptual entities such as ideas, concepts, products of the imagination or common names.

- Examples: Leonardo da Vinci, Stonehenge, the hole in the ozon layer
E77. Persistent_Item

G. Görz, FAU, Inf. 8
E77.Persistent_Item in OWL-DL (1)

```xml
<owl:Class rdf:about="#E77.Persistent_Item">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#P921.was_brought_into_existence_by"/>
      </owl:onProperty>
    </owl:Restriction>
    <owl:someValuesFrom>
      <owl:Class rdf:about="#E63.Beginning_of_Existence"/>
    </owl:someValuesFrom>
  </rdfs:subClassOf>
</owl:Class>
```

G. Görz, FAU, Inf. 8
E77.Persistent_Item in OWL-DL (2)

```xml
<owl:cardinality rdf:datatype="#int" >1</owl:cardinality>
<owl:onProperty>
  <owl:ObjectProperty rdf:about="#P92I.was_brought_into_existence_by"/>
</owl:onProperty>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="#E1.CRM_Entity"/>
...
</owl:Class>
```
Properties of E77.Persistent_Item

G. Görz, FAU, Inf. 8
P92I.was_brought_into_existence_by in OWL-DL

<owl:ObjectProperty
  rdf:about="#P92I.was_brought_into_existence_by">
    <rdfs:range
      rdf:resource="#E63.Beginning_of_Existence"/>
    <rdfs:subPropertyOf
      rdf:resource="#P12I.was_present_at"/>
    <owl:inverseOf>
      <owl:ObjectProperty
        rdf:about="#P92.brought_into_existence"/>
    </owl:inverseOf>
    <rdfs:domain
      rdf:resource="#E77.Persistent_Item"/>
</owl:ObjectProperty>
Datatypes in CRM

- Used for representation of strings, etc.
- Datatype properties are not permitted as inverse-functional properties (as opposed to object properties) \( \Rightarrow \) Contradictions!
  - Entities point to E59.Primitive_Value or subclasses via inverse-functional object properties
  - Datatype properties point to xsd properties of XML Schema
  - cf. following example
E52.Time_Span

- E1.CRM_Entity
  - E2.Temporal_Entity
    - E52.Time_Span
    - E53.Place
    - E54.Dimension
  - E77.Persistent_Item
  - E59.Primitive_Value

- E1.CRM_Entity
  - P41.is_time-span_of some E2.Temporal_Entity
  - P79.beginning_is_qualified_by max 1
  - P81.ongoing_throughout exactly 1
  - P83.had_at_least_duration exactly 1
  - P84.had_at_most_duration exactly 1
  - P3.has_note some E62.String

G. Görz, FAU, Inf. 8
Properties of E52.Time_Span

- P119.meets_in_time_with ↔ P119.meets_in_time_with
- P115l.is_finished_by ↔ P115l.is_finished_by
- P86.falls_within ↔ P86.falls_within
- P117l.includes ↔ P117l.includes
- P117.occurs_during ↔ P117.occurs_during
- P90.has_value
- P57.has_number_of_parts
- P3.has_note
  - P80.end.is_qualified_by
  - P79.beginning.is_qualified_by
- P81.ongoing_throughout
- P82.at_some_time_within
E62.String
The Problem with E55.Type

• The class E55.Type (notice: different from the term “type” in computer science!) is described as a “metaclass”.
  - Which of you know what that is and what you could get from it??
  - With metaclasses, which are higher-order logic constructs, decidability is lost.

• Therefore, E55.Type has been implemented as a class which - for the purpose of reasoning on the conceptual level - may serve as an interface to external concepts of formal domain ontologies (or thesauri) as subclasses or as individuals.

G. Görz, FAU, Inf. 8
Working with E55.Type

- Easiest way to attach concepts of a domain ontology to the CRM is direct subclassing, e.g., the (domain) class **Artist** as a subclass of **E21.Person**.
  - E.g., “Vincent van Gogh” would be an instance of **Artist** and inherit all properties of **E21.Person**.
  - Representing **Artist** also as a subclass of **E55.Type** would lead to contradictions.

- Alternatively, use an individual “**Artist**”, e.g., a term in a domain-specific thesaurus
  - admitted in OWL-DL with the **one-of** construct.
  - So, we could represent “Vincent van Gogh” as an immediate instance of **E21.Person** and relate it by **P2.has type** to **E55.Type** with value “**Artist**”.
  - In this case, the individuals cannot have instances in turn!

G. Görz, FAU, Inf. 8
E55.Type: Attaching Thesauri, etc.

- The second way is precisely what the SKOS recommendation proposes ("Simple Knowledge Organization System"):  
- Connect CRM with a thesaurus  
  - Given a thesaurus in SKOS containing the term "Painting"  
  - "Painting" should be value of E55.Type  
  - Use dedicated annotation properties to bridge them
    
    ex:PaintingClass rdf:type owl:Class.  
    ex:PaintingConcept rdf:type skos:Concept.  
    ex:PaintingClass ex:correspondingConcept ex:PaintingConcept.

G. Görz, FAU, Inf. 8
SKOS ("Simple Knowledge Organization System")

Representation language for thesauri etc. (W3C), based on RDF

© Miles et al.
The GNM-DMS Ontology as an Extension to the CRM
CIDOC-CRM OWL-DL

About
This is the website of the OWL-DL 1.0 Implementation of the CIDOC Conceptual Reference Model.
The OWL-DL Version of the CRM is being developed by Martin Oischinger, Bernhard Schiemann and Günther Götz at the University of Erlangen-Nuremberg, Department of Computer Science, Chair of Artificial Intelligence and supported by the Department of Museum Informatics of the Germanisches Nationalmuseum Nuremberg.

Information about the CIDOC CRM itself and related work is available on the corresponding cidoc website. The CRM is also available as ISO 21127.

Resources & Namespaces
- Current version:
  http://www.cidoc-crm.org/2008/09/01/cidoc-crm-4.2.4.owl#
  (332 KB, OWL-DL 1.0)
- Current version without cardinality restrictions ("weak" version):
  http://www.cidoc-crm.org/2008/09/01/cidoc-crm-4.2.4.weak.owl#
  (271 KB, OWL-DL 1.0)

Documentation
- Full documentation of the current version (created with OWLDoc)
  http://www.cidoc-crm.org/doc/
- Martin Oischinger, Bernhard Schiemann, Günther Götz
  Short Documentation of the CIDOC CRM (4.2.4) Implementation in OWL-DL
  (46 KB, PDF)