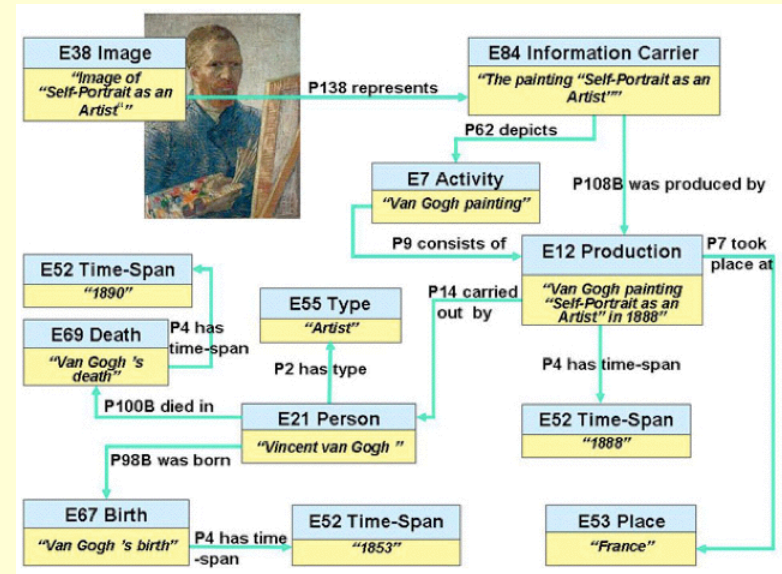


# An Implementation of the CIDOC CRM in OWL-DL



© Doerr

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

G. Görz, FAU, Inf. 8

# 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"

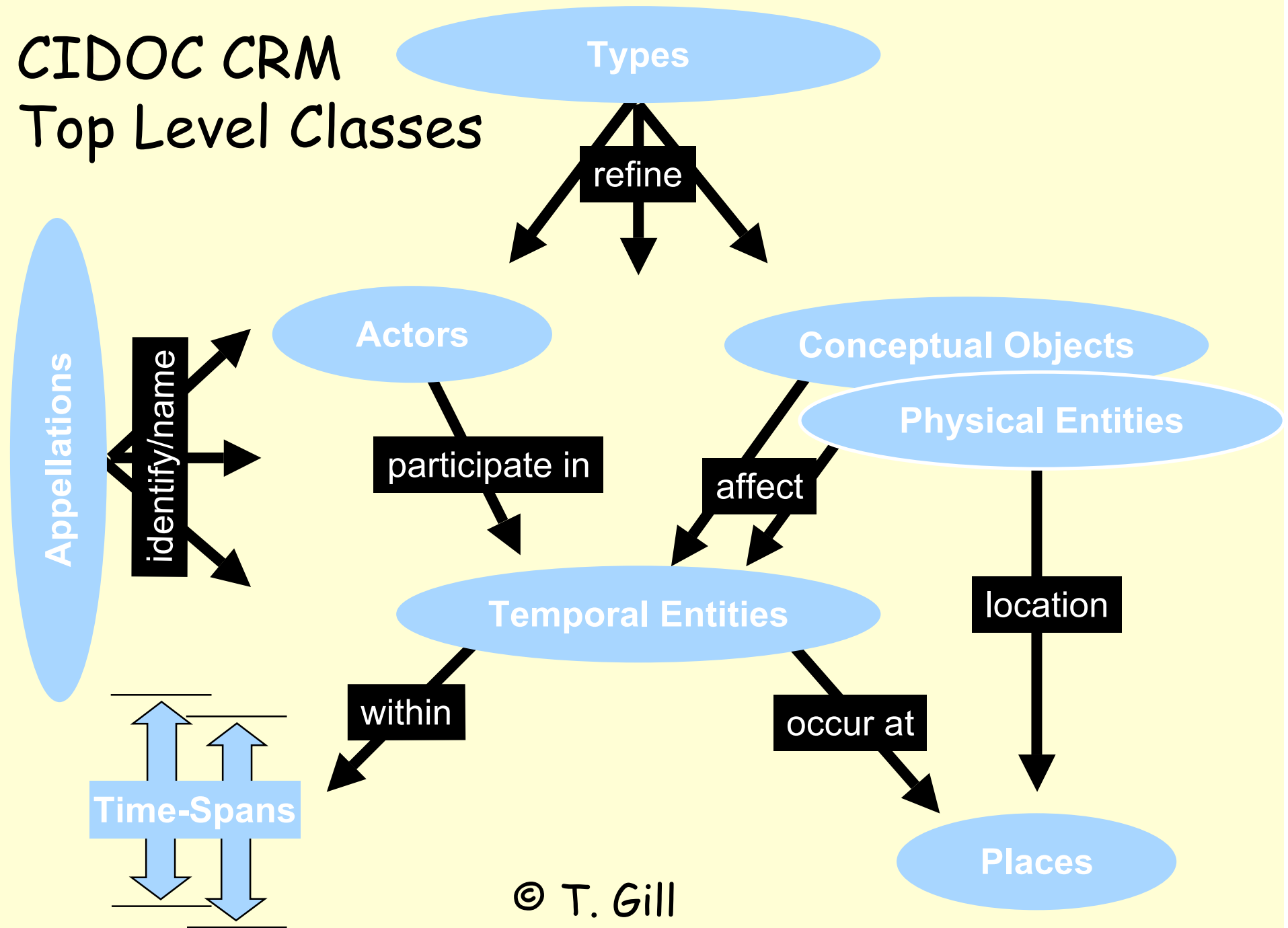
# Knowledge Modelling

- “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  
⇒ 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:

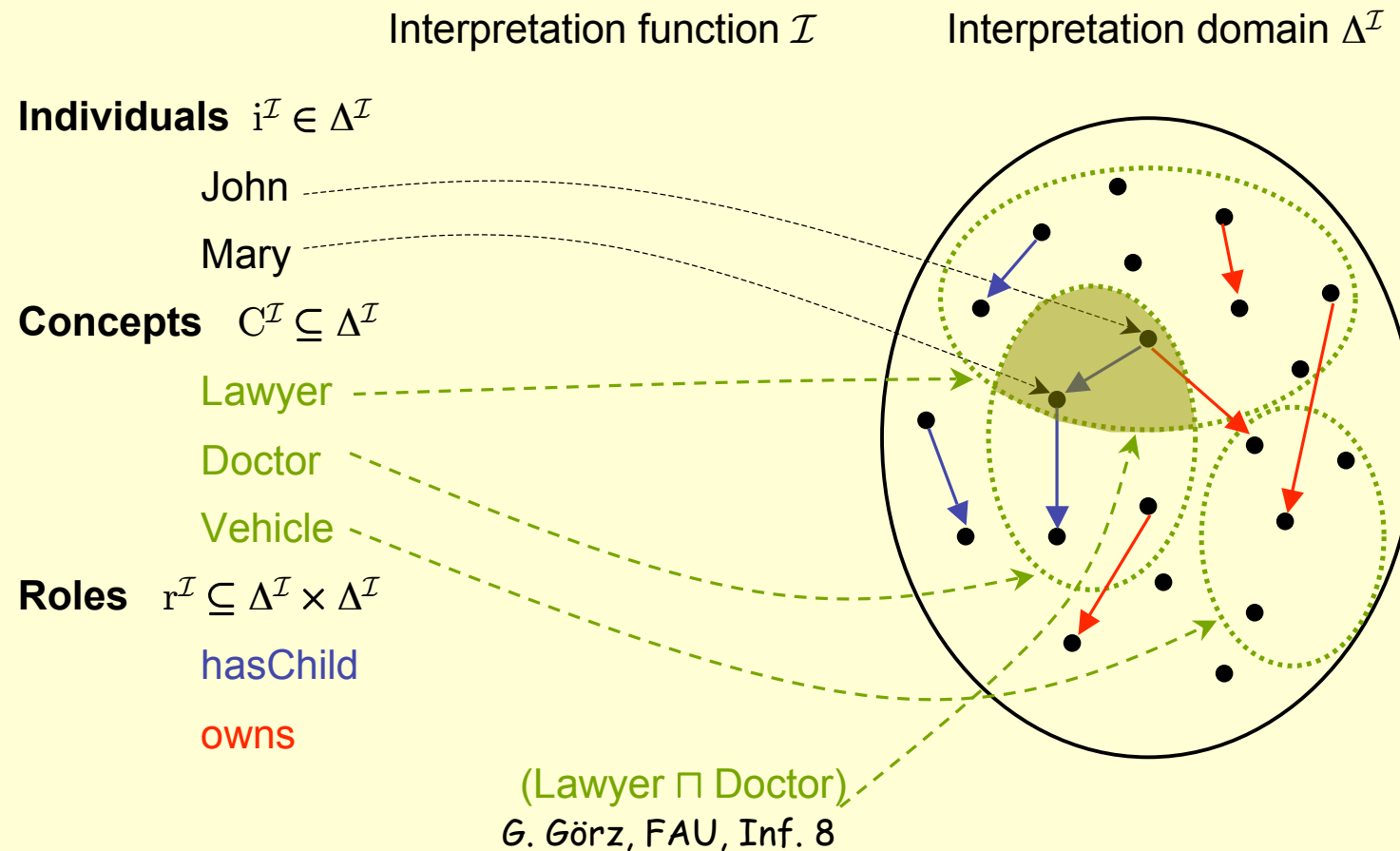
A lot of language constructs convenient for knowledge representation are missing.

# 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:



# DL Knowledge Bases

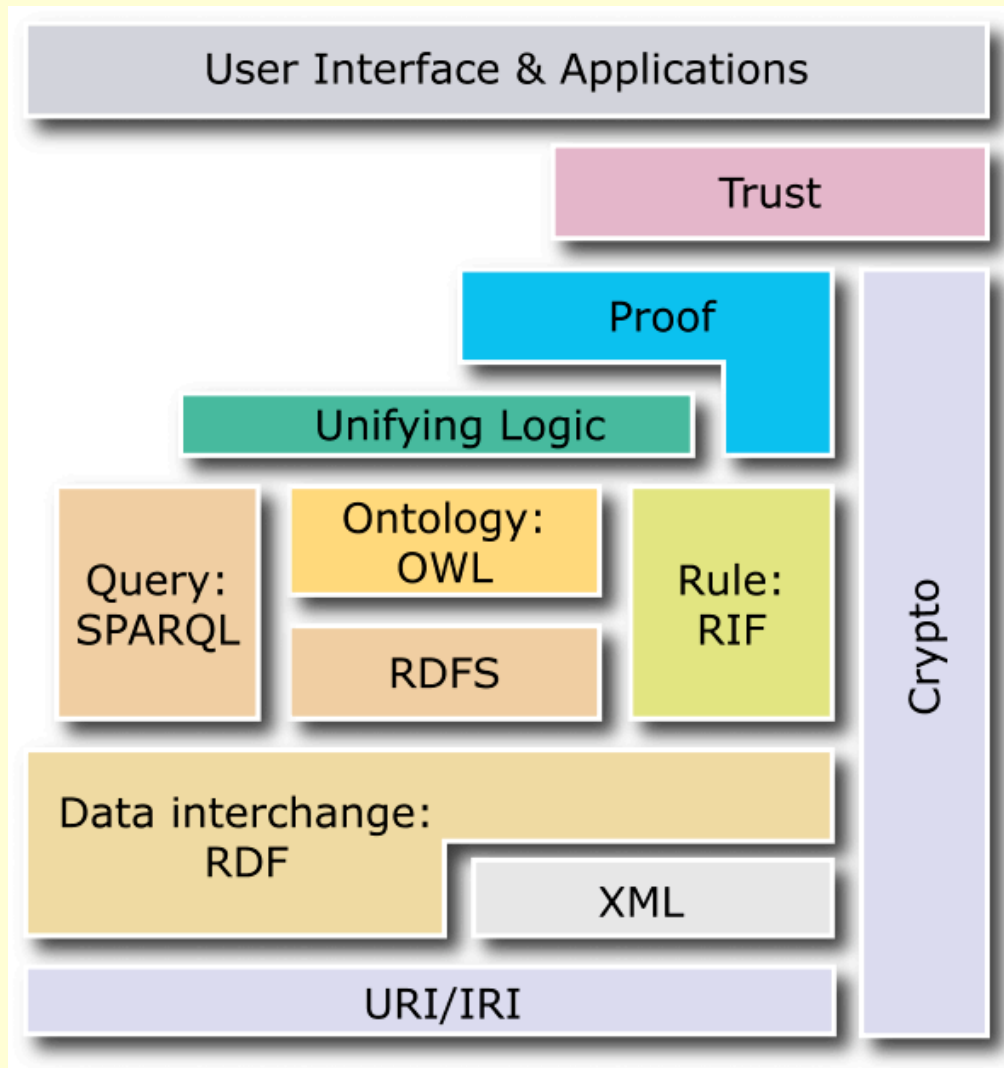
Separation into

- **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, ...)

# „Semantic Web“: Language Layers

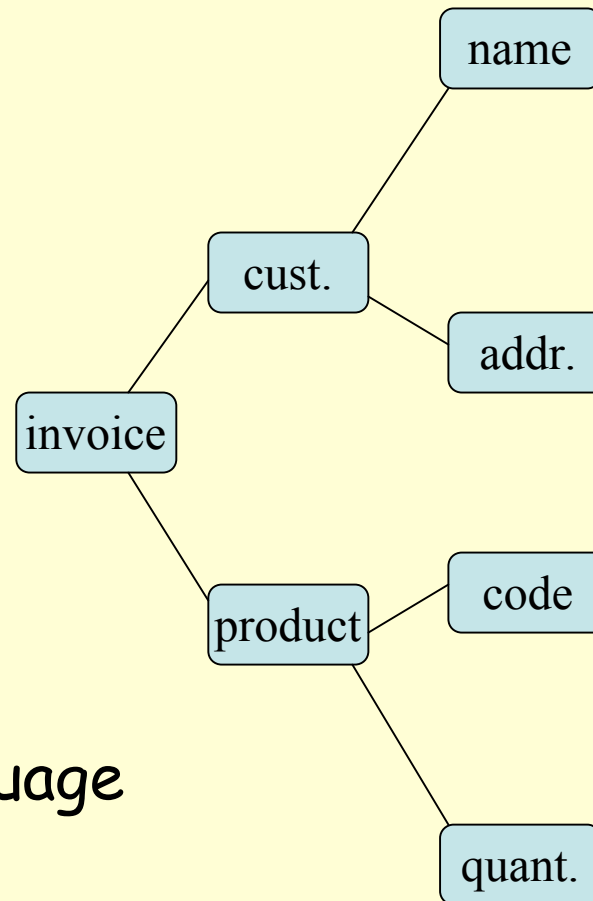


← OWL-DL



# XML : Data Representation

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



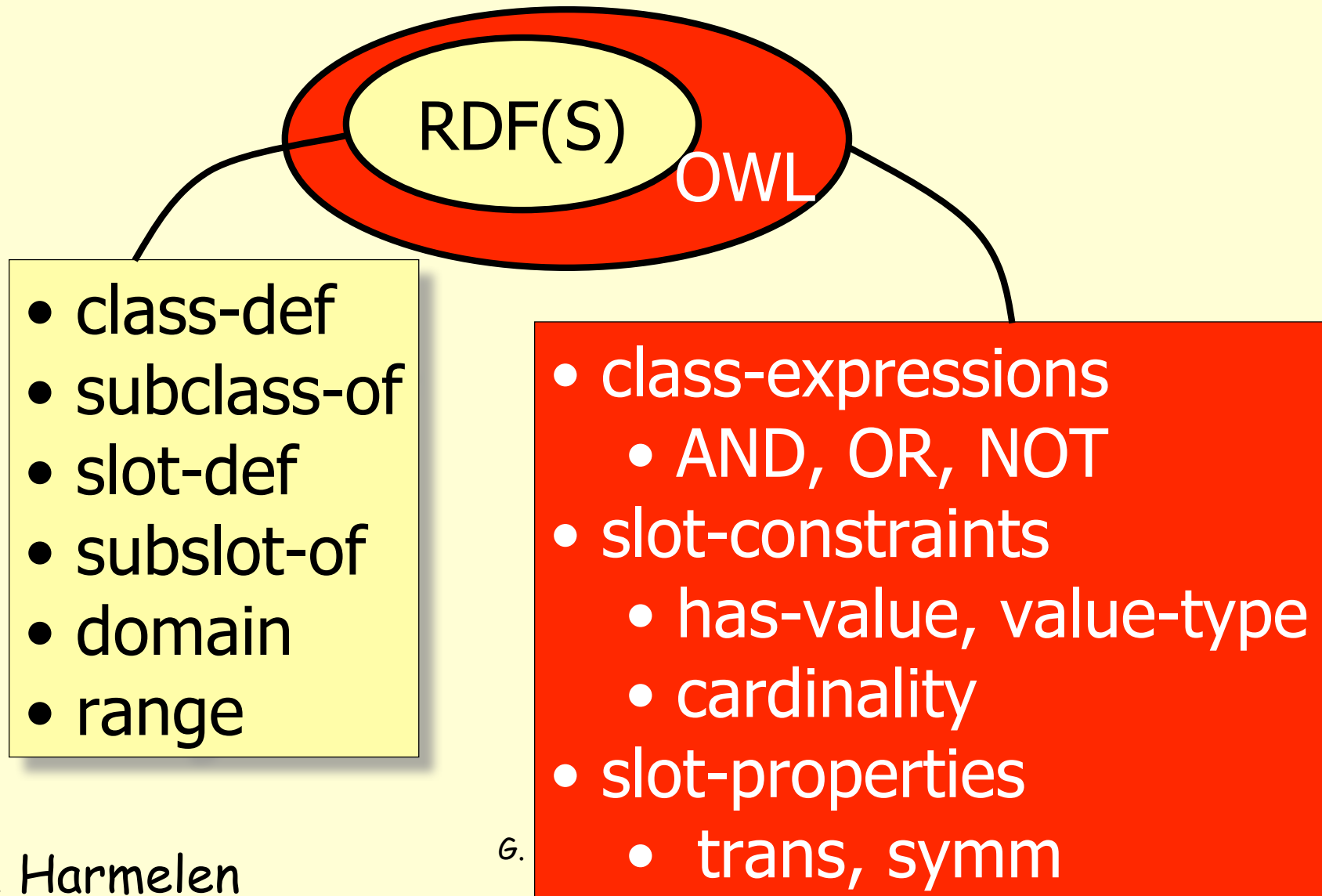
**XML** : eXtended Markup Language

- Metalanguage

- Grammar / Schema

↪ special Markup Language

# OWL as RDF(S) extension (1/2)



# 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	$C_1 \sqcap \dots \sqcap C_n$	Human $\sqcap$ Male	$C_1(x) \wedge \dots \wedge C_n(x)$
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor $\sqcup$ Lawyer	$C_1(x) \vee \dots \vee C_n(x)$
complementOf	$\neg C$	$\neg$ Male	$\neg C(x)$
oneOf	$\{x_1\} \sqcup \dots \sqcup \{x_n\}$	{john} $\sqcup$ {mary}	$x = x_1 \vee \dots \vee x = x_n$
allValuesFrom	$\forall P.C$	$\forall$ hasChild.Doctor	$\forall y.P(x, y) \rightarrow C(y)$
someValuesFrom	$\exists P.C$	$\exists$ hasChild.Lawyer	$\exists y.P(x, y) \wedge C(y)$
maxCardinality	$\leq_n P$	$\leq 1$ hasChild	$\exists \leq_n y.P(x, y)$
minCardinality	$\geq_n P$	$\geq 2$ hasChild	$\exists \geq_n y.P(x, y)$

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

# OWL DL: Axioms

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 <sup>-</sup>
transitiveProperty	$P^+ \sqsubseteq P$	ancestor <sup>+</sup> $\sqsubseteq$ ancestor
uniqueProperty	$\top \sqsubseteq \leq 1P$	$\top \sqsubseteq \leq 1$ hasMother
unambiguousProperty	$\top \sqsubseteq \leq 1P^-$	$\top \sqsubseteq \leq 1$ hasSSN <sup>-</sup>

☞  $\mathcal{I}$  **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

The screenshot displays the Protégé 3.1.1 interface for editing the ontology 'cidoc\_crm\_4.2\_owl\_dl'. The 'CLASS EDITOR' is active for the class 'E73.Information\_Object'. The 'Asserted Hierarchy' shows a tree structure with 'E73.Information\_Object' selected, which includes subclasses like 'E29.Design\_or\_Procedure', 'E31.Document', and 'E32.Authority\_Document'. The 'Superclasses' list includes 'E28.Conceptual\_Object' and 'E72.Legal\_Object'. The main table lists various classes and their associated conditions.

Name	Prefix	Conditions
E64.End_of_Existence		E5.Event $\exists$ P93.took_out_of_existence E77.Persistent_Item
E65.Creation		E63.Beginning_of_Existence E7.Activity $\exists$ P94.has_created E28.Conceptual_Object
E66.Formation		E63.Beginning_of_Existence E7.Activity $\exists$ P95.has_formed E74.Group
E67.Birth		P96.by_mother = 1 $\exists$ P92.brought_into_existence E70.Thing E63.Beginning_of_Existence $\exists$ P97.from...
E68.Dissolution		E64.End_of_Existence $\exists$ P99.dissolved E74.Group
E69.Death		E64.End_of_Existence $\exists$ P100.was_death_of E21.Person
E7.Activity		E5.Event $\exists$ P14.carried_out_by E39.Actor
E70.Thing		E77.Persistent_Item $\exists$ P921.was_brought_into_existence_by E67.Birth
E71.Man-Made_Thing		E70.Thing
E72.Legal_Object		E70.Thing $\exists$ P105.right_held_by E39.Actor $\exists$ P104.is_subject_to E30.Right
E73.Information_Object		E28.Conceptual_Object E72.Legal_Object
E74.Group		P951.was_formed_by = 1 E39.Actor
E75.Conceptual_Object_Appellation		E41.Appellation
E77.Persistent_Item		E1.CRM_Entity P921.was_brought_into_existence_by = 1 P1241.was_transformed_by $\leq$ 1 P931.was_tak...
E78.Collection		E24.Physical_Man-Made_Thing $\exists$ P109.has_current_or_former_curator E39.Actor
E79.Part_Addition		E11.Modification $\exists$ P110.augmented E24.Physical_Man-Made_Thing $\exists$ P111.added E18.Physical_Thing
E8.Acquisition_Event		E7.Activity $\exists$ P24.transferred_title_of E18.Physical_Thing
E80.Part_Removal		E11.Modification $\exists$ P112.diminished E24.Physical_Man-Made_Thing $\exists$ P113.removed E18.Physical_Thing
E81.Transformation		E63.Beginning_of_Existence E64.End_of_Existence $\exists$ P123.resulted_in E77.Persistent_Item $\exists$ P124.tra...
E82.Actor_Appellation		E41.Appellation
E83.Type_Creation		E65.Creation $\exists$ P125.created_type E55.Type

An 'All Classes' button is located at the bottom center of the table.

# Ontology Editor Protégé: CRM in OWL-DL

cidoc\_crm\_4.2\_owl\_dl Protégé 3.1.1 (file:/Users/goerz/logic\_KR/cidoc\_crm/cidoc\_crm\_v\_4.2.1/V4.2inOWLDL/cidoc\_crm\_4.2\_owl\_dl.pprj, OWL Files (.owl or .rdf))

Metadata OWLClasses Properties Individuals Queries Slots Instances OWLViz

CLASS BROWSER  
For Project cidoc\_crm\_4.2\_owl\_dl  
Asserted Hierarchy

- E74.Group
  - E71.Man-Made\_Thing
    - E24.Physical\_Man-Made\_Thing
      - E28.Conceptual\_Object
        - E30.Right
        - E55.Type
        - E73.Information\_Object

Name	Prefix	Conditions
Axiom_1	E2.Temporal_Entity ⊆ E52.Time-Span	owl:Thing
Axiom_2	E2.Temporal_Entity ⊆ E52.Time-Span	owl:Thing
E1.CRM_Entity	owl:Thing	∃ P3.has_note E62.String
E10.Transfer_of_Custody	E7.Activity	∃ P30.transferred_custody_of E18.Physical_Thing
E11.Modification	E7.Activity	∃ P31.has_modified E24.Physical_Man-Made_Thing

All Classes

# 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

The screenshot displays a software interface with a class hierarchy on the left and a list of properties for the selected class on the right.

**Class Hierarchy (Left):**

- ▼ E1.CRM\_Entity
  - ▶ E2.Temporal\_Entity
    - E52.Time-Span
    - E53.Place
    - E54.Dimension
    - ▼ E77.Persistent\_Item (highlighted)
    - ▶ E39.Actor
    - ▶ E41.Appellation
    - ▶ E51.Contact\_Point
    - ▶ E70.Thing
  - ▶ E59.PrimitiveValue

**Properties List (Right):**

- E1.CRM\_Entity
- Ⓜ P124I.was\_transformed\_by **max** 1
- Ⓜ P92I.was\_brought\_into\_existence\_by **exactly** 1
- Ⓜ P93I.was\_taken\_out\_of\_existence\_by **max** 1
- Ⓜ P3.has\_note **some** E62.String

Additional visible text on the right side of the interface includes: "obje", "inte", "ider", "wou", "con", "pro", "also", "lder", "stat".

# 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

The screenshot displays a software interface with a list of properties on the left and a detailed view of a property on the right.

**Property List (Left Panel):**

- ▼ P12I.was\_present\_at ↔ P12.occurred\_in\_the\_presence\_of
  - ▶ P31I.was\_modified\_by ↔ P31.has\_modified
  - ▶ P92I.was\_brought\_into\_existence\_by ↔ P92.brought\_into\_existence
  - ▶ P11I.participated\_in ↔ P11.had\_participant
    - ▶ P16I.was\_used\_for ↔ P16.used\_specific\_object
  - ▶ P93I.was\_taken\_out\_of\_existence\_by ↔ P93.took\_out\_of\_existence
    - ▶ P33I.was\_used\_by ↔ P33.used\_specific\_technique
    - ▶ P25I.moved\_by ↔ P25.moved
  - ▶ P43.has\_dimension ↔ P43I.is\_dimension\_of
  - ▶ P1.is\_identified\_by ↔ P1I.identifies
  - ▶ P141.assigned ↔ P141I.was\_assigned\_by
  - ▶ P45I.is\_incorporated\_in ↔ P45.consists\_of
  - ▶ P137I.exemplifies ↔ P137.is\_exemplified\_by
  - ▶ P137.is\_exemplified\_by ↔ P137I.exemplifies
  - ▶ P15.was\_influenced\_by ↔ P15I.influenced
  - ▶ P15I.influenced ↔ P15.was\_influenced\_by

**Property Detail (Right Panel):**

The right panel shows the domain and range for the selected property:

Domain	Range
E77.Persistent_Item	E63.Beginning_of_Existence

Below the domain and range panels, there is a section for "Super Properties" which lists the same property: P12I.was\_present\_at ↔ P12.occurred\_in\_the\_presence\_of.



# 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

The screenshot shows a software interface with a class hierarchy on the left and a list of properties for the selected class on the right.

**Class Hierarchy (Left):**

- ▼ ● E1.CRM\_Entity
  - ▶ ● E2.Temporal\_Entity
    - E52.Time\_Span
    - E53.Place
    - E54.Dimension
  - ▶ ● E77.Persistent\_Item
  - ▶ ● E59.PrimitiveValue

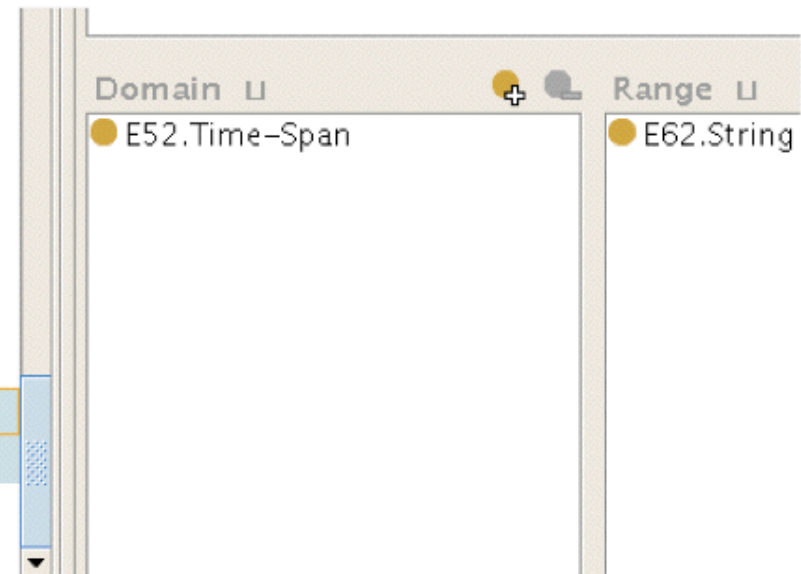
**Properties List (Right):**

- 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

Additional text visible in the top right of the interface: "define the tem... valid for a cert... E49.Time\_Appo... be best consid... properties of t... precise way. A"

# Properties of E52.Time\_Span

- P119.meets\_in\_time\_with ↔ P119.meets\_in\_time\_with
- P115I.is\_finished\_by ↔ P115I.is\_finished\_by
- P86.falls\_within ↔ P86.falls\_within
- P117I.includes ↔ P117I.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 screenshot displays a software interface with a tree view on the left and a detail view on the right. The tree view lists several categories, with 'E62.String' selected. The detail view shows a 'has\_PrimitiveString' property with a value of '(multiple string)' and a cardinality of '(cardinality 1)'.

- 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 !

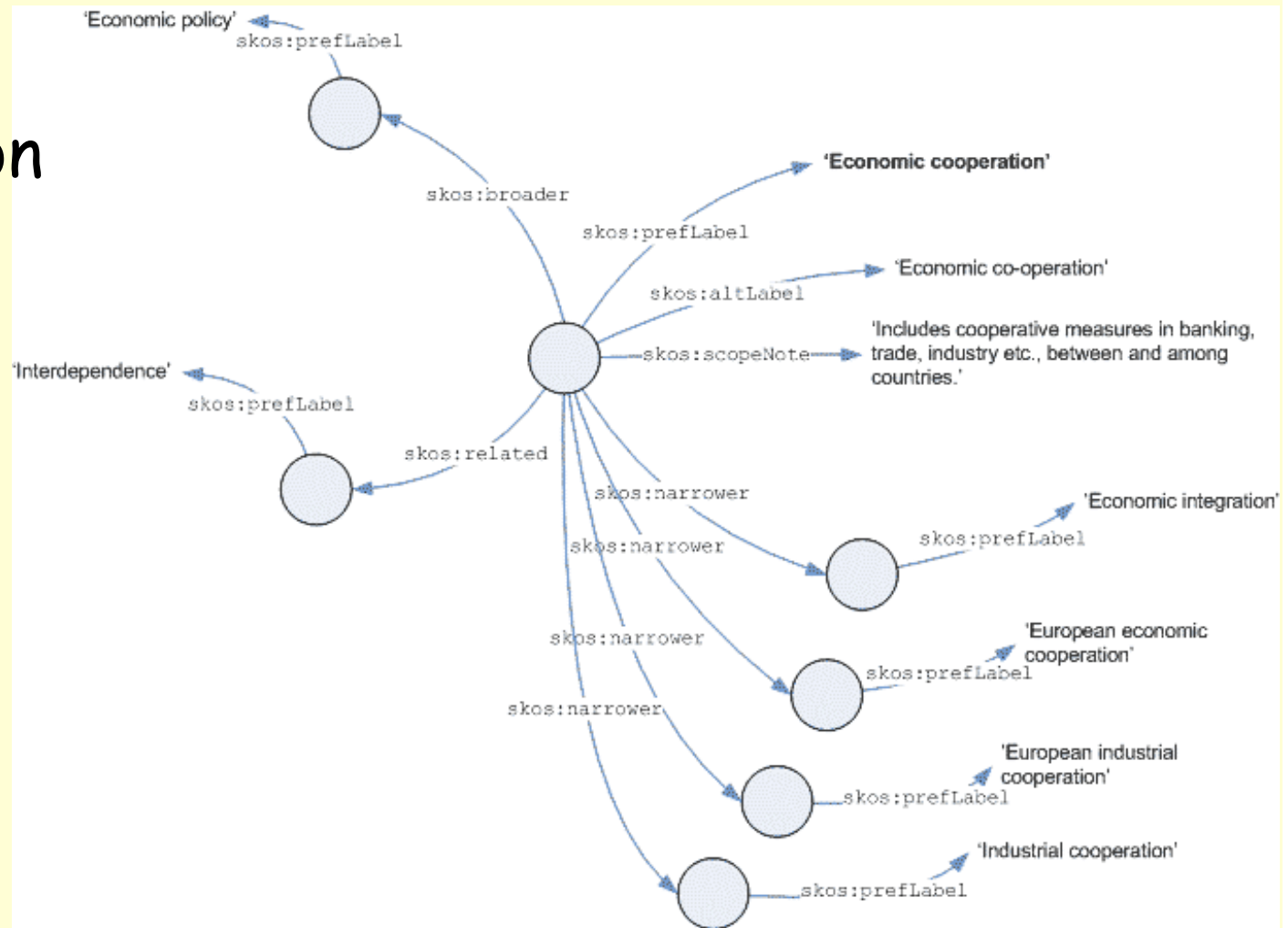
# E55.Type: Attaching Thesauri, etc.

- 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



© Miles et al.

prefix skos: <<http://www.w3.org/2004/02/skos/core#>>

# The GNM-DMS Ontology as an Extension to the CRM

interface class ⇒

The screenshot shows an ontology editor interface. On the left, a tree view displays the class hierarchy. The class `cidoc:E70.Thing` is expanded, and `Museumobjekt` is highlighted with a yellow border. On the right, a table lists the properties of `Museumobjekt`.

| Property                                                    | Description                       |
|-------------------------------------------------------------|-----------------------------------|
| <code>rdfs:comment</code>                                   | Diese Klasse beschreibt da        |
| <code>cidoc:E24.Physical_Man-Made_Thing</code>              |                                   |
| <code>hat_Literatur</code>                                  | <code>min 0</code>                |
| <code>hat_Objektgeschichte</code>                           | <code>max 1</code>                |
| <code>hat_Titel</code>                                      | <code>exactly 1</code>            |
| <code>hat_Versionsnummer</code>                             | <code>min 1</code>                |
| <code>was_produced_by_Herstellung_Museumobjekt</code>       | <code>exactly 1</code>            |
| <code>wird_bestimmt_durch_HM_Ref_ID</code>                  | <code>min 0</code>                |
| <code>wird_bestimmt_durch_InvNr</code>                      | <code>min 1</code>                |
| <code>wird_durch_Augenscheinbeschreibung_beschrieben</code> | <code>exactly 1</code>            |
| <code>wird_durch_Messung_beschrieben</code>                 | <code>min 0</code>                |
| <code>wird_eindeutig_bestimmt_durch</code>                  | <code>exactly 1</code>            |
| <code>cidoc:P104.is_subject_to</code>                       | <code>some cidoc:E30.Right</code> |
| <code>cidoc:P105.right_held_by</code>                       | <code>some cidoc:E39.Actor</code> |
| <code>cidoc:P1081.was_produced_by</code>                    | <code>exactly 1</code>            |
| <code>cidoc:P1241.was_transformed_by</code>                 | <code>max 1</code>                |
| <code>cidoc:P131.was_destroyed_by</code>                    | <code>max 1</code>                |

<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)



G. Görz, FAU, Inf. 8

