Taking CIDOC apart: Exercise in modularisation and future steps

Emilio M. Sanfilippo
Laboratory for Applied Ontology ISTC-CNR, Trento, IT
emilio.sanfilippo@cnr.it
Presentation overview

1. Ontology modularisation
   - Basic Hints
   - Modularisation in OWL

2. CIDOC CRM meets modularity (based on paper at FOIS 2020)
   - Overview
   - Remarks

3. (Possible) Future steps
   - Ontology Patterns (OPs) for CIDOC CRM
Refer to

Preliminary research work (during a post-doc scholarship in collaboration with the CESR at the University of Tours, France):


Available at [IOS Press](https://www.iospress.nl), [Research Gate](https://www.researchgate.net) (send me an email otherwise)
Ontology modularisation
Ontology modularisation

**Modularity** (Khan-Keet 2015)

In its most generic meaning, *modularity* denotes the possibility to perceive a large knowledge repository [...] as a set of modules, i.e. smaller repositories that, in some way, are parts of and compose the whole thing

**Module**

A module is a subset of an ontology that captures all the knowledge the ontology contains about a given set of terms
Ontology modularisation (con’t)

**Why** modularity (Khan-Keet 2015):

- Maintenance
- Partial reuse
- Comprehension
- (Collaborative) Development
- Automated reasoning
- Visualisation
- ....

Consider the **Foundational Model of Anatomy** ([FMA](#)) with > 100,000 classes
Ontology modularisation in OWL

OWL imports mechanism is **axiom-based** (Rector et al. 2012)

**Advantage:**

- Information about the same entity (e.g., class) but in different modules can be **easily merged** [**IMPORTANT:** keep logical consistency!!]
- **Order** of imports does **not** matter; axioms are aggregated
Ontology modularisation in OWL: Simple example

Module 1

Module 2

Protégé views
Ontology modularisation in OWL: Simple example (con’t)

View of OWL imports mechanism

Example of binding axioms between the modules, including disjunction (not shown)
CIDOC CRM meets modularity
CIDOC CRM meets modularity

Goals: Partial reuse (users’ comprehension)

Criteria and desiderata driving the modularisation:

• Levels of generality and ontological similarity between classes (see next slides)
• Allow for the automatic integration of modules when joined together to re-built the entire CRM taxonomy

Recall (from 48th CRM SIG) that we have also **revised** CRM v.6.2.1
Top-down modularisation

- Leaf modules (e.g., actors, artefacts, etc.) import (via `owl:imports`) higher-level modules (e.g., physical thing)
- Higher-level modules provide the common taxonomical structure to integrate leaf modules

Example of imports between modules
Overview of modules for persistent items
Top-down modularisation: Example

**Imported ontologies:**

**Direct Imports**

- `<http://erlangen-crm.org/actor>`
  - Ontology IRI: `<http://erlangen-crm.org/actor>`

- `<http://erlangen-crm.org/artefact>`
  - Ontology IRI: `<http://erlangen-crm.org/artefact>`

**Indirect Imports**

- `<http://erlangen-crm.org/physical-thing>`
  - Ontology IRI: `<http://erlangen-crm.org/physical-thing>`
  - Location: `/Users/emiliosanfilippo/Desktop/CRM-SIG21_talk/cidoc-modularization-master/physical_thing_module.owl`

- `<http://erlangen-crm.org/dimension>`
  - Ontology IRI: `<http://erlangen-crm.org/dimension>`

- `<http://erlangen-crm.org/persistent-item-top>`
  - Ontology IRI: `<http://erlangen-crm.org/persistent-item-top>`
  - Location: `/Users/emiliosanfilippo/Desktop/CRM-SIG21_talk/cidoc-modularization-master/persistent_item_top_module.owl`

- `<http://erlangen-crm.org/place>`
  - Ontology IRI: `<http://erlangen-crm.org/place>`
  - Location: `/Users/emiliosanfilippo/Desktop/CRM-SIG21_talk/cidoc-modularization-master/place_module.owl`

**Class hierarchy:**

```
owl:Thing
  E53_Place
  E54_Dimension
  E77_Persistent_Item
    E18_Physical_Thing
      E19_Physical_Object
      E24_Physical_Man-Made_Thing
        E22_Man-Made_Object
        E25_Man-Made_Feature
        E78_Collection
      E26_Physical_Feature
        E25_Man-Made_Feature
        E27_Site
    PhysicalAggregation
      E74_Group
        E40_Legal_Body
        E78_Collection
      E39_Actor
        E21_Person
        E74_Group
        E40_Legal_Body
    E71_Man-Made_Thing
    NonPhysicalThing
      QualitativeQualitySpace
```
Overview of modular library

Library of modules comprises:

• **Persistent items**: 6 modules
• **Temporal entities**: 8 modules
• **Places**: 1 module
• **Dimensions**: 1 module
• To build the entire CIDOC CRM: 2 modules
Remark (1)

Advantage:

- **Automatic integration** of modules via the common high-level structure
- Child classes **inherit** relations and axioms from their parent classes

For example:

- `E22_Man_Made_Object` inherits `P46_is_composed_of` from `E18_Physical_Thing`
Remark (2)

Disadvantage:

- Each module consists of modelling elements relevant in the scope of the module (e.g., actors) **and** higher-level modelling elements

This choice:

- Increases the complexity of the conceptual and formal structures of each module

In addition, no evaluation against case studies (**preliminary work**).
(Possible) Future steps
Ontology Patterns (OP)

An ontology pattern (aka ontology design pattern, knowledge pattern, linked data pattern):

- **Established modelling solution** to solve a **recurrent ontology development problem** (Falbo et al. 2013)

That is:

- **Established** modelling solution: the pattern is a well-proven solution
- **Recurrent** ontology development problem, e.g., for domain ontologies
Different types of OPs

For the sake of this discussion: **Ontology Conceptual Pattern**

- Focus only on **conceptual aspects** without any concern with the technology or language to be used for an operational ontology
- Extracted from foundational ontologies (**Foundational Ontology Pattern**) or domain ontologies (**Domain-Related Ontology Pattern**)
Examples of Ontology Conceptual Patterns

Foundational Ontology Pattern

Object $\xrightarrow{1..*}$ Event

$\text{participatesIn}$

$\text{occurs in}$

Time region

Domain-Related Ontology Pattern

Manufacturing $\text{machine}$ $\xrightarrow{1}$ Component

$\text{has component}$
Reuse of OPs: By analogy and By extension

**Reuse by analogy:** reproduce the selected pattern with domain notions.

**Reuse by extension:** embed domain notions within the selected pattern by extending it via subsumption relations.
CIDOC CRM meets OPs
CIDOC CRM meets OPs

Develop conceptual OPs — possibly leading to OPs in OWL — for recurrent CIDOC-based modeling solutions

**Basic requirement:** the OPs must be **coherent** with the structure of CIDOC

In principle, this would

- Enable the **partial reuse** of CIDOC, e.g., with respect to application scenarios
- Allow for the **extension** of CIDOC in a selected manner
- Perhaps, **facilitate the understanding** of CIDOC for novel users
**CIDOC CRM meets OPs: Example**

E5 inherits P7 from E4 Period

E5 Event \(\rightarrow\) E53 Place

\[\text{P7 took place at} \]

\[\text{P87 is identified by} \]

E44 Place Appellation

E5 Event \(\Rightarrow\) E52 Time-Span

\[\text{P4 has time-span} \]

\[\text{P78 is identified by} \]

E49 Time Appellation

E5 Event \(\rightarrow\) E39 Actor

\[\text{P11 had participant} \]

**Remark:** these are just examples! Also, no cardinality in relationships just for simplicity
CIDOC CRM meets OPs: Example (con’t)

Integration of the three patterns
OPs: At which granularity, level of detail?

“Ideally, ontology design patterns should be extendable but self-contained, minimize ontological commitments to foster reuse, address one or more explicit requirements (or use cases, competency questions) […], be the representation of a core notion in a domain of expertise […], be alignable to other patterns, span more than one application area or domain, address a single invariant instead of targeting multiple reoccurring issues at the same time, follow established modelling best practices, and so forth.” (Janowicz et al. 2016)
OPs: At which granularity, level of detail? Example

MASA consortium, from Issue 364
OPs: At which granularity, level of detail? Example (con’t)

Req: Model/retrieve E22’s meta-data

Req: Model/retrieve E22’s dimensions

Req: Model/retrieve E22’s type(s)

Req: Model/retrieve E22’s material(s)

Remark: just examples
Ontology modularisation: Some references


Ontology patterns: Some references

Thank you!
(A special thank to Béatrice Markhoff and Martin Doerr)

Emilio M. Sanfilippo
Laboratory for Applied Ontology ISTC-CNR, IT
emilio.sanfilippo@cnr.it