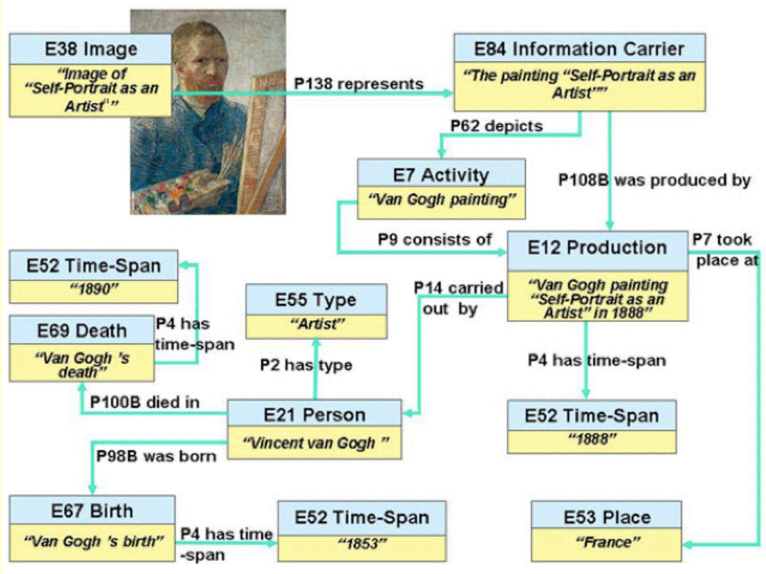
An Implementation of the CIDOC CRM in OWL-DL



© Doerr

Guenther Goerz, Bernhard Schiemann, Martin Oischinger

University of Erlangen-Nuremberg

Computer Science Department

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*"

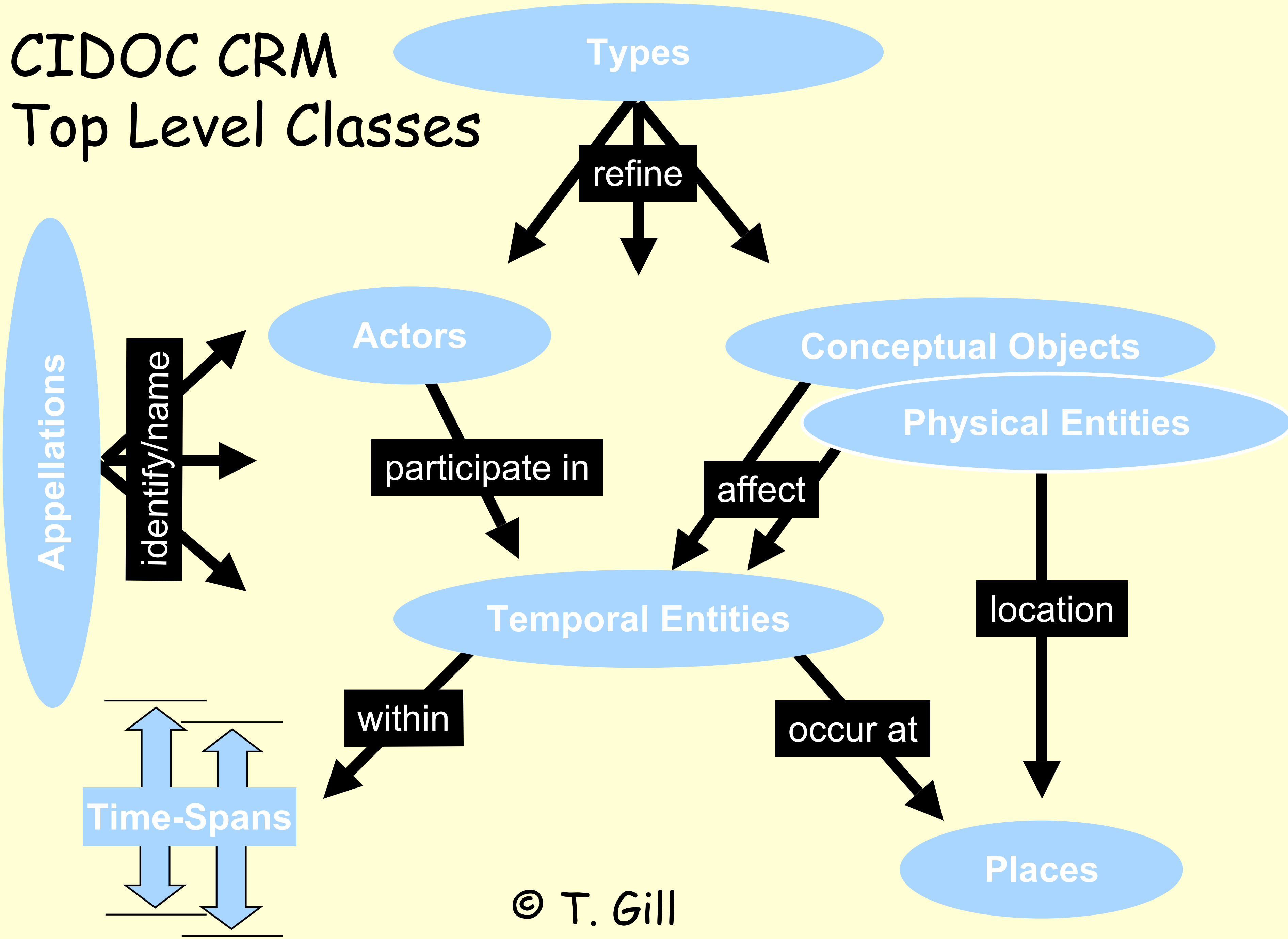
Knowledge Modelling

- “**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



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.

Why an (OWL-DL) Implementation?

- Study feasibility of CRM for implementation
  - which (formal) linguistic features are required?
- Make CRM available for automatic processing $\Rightarrow$ 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 ?

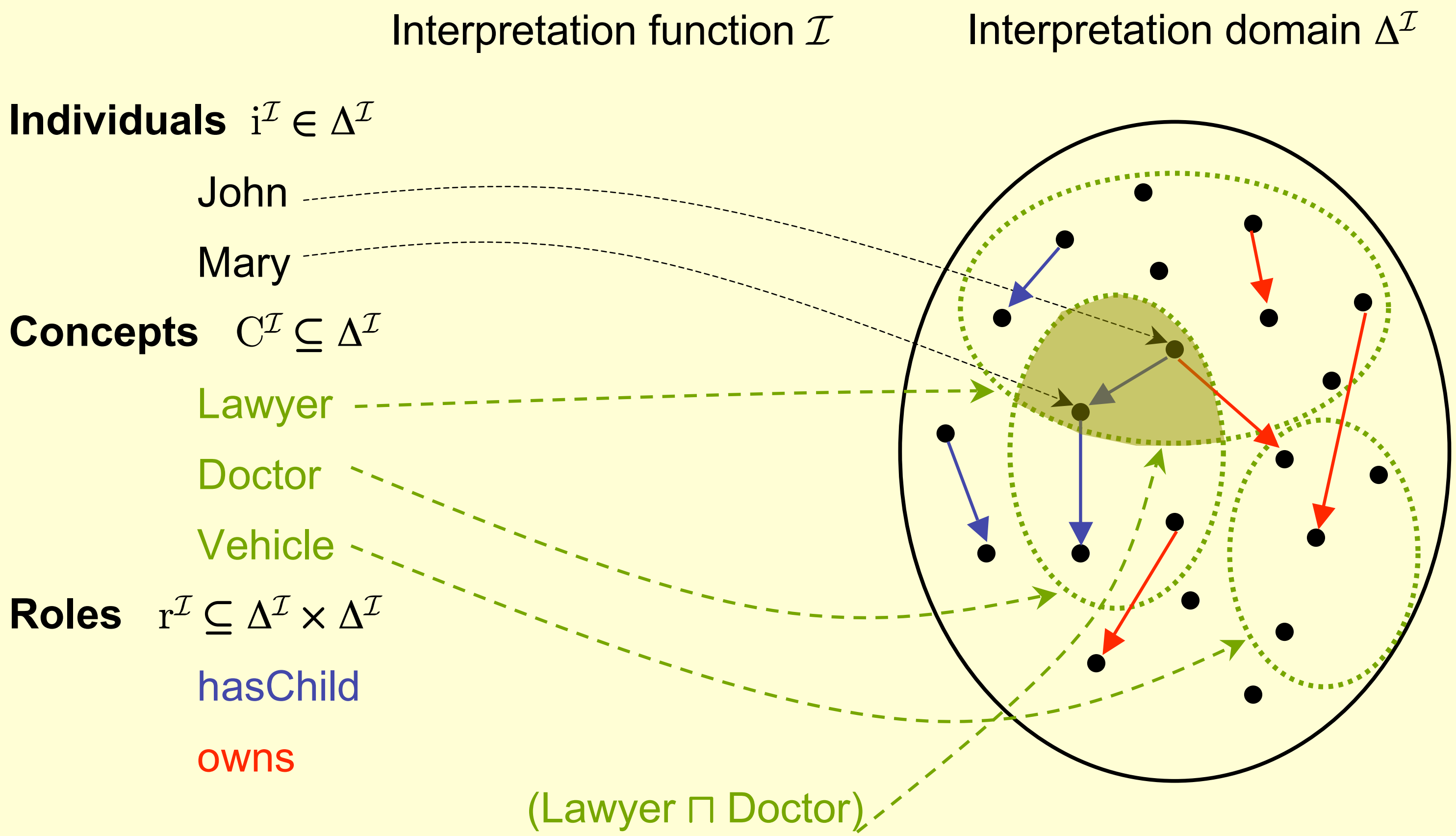
- It is too powerful:
  The problem of deciding whether a formula is logically implied by a theory is **undecidable**, i.e., there is no algorithm that solves the problem in a finite number of steps (*a problem property* !).
- It is too poor:
  A lot of language constructs convenient for knowledge representation 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.

DL Semantics

given by standard model theory:



G. Görz, FAU, Inf. 8

DL Knowledge Bases

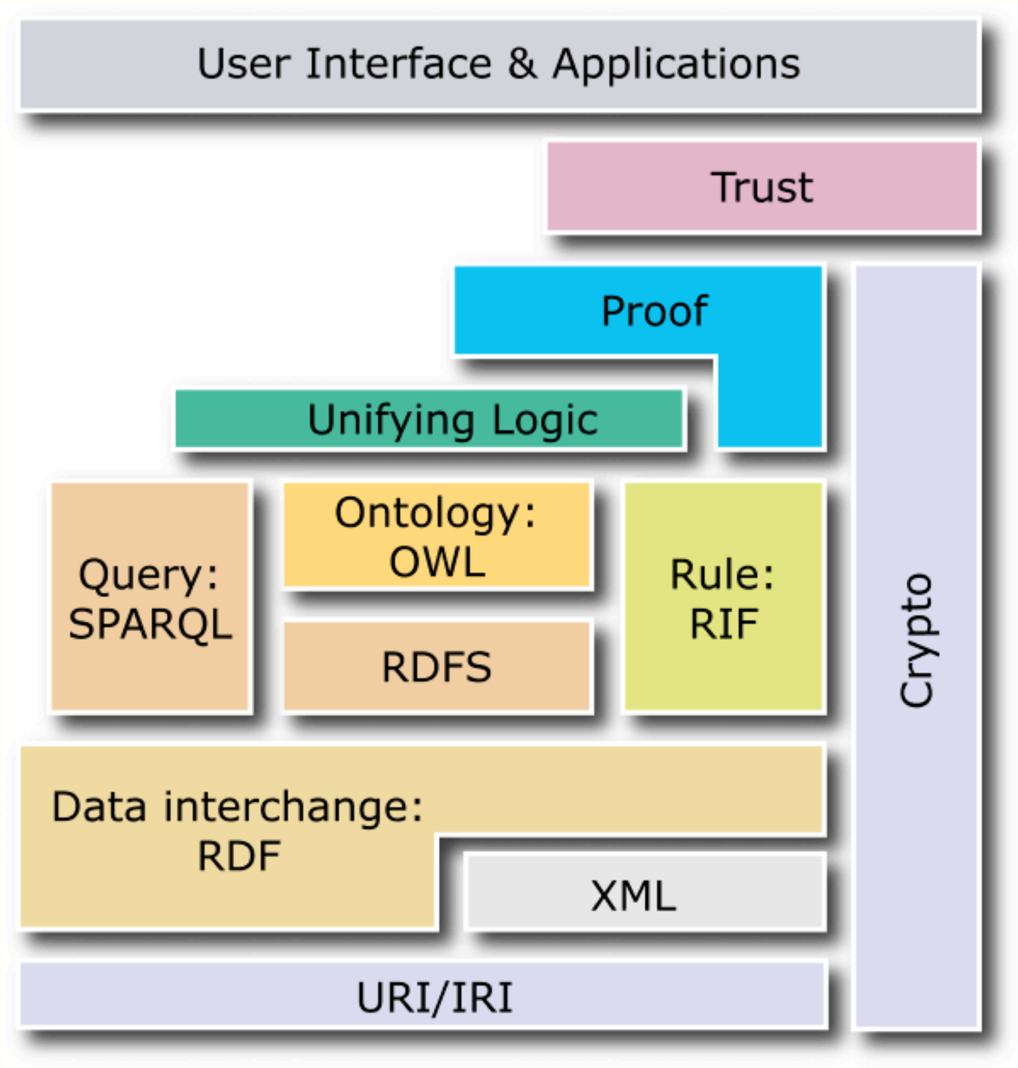
Separation into

- **T-Box**: conceptual (**T**erminological) knowledge
  - Concepts (classes), properties, constraints
  - Inheritance hierarchy
- **A-Box**: knowledge about individuals, i.e. concept instances (**A**ssertional)

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, …)

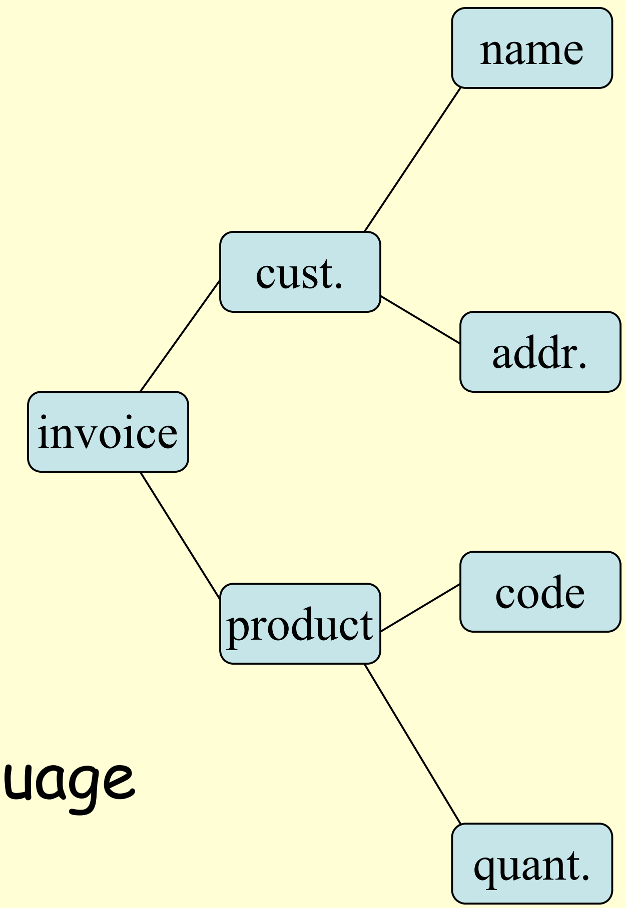
„Semantic Web": Language Layers



⇐ OWL-DL

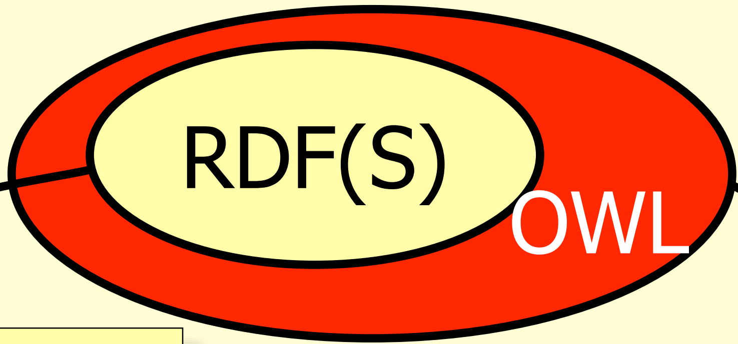
XML : Data Representation

```xml
<?xml version="1.0" encoding="UTF-8"?>
<invoice>
    <customer>
        <name>Carl Lagoze</name>
        <address>Ithaca</address>
    </customer>
    <product>
        <code>x022</code>
        <quantity>2</quantity>
    </product>
</invoice>
```



XML : eXtended Markup Language
- Metalanguage
- Grammar / Schema
  ⇝ special Markup Language

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

G.

OWL as RDF(S) extension (2/2)

```
<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>
```

OWL as RDF(S) extension (2/2)

```
<rdfs:Class rdf:ID="herbivore">


   <rdfs:subClassOf rdf:resource="#animal"/>
   <rdfs:subClassOf>




   </rdfs:subClassOf>
</rdfs:Class>
```

OWL DL: Concept Expressions

Constructor	DL Syntax	Example	FOL Syntax
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human ⊓ Male	$C_1(x) \wedge \ldots \wedge C_n(x)$
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor ⊔ Lawyer	$C_1(x) \vee \ldots \vee C_n(x)$
complementOf	$\neg C$	¬Male	$\neg C(x)$
oneOf	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	{john} ⊔ {mary}	$x = x_1 \vee \ldots \vee x = x_n$
allValuesFrom	$\forall P.C$	∀hasChild.Doctor	$\forall y.P(x,y) \rightarrow C(y)$
someValuesFrom	$\exists P.C$	∃hasChild.Lawyer	$\exists y.P(x,y) \wedge C(y)$
maxCardinality	$\leqslant nP$	⩽1hasChild	$\exists^{\leqslant n} y.P(x,y)$
minCardinality	$\geqslant nP$	⩾2hasChild	$\exists^{\geqslant n} y.P(x,y)$

- C is a concept (class); P is a role (property); x is an individual name
- XMLS data types and also classes in ∀P.C and ∃P.C
  - e.g. ∃hasAge.nonNegativeInteger
- Unlimited nesting of constructors
  - e.g., Person ⊓ ∀hasChild.(Doctor ⊔ ∃hasChild.Doctor)

OWL DL: Axioms

gelfond_lifschitz.StableModelSemantics.ps

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
sameClassAs	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} $\equiv$ {G_W_Bush}
differentIndividualFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq \neg${peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
samePropertyAs	$P_1 \equiv P_2$	cost $\equiv$ price
inverseOf	$P_1 \equiv P_2^-$	hasChild $\equiv$ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+$ $\sqsubseteq$ ancestor
uniqueProperty	$\top \sqsubseteq \leqslant 1 P$	$\top \sqsubseteq \leqslant 1$hasMother
unambiguousProperty	$\top \sqsubseteq \leqslant 1 P^-$	$\top \sqsubseteq \leqslant 1$hasSSN$^-$

- ☞ $\mathcal{I}$ **satisfies** $C_1 \sqsubseteq C_2$ iff $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$; satisfies $P_1 \sqsubseteq P_2$ iff $P_1^{\mathcal{I}} \subseteq P_2^{\mathcal{I}}$
- ☞ $\mathcal{I}$ satisfies ontology $\mathcal{O}$ (is a **model** of $\mathcal{O}$) iff satisfies every axiom in $\mathcal{O}$

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

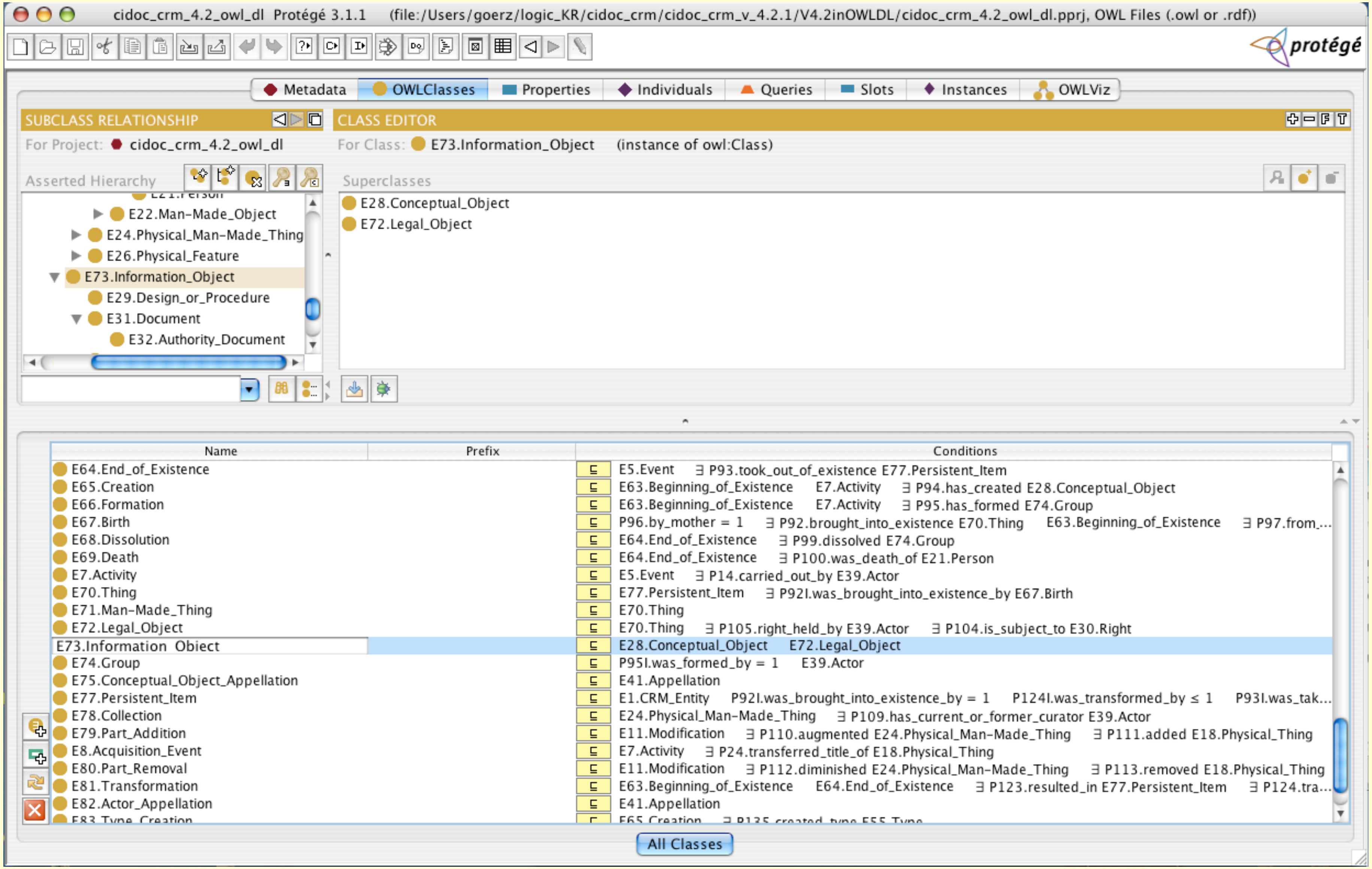
Principles of the Implementation

- 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.

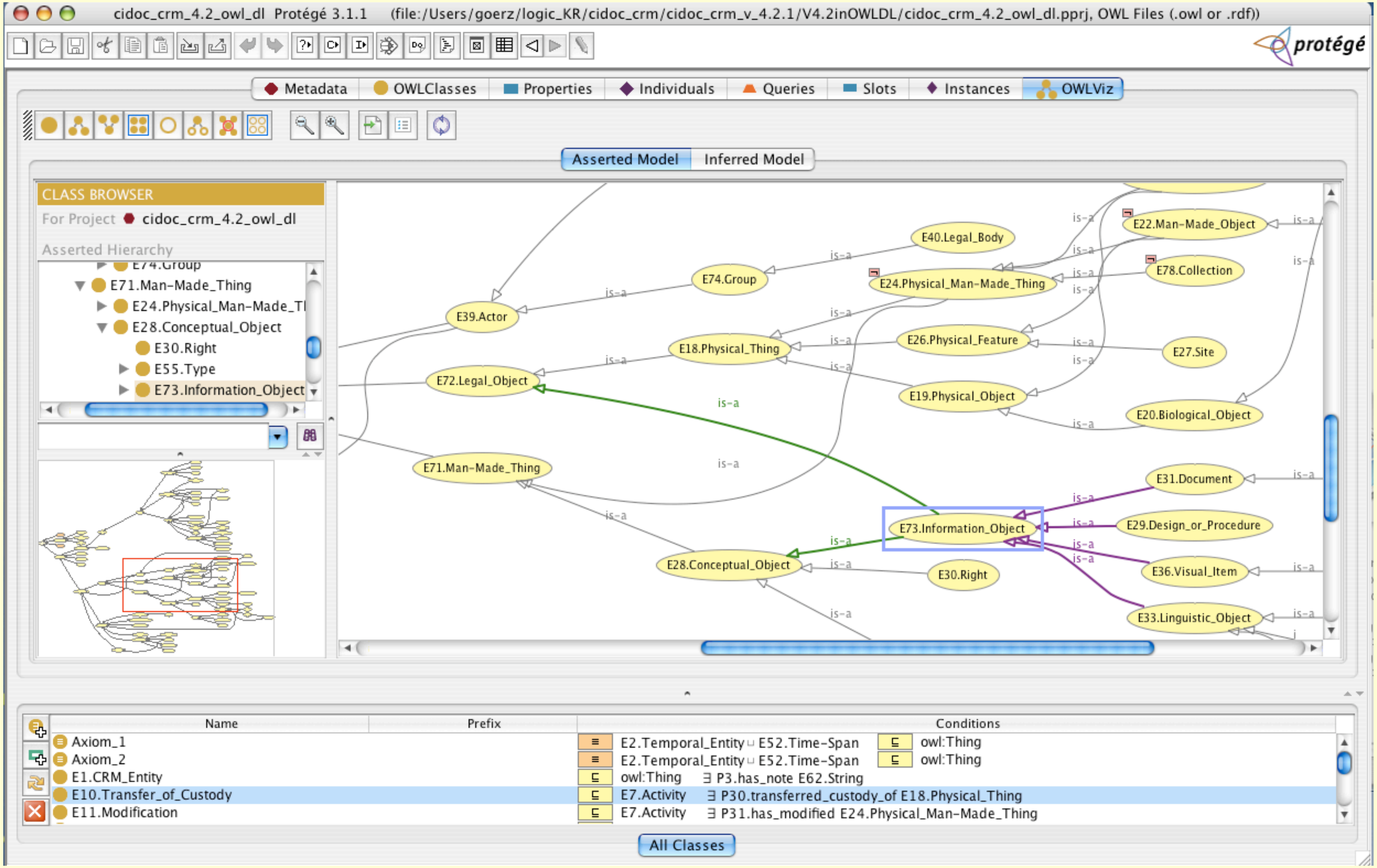
Tools

- 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



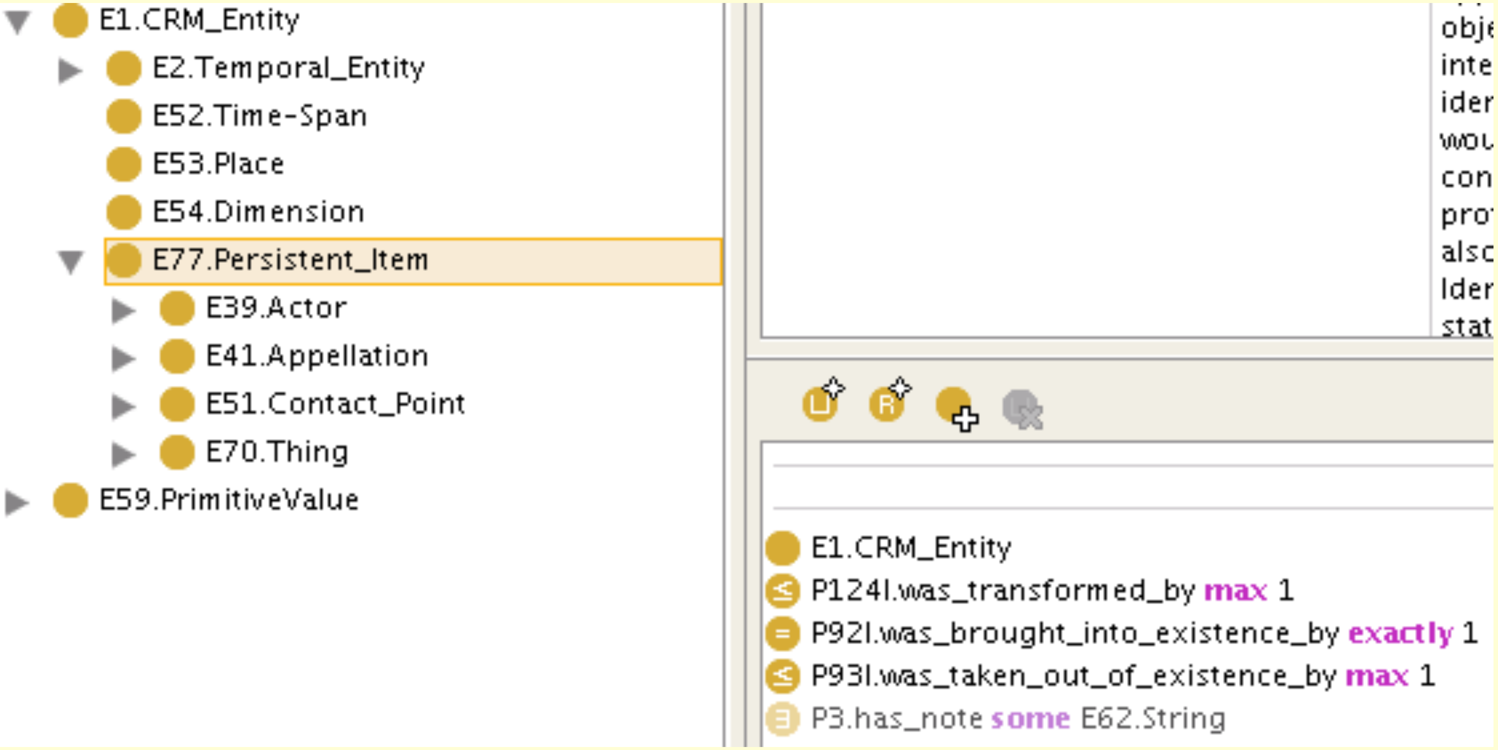
Ontology Editor Protégé: CRM in OWL-DL



Specification of E77.Persistent_Item

- Subclass of E1.CRM_Entity
- Superclass of E39.Actor, E41.Appellation, E51.Contact_Point, E70.Thing
- 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



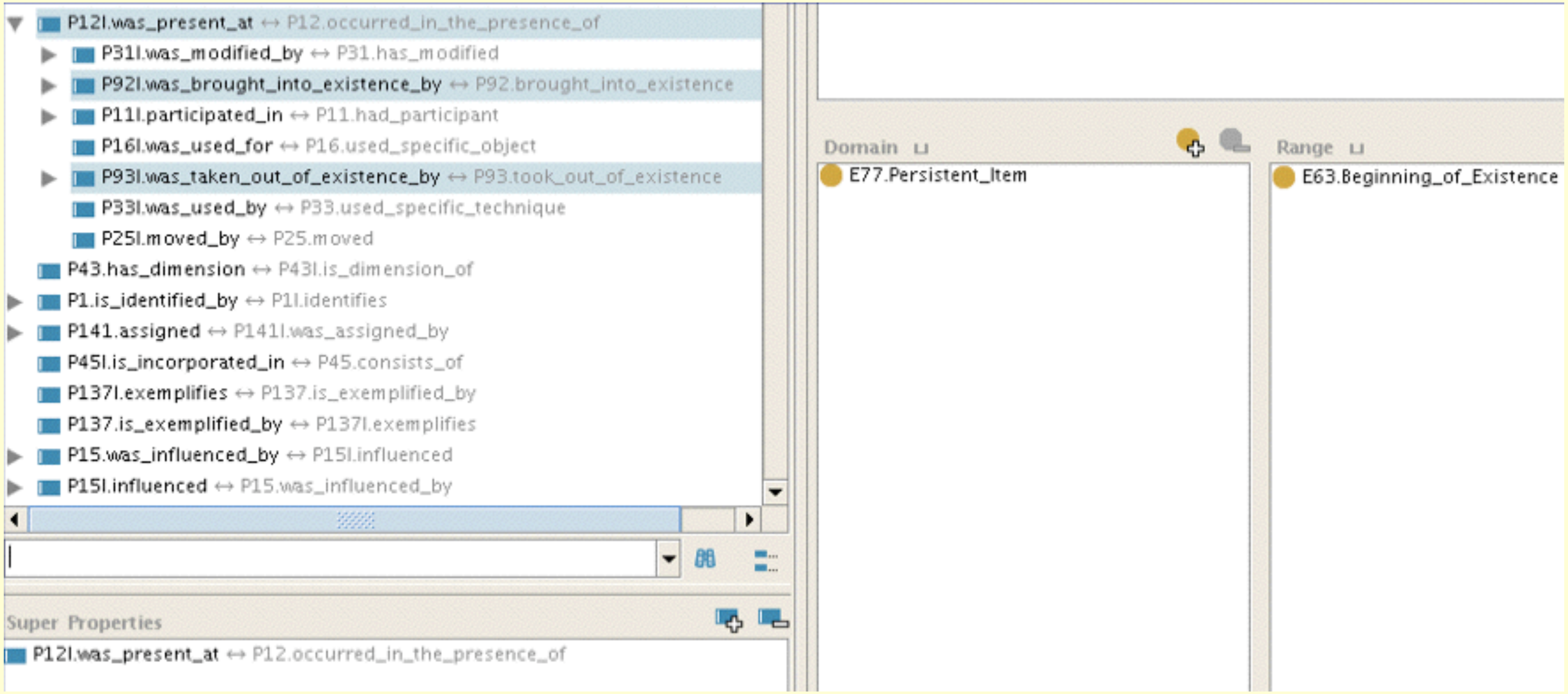
E77.Persistent_Item in OWL-DL (1)

```
<owl:Class rdf:about="#E77.Persistent_Item">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about=
        "#P92I.was_brought_into_existence_by"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class rdf:about=
        "#E63.Beginning_of_Existence"/>
      </owl:someValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
```

E77.Persistent_Item in OWL-DL (2)

```
        <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



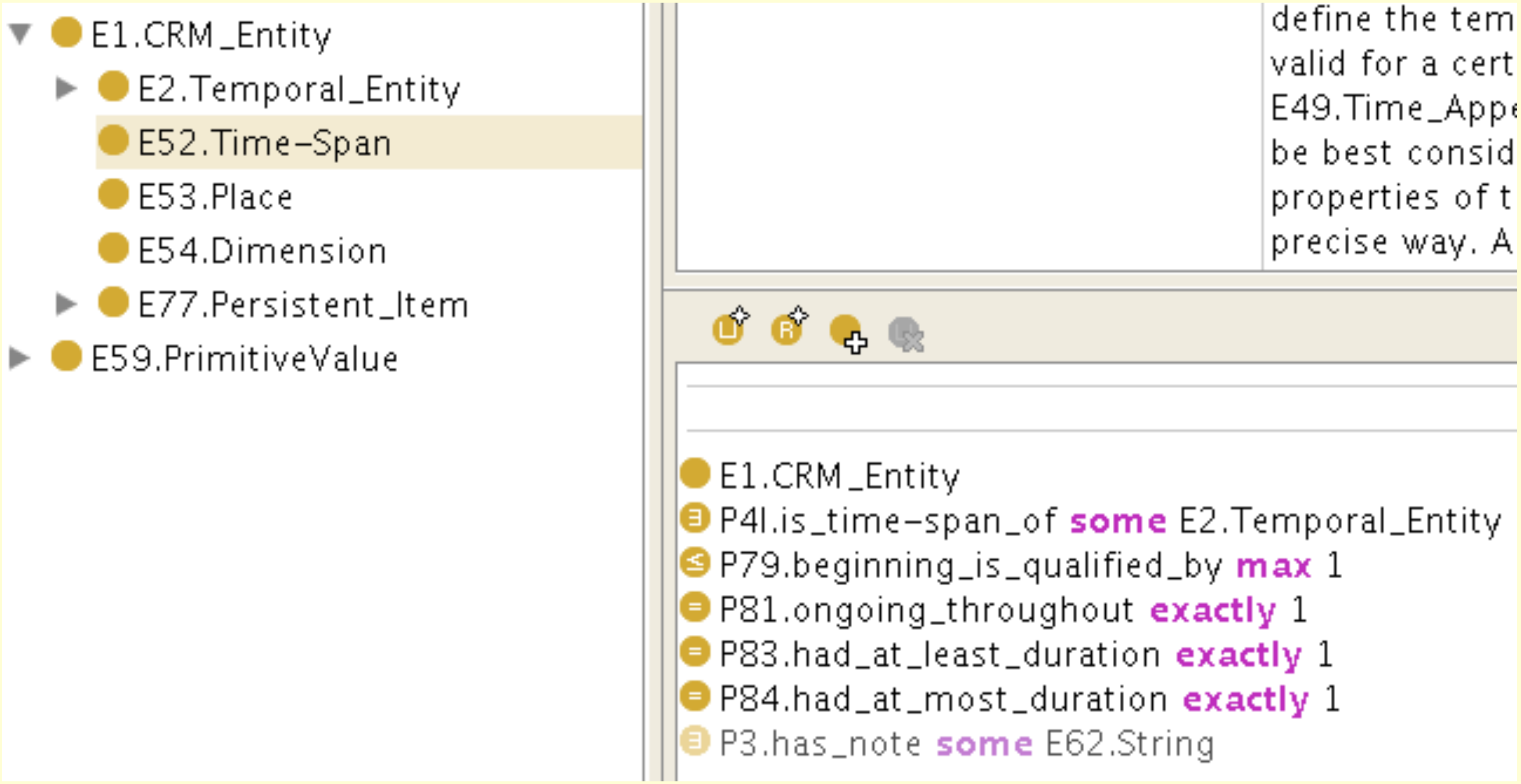
P92I.was_brought_into_existence_by in OWL-DL

```
<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>
```

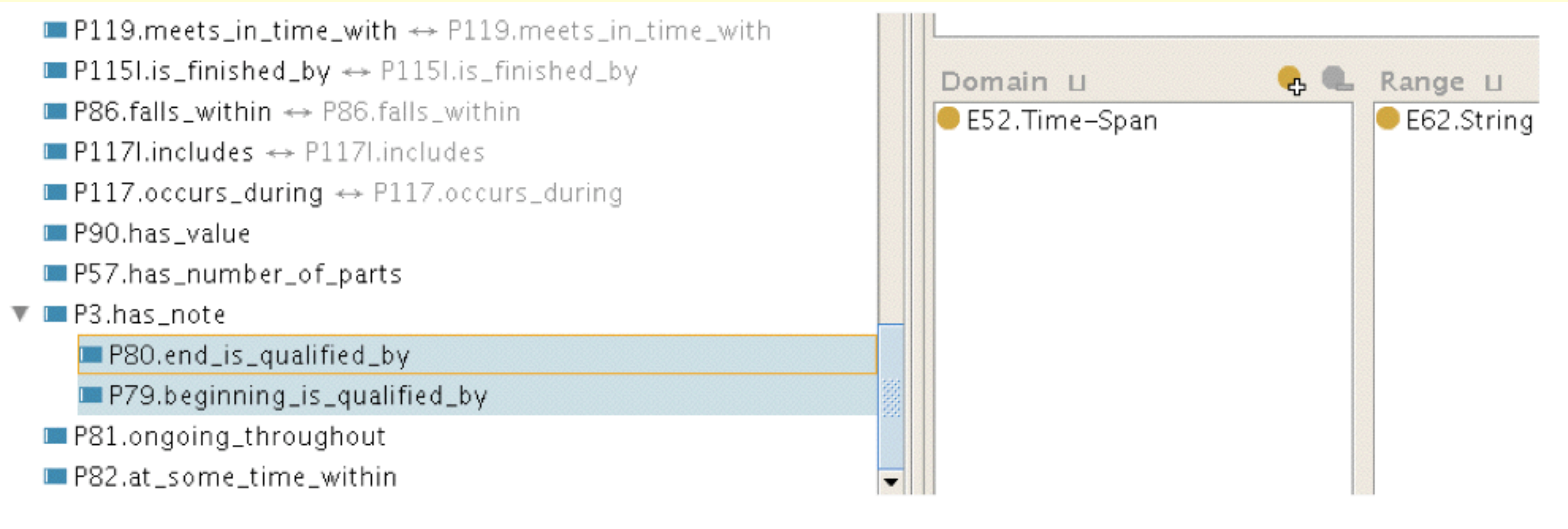
Datatypes in CRM

- Used for representation of strings, etc.
- Datatype properties are not permitted as inverse-functional properties (as opposed to object properties) ⇒ 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

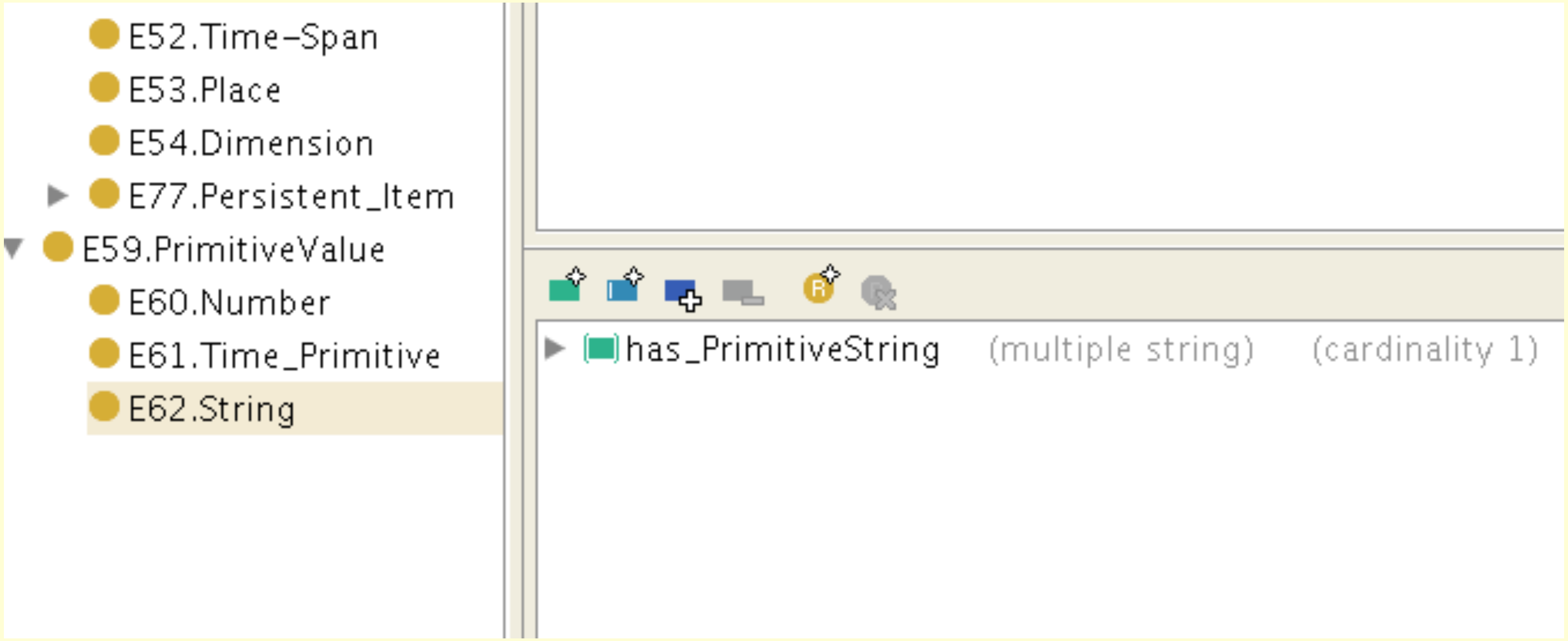
E52.Time_Span



Properties of E52.Time_Span



E62.String


E52.Time-Span
E53.Place
E54.Dimension
E77.Persistent_Item
E59.PrimitiveValue
E60.Number
E61.Time_Primitive
E62.String
has_PrimitiveString (multiple string) (cardinality 1)

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*.

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 !

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.

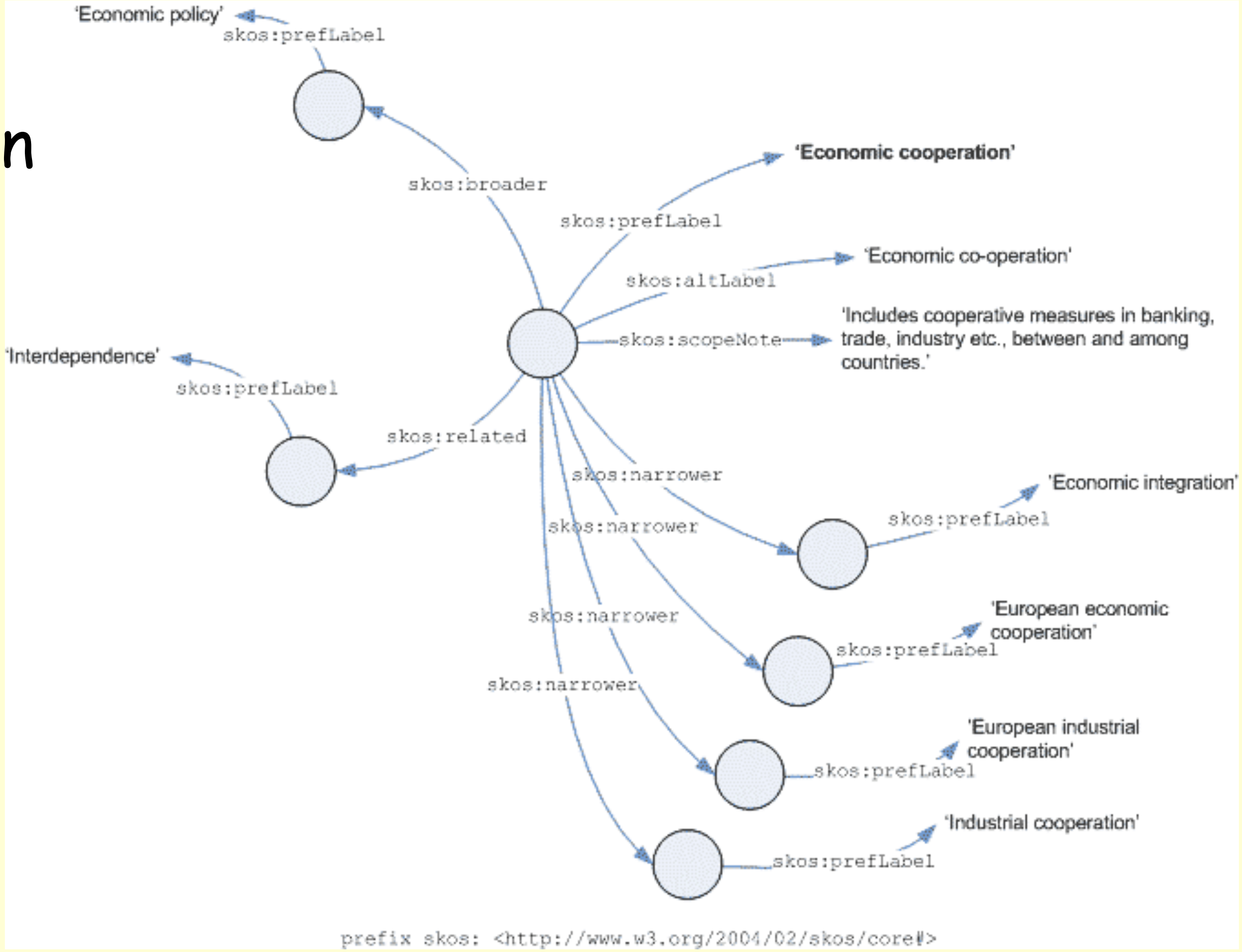
    ex:PaintingConcept rdf:type skos:Concept.

    ex:PaintingClass ex:correspondingConcept ex:PaintingConcept.

SKOS ("Simple Knowledge Organization System")

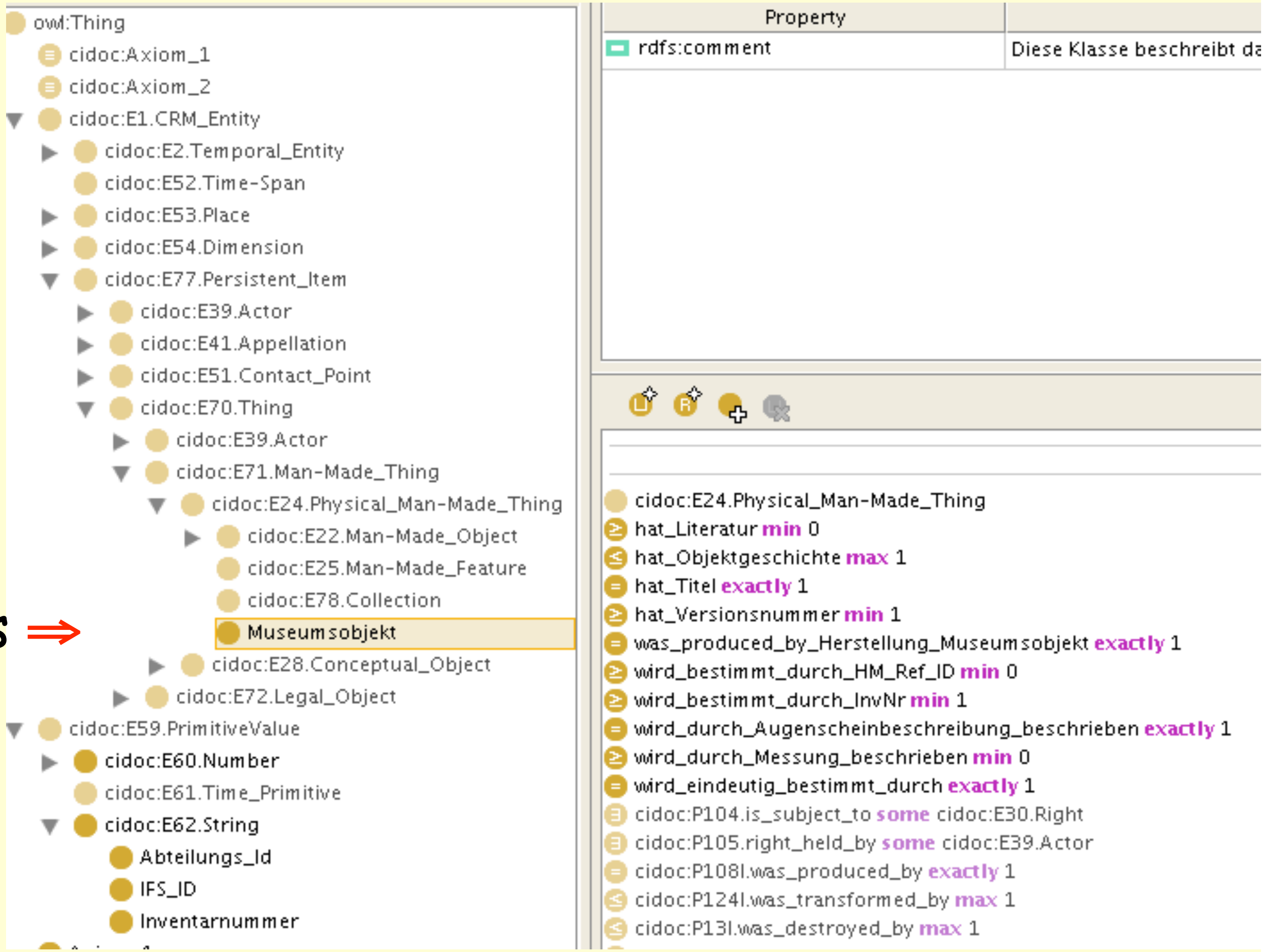
Representation language for thesauri etc. (W3C),

based on RDF



The GNM-DMS Ontology as an Extension to the CRM

interface class ⇒



http://www8.informatik.uni-erlangen.de/IMMD8/Services/cidoc-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 Görz 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 Görz
  Short Documentation of the CIDOC CRM (4.2.4) Implementation in OWL-DL
  (46 KB, PDF)


SIGILLVM ACADEMIAE FRIDERICO ALEXANDRINAE