Taking CIDOC apart: Exercise in modularisation and future steps

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

Consiglio Nazionale delle Ricerche

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):

> Sanfilippo, E.M., Markhoff, B., Pittet, P. (2020). *Ontological analysis and modularization of CIDOC-CRM*, Proceedings of FOIS XI, IOS Press

Available at IOS Press, Research Gate (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)

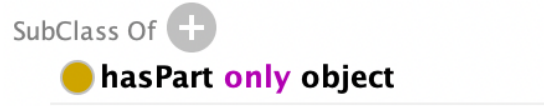
Advantage:

- 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

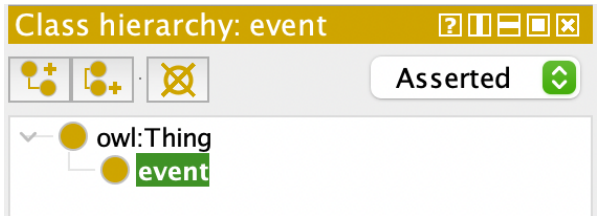
Ontology modularisation in OWL: Simple example

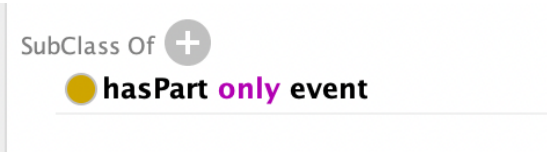
Protégé views

Module 1



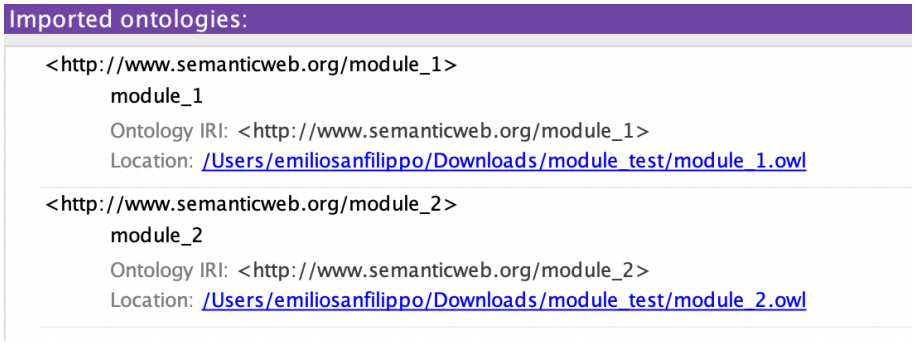
Module 2



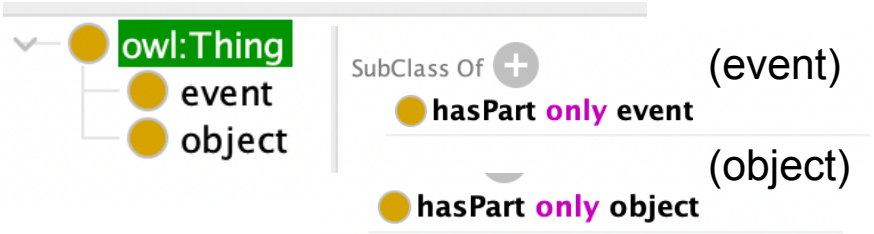


Ontology modularisation in OWL: Simple example (con't)

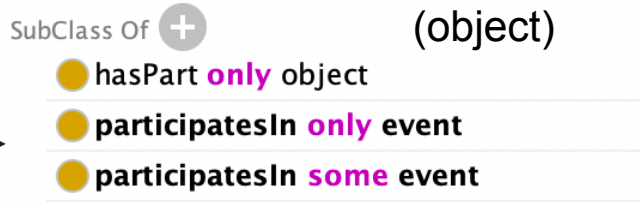
View of OWL imports mechanism



Module 3 importing Module 1 and Module 2



Example of binding axioms between the modules, including disjunction (not shown)



CIDOC CRM meets modularity

CIDOC CRM meets modularity

Goals: Partial reuse (users' comprehension)

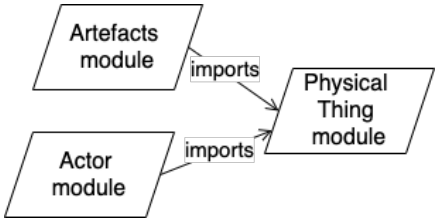
Criteria and **desiderata** driving the modularisation:

- 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

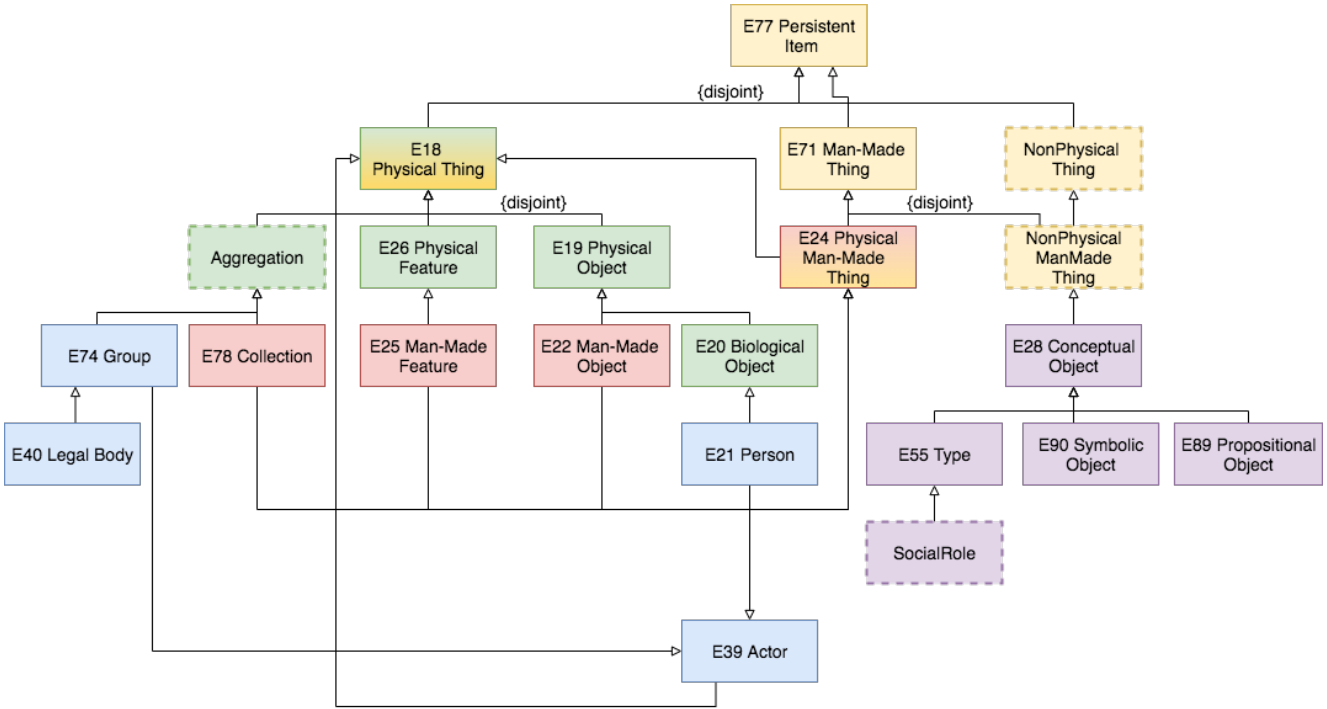
Top-down modularisation

- 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

Directly imported modules

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

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

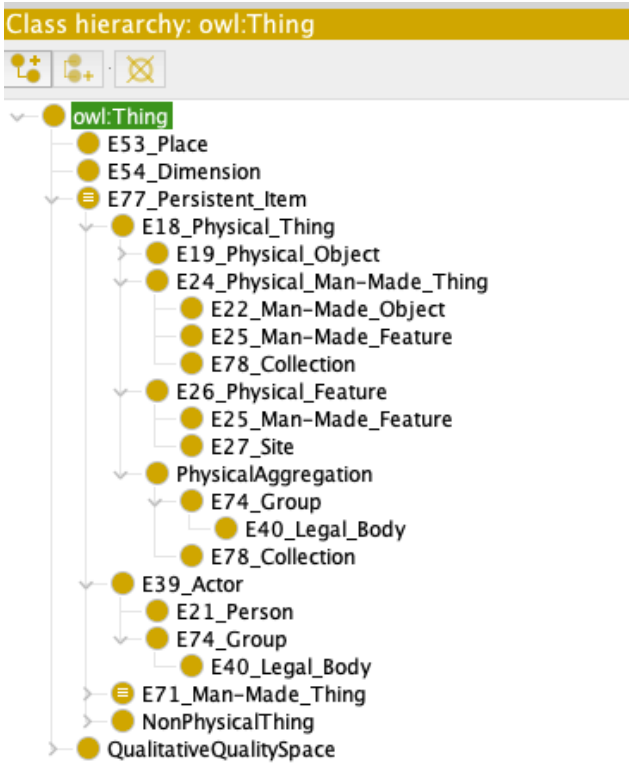
Indirect Imports

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

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



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)

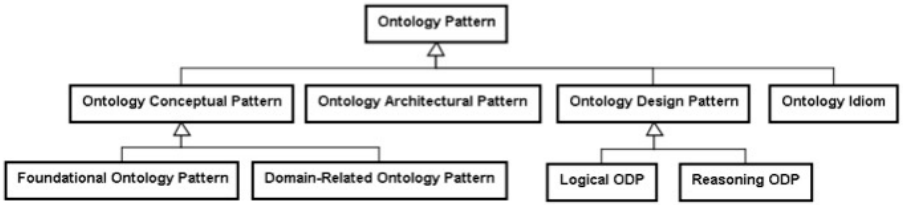
An ontology pattern (aka ontology design pattern, knowledge pattern, linked data pattern):

- 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

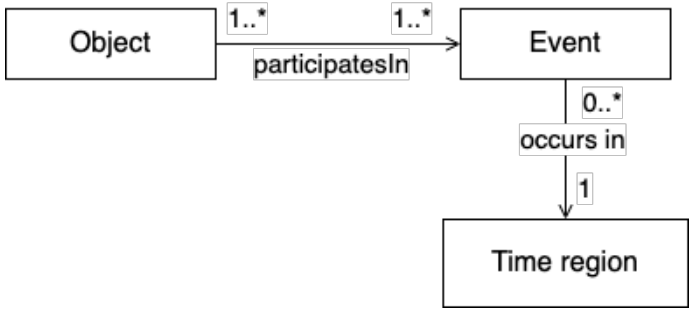


From Falbo et al. 2013

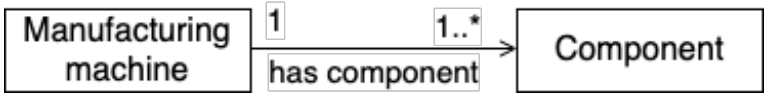
For the sake of this discussion: Ontology Conceptual Pattern

- 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

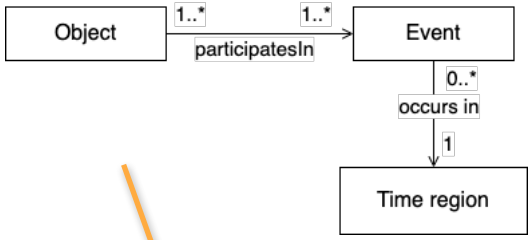


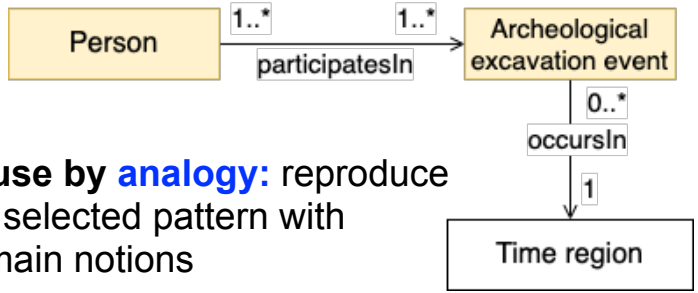
Foundational Ontology Pattern



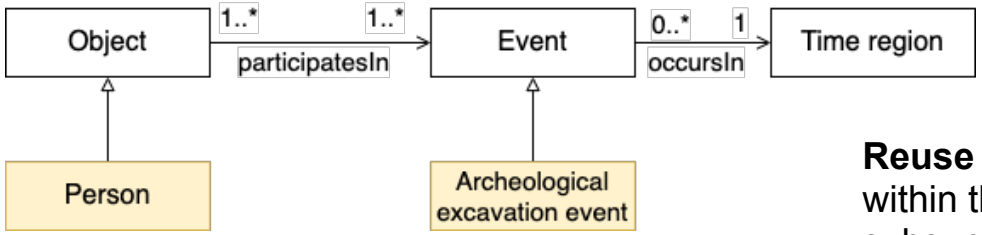
Domain-Related Ontology Pattern

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

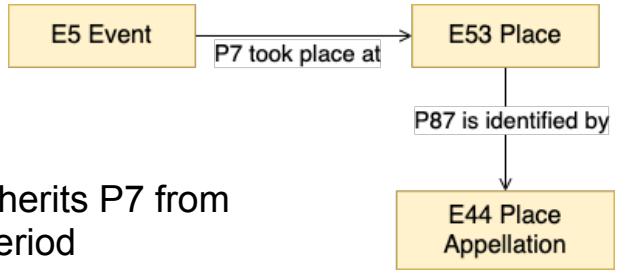
CIDOC-based modeling solutions

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

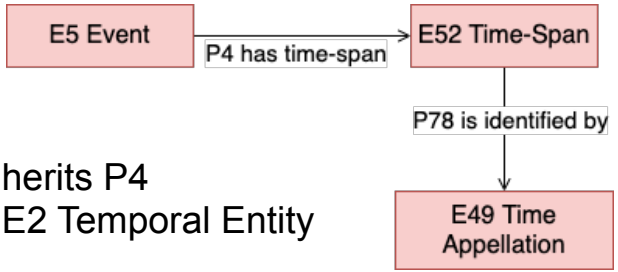
In principle, this would

- 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

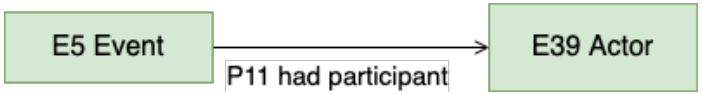
CIDOC CRM meets OPs: Example



E5 inherits P7 from
E4 Period

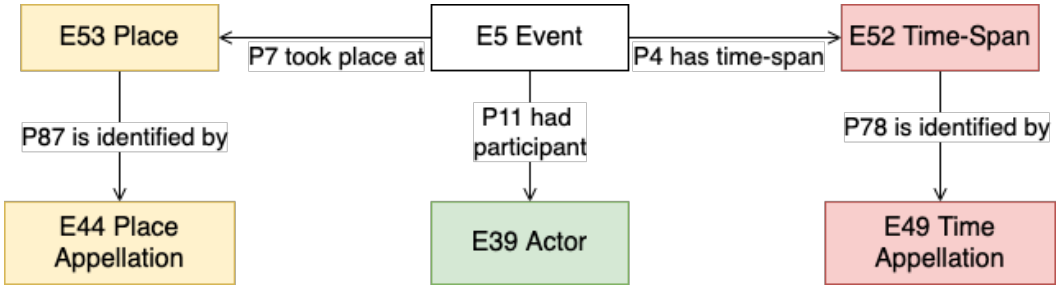


E5 inherits P4
from E2 Temporal Entity



Remark: these are just examples! Also, no cardinality in relationships just for simplicity

CIDOC CRM meets OPs: Example (con't)

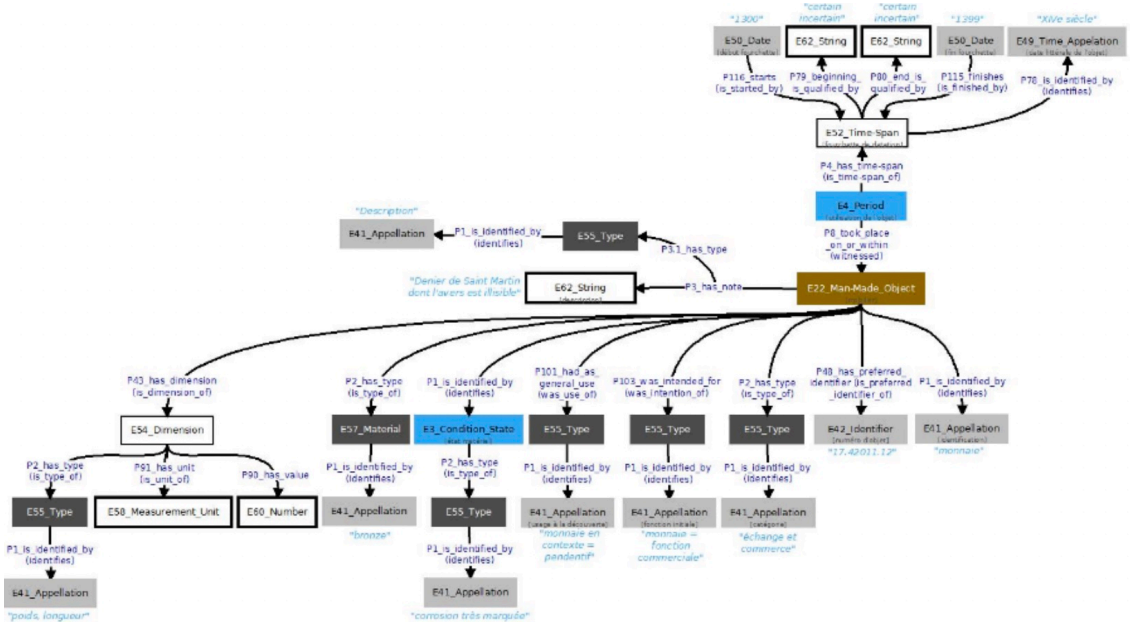


Integration of the three patterns

OPs: At which granularity, level of detail?

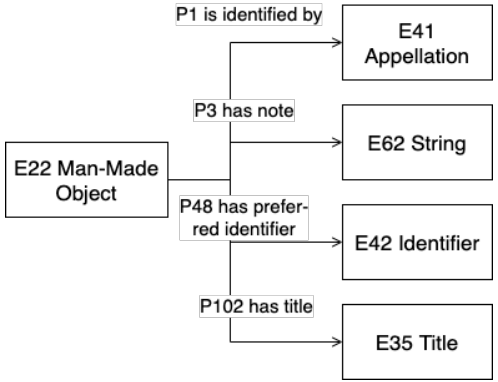
"Ideally, ontology design patterns should be extendable but self-contained, minimize ontological commitments to foster reuse, address one or more explicit requirements (or use cases, competency questions) […], be the representation of a core notion in a domain of expertise […], be alignable to other 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

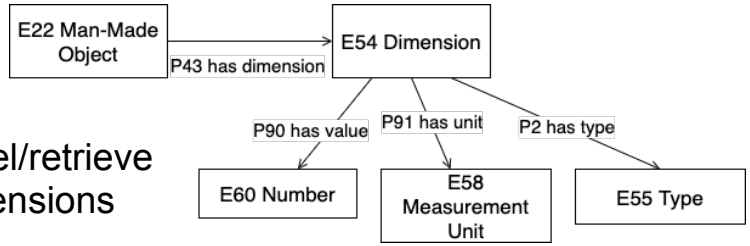


MASA consortium, from [Issue 364]

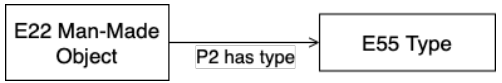
OPs: At which granularity, level of detail? Example (con't)



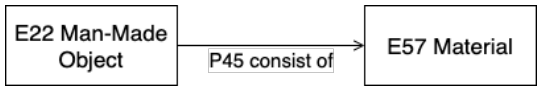
Req: Model/retrieve E22's meta-data



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

- Del Vescovo, C., Horridge, M., Parsia, B., Sattler, U., Schneider, T., & Zhao, H. (**2020**). Modular Structures and Atomic Decomposition in Ontologies. Journal of Artificial Intelligence Research, 69, 963-1021
- Khan, Z. C., & Keet, C. M. (**2015**). *An empirically-based framework for ontology modularisation*. Applied Ontology, 10(3-4), 171-195
- Rector, A., Brandt, S., Drummond, N., Horridge, M., Pulestin, C., & Stevens, R. (**2012**). Engineering use cases for modular development of ontologies in OWL. Applied Ontology, 7(2), 113-132 (within special issue: Modularity in Ontologies)
- Shimizu, C., Krisnadhi, A., & Hitzler, P. (**2020**). Modular ontology modeling: A tutorial. *Applications and Practices in Ontology Design, Extraction, and Reasoning*. *Studies on the Semantic Web*. *IOS Press*. *To appear*. [new paper submitted to SW Journal]
- Stuckenschmidt, H., Parent, C., & Spaccapietra, S. (Eds.). (**2009**). Modular ontologies: concepts, theories and techniques for knowledge modularization (Vol. 5445). Springer.

Ontology patterns: Some references

- Carriero, V. A., Gangemi, A., Mancinelli, M. L., Nuzzolese, A. G., Presutti, V., & Veninata, C. (2019). Pattern-based design applied to cultural heritage knowledge graphs. *Semantic Web*, (Preprint), 1-45.
- Falbo, R. D. A., Guizzardi, G., Gangemi, A., & Presutti, V. (2013, October). Ontology patterns: clarifying concepts and terminology. In *Proceedings of the 4th Workshop on Ontology and Semantic Web Patterns*.
- Falbo, R. A., Barcellos, M. P., Ruy, F. B., Guizzardi, G., & Guizzardi, R. S. S. (2016). Ontology pattern languages. In *Ontology Engineering with Ontology Design Patterns: Foundations and Applications*. IOS Press.
- Hitzler, P., Gangemi, A., Janowicz, K., Krisnadhi, A. A., & Presutti, V. (2017). Towards a Simple but Useful Ontology Design Pattern Representation Language. In *WOP@ ISWC*.
- Hitzler, P., Gangemi, A., & Janowicz, K. (Eds.). (2016). *Ontology engineering with ontology design patterns: Foundations and applications* (Vol. 25). IOS Press. (book)
- Karima, N., Hammar, K., & Hitzler, P. (2017). How to document ontology design patterns. *Advances in Ontology Design and Patterns*, *32*, 15-28.

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