Ontology Modeling 2.0: Next Steps

Pascal Hitzler

Data Semantics Laboratory (DaSe Lab)
Data Science and Security Cluster (DSSC)
Wright State University
http://www.pascal-hitzler.de

Modules

A module is a part of an ontology which corresponds to a conceptual part of the domain which can be summarized under one (key) notion within the topic domain.

E.g., in a chess games ontology there may be an “opening” module, and an “tournament” module.

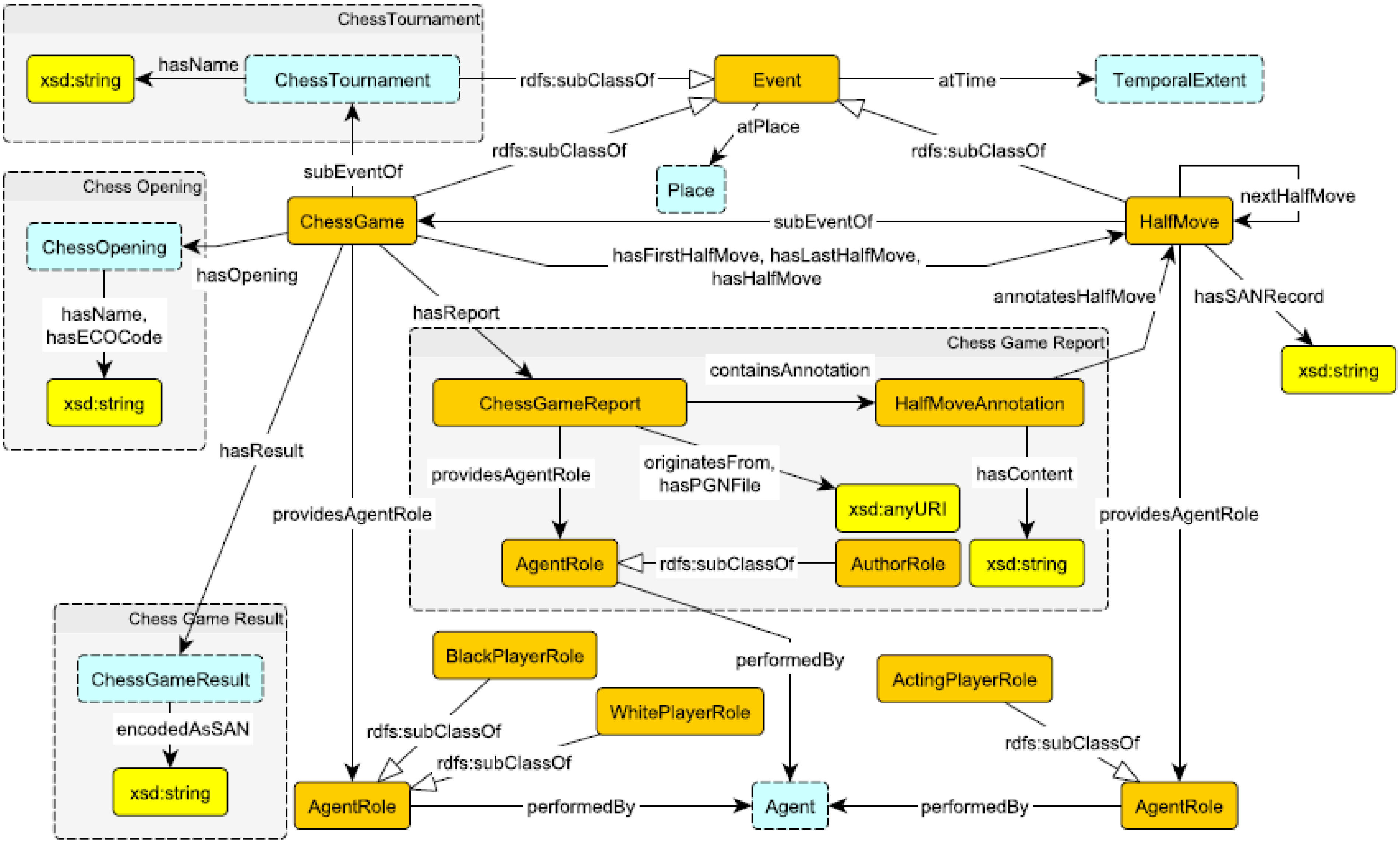
E.g., in a travel planning support ontology, there may be a “hotel” module, and a “trajectory” module.

We understand ontologies to be composed of modules.

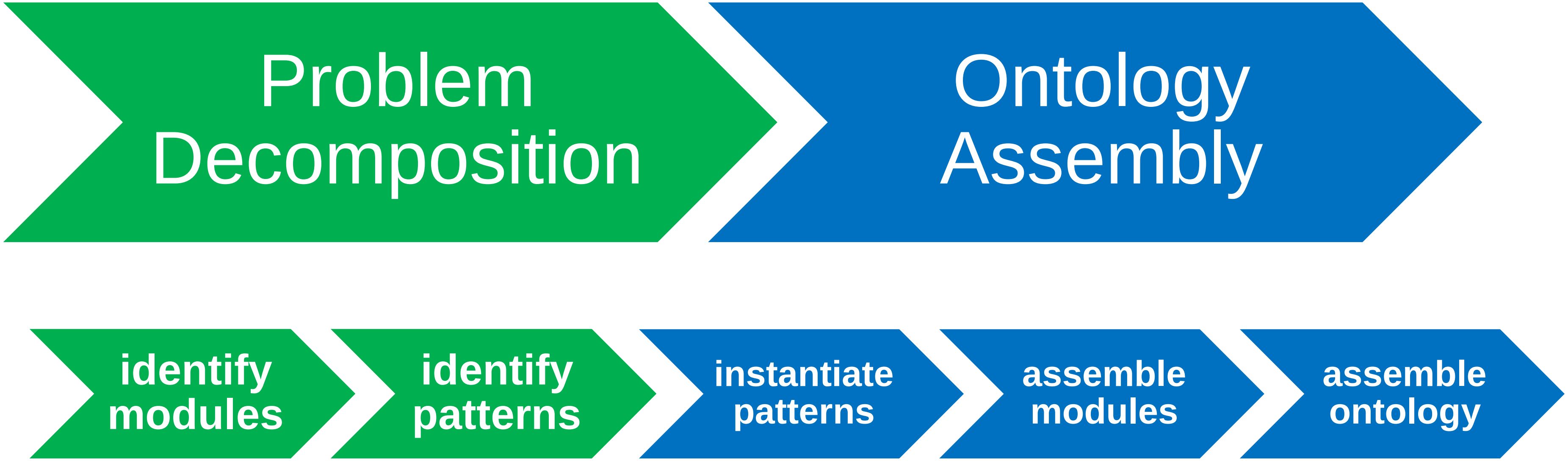
We understand modules to be obtained by instantiating, joining and modifying patterns.

Modules

DaSe Lab



Divide and Conquer



Problem Decomposition

Ontology Assembly

identify modules

identify patterns

instantiate patterns

assemble modules

assemble ontology

Decomposition

identify modules
identify patterns

Module identification:

- Main notions relevant to data and modeling problem.
- Driven by competency questions, use case descriptions, and inspection of available data sources.
- Best done on a whiteboard.
- E.g. (chess), “moves”, “opening”, “players”, “tournament”, …

Decomposition

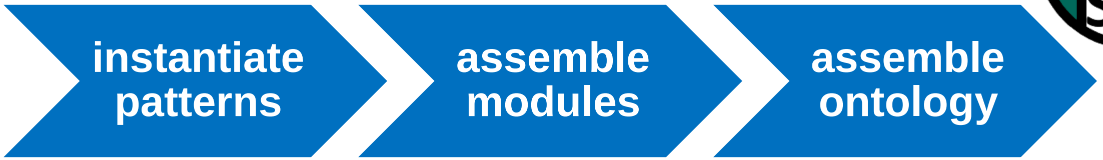
identify modules
identify patterns

Pattern identification:

- **For each module, which pattern(s) reflect the nature of this module?**
- **Driven by competency questions, use case descriptions, inspection of available data sources, identified modules, repository of available patterns.**
- **Best done on a whiteboard(?)**
- **E.g. (chess),**
  “moves” → list pattern
  “players” → agent role pattern
  “tournament” → event pattern

Assembly

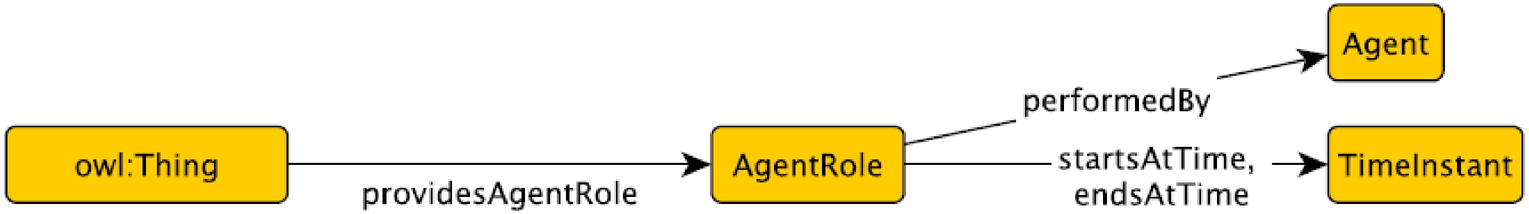
Pattern instantiation:

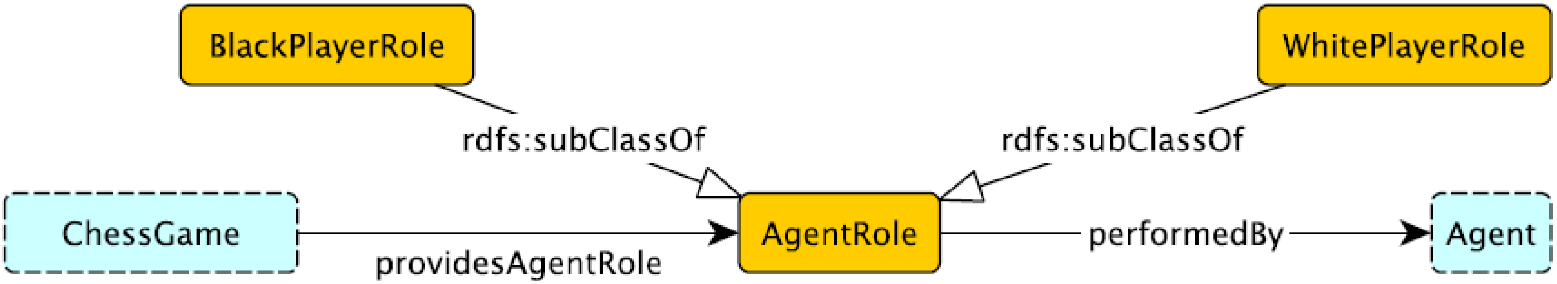


- Identified (generic) patterns to be used as templates:
  - Instantiate (change class/property names), and import to local namespace
  - Modify (adapt to the specific need)
- Provenance information (which pattern was used and how) should be kept.

- Strong tool support needed. Could be graphical, but needs to work on axiom level.
  (draft solution: Karl Hammar’s XD Protégé plug-in)

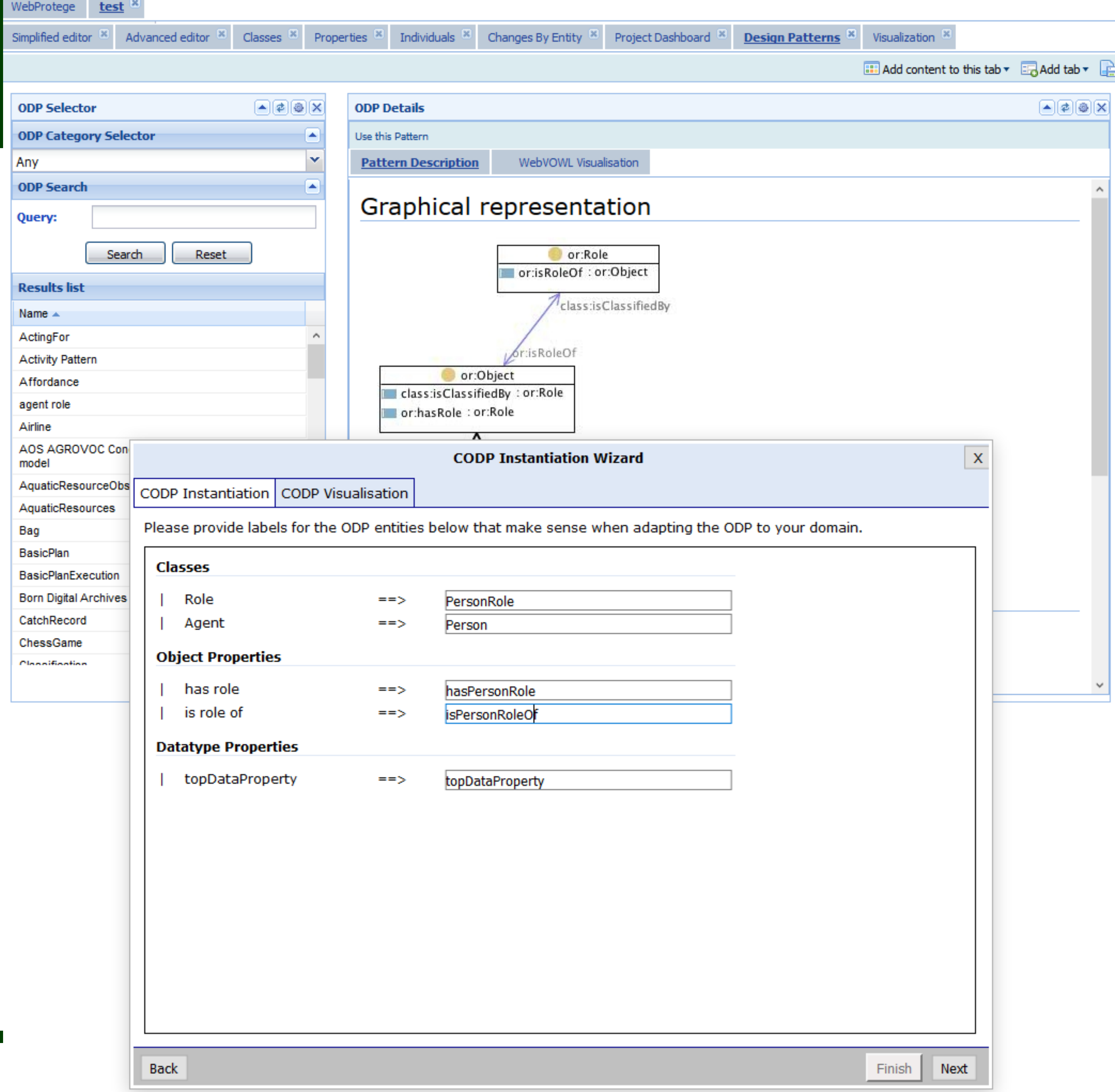
Player as AgentRole





XD plug-in

[Hammar]



Please provide labels for the ODP entities below that make sense when adapting the ODP to your domain.

Classes

|	Role	==>	PersonRole
|	Agent	==>	Person

Object Properties

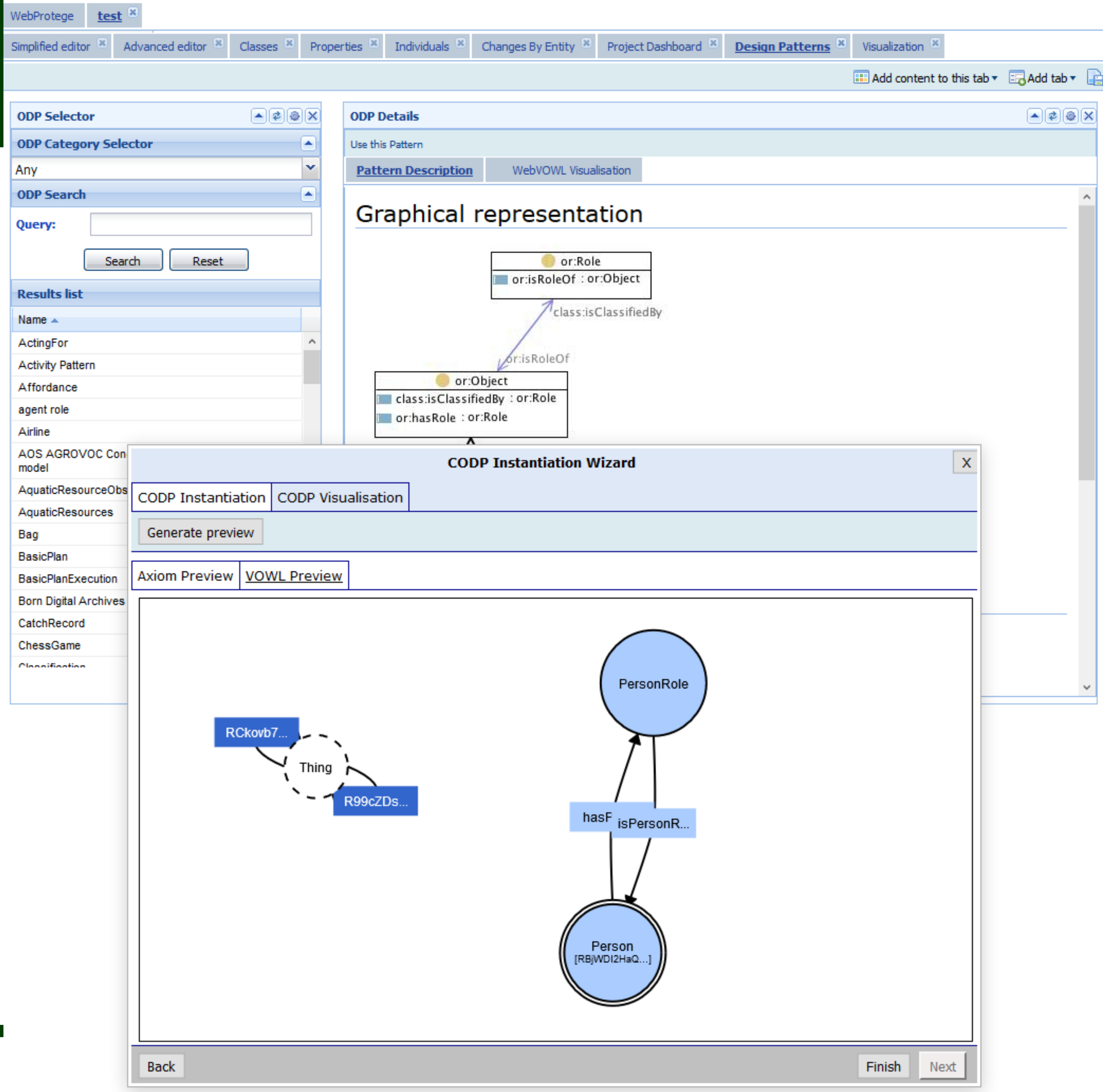
|	has role	==>	hasPersonRole
|	is role of	==>	isPersonRoleOf

Datatype Properties

|	topDataProperty	==>	topDataProperty

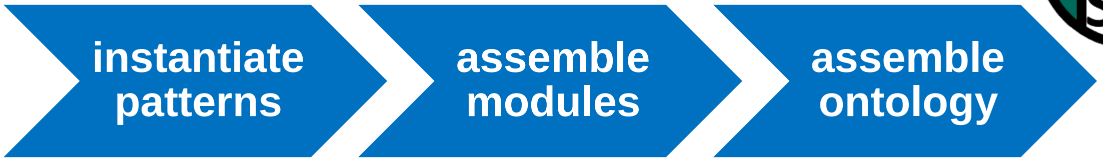
XD plug-in

[Hammar]



Assembly

Module assembly:

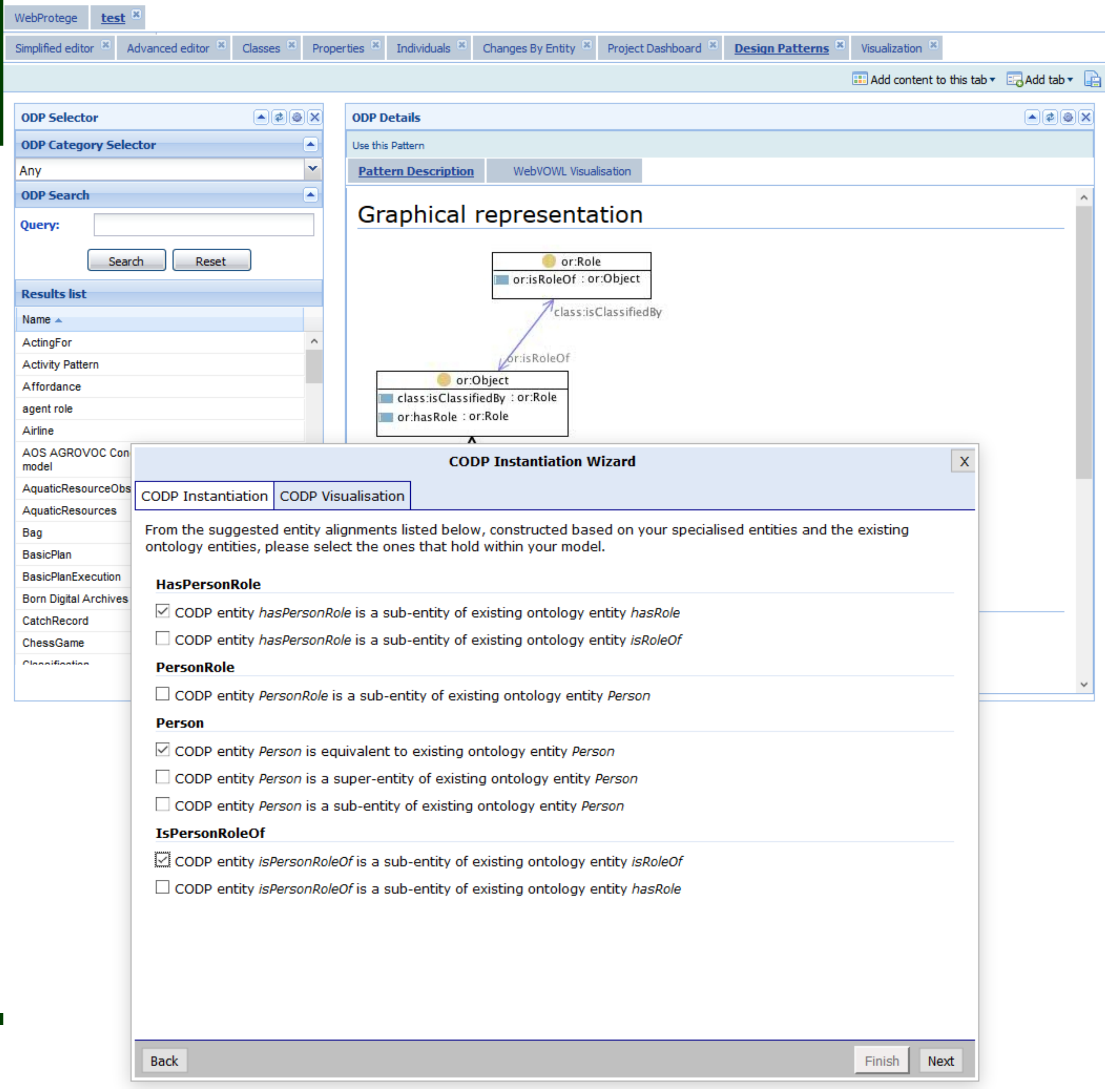


- From instantiated patterns.
- Plus additional modifications/additions as needed.
- Modules may contain other modules.
- Provenance information (which pattern was used an how) should be kept.
- Modules should be identifiable from the OWL file.

- Strong tool support needed. Could be graphical, but needs to work on axiom level.
  (draft solution: Karl Hammar’s XD Protégé plug-in)

XD plug-in

[Hammar]



CODP Instantiation Wizard

From the suggested entity alignments listed below, constructed based on your specialised entities and the existing ontology entities, please select the ones that hold within your model.

HasPersonRole

☑ CODP entity *hasPersonRole* is a sub-entity of existing ontology entity *hasRole*

☐ CODP entity *hasPersonRole* is a sub-entity of existing ontology entity *isRoleOf*

PersonRole

☐ CODP entity *PersonRole* is a sub-entity of existing ontology entity *Person*

Person

☑ CODP entity *Person* is equivalent to existing ontology entity *Person*

☐ CODP entity *Person* is a super-entity of existing ontology entity *Person*

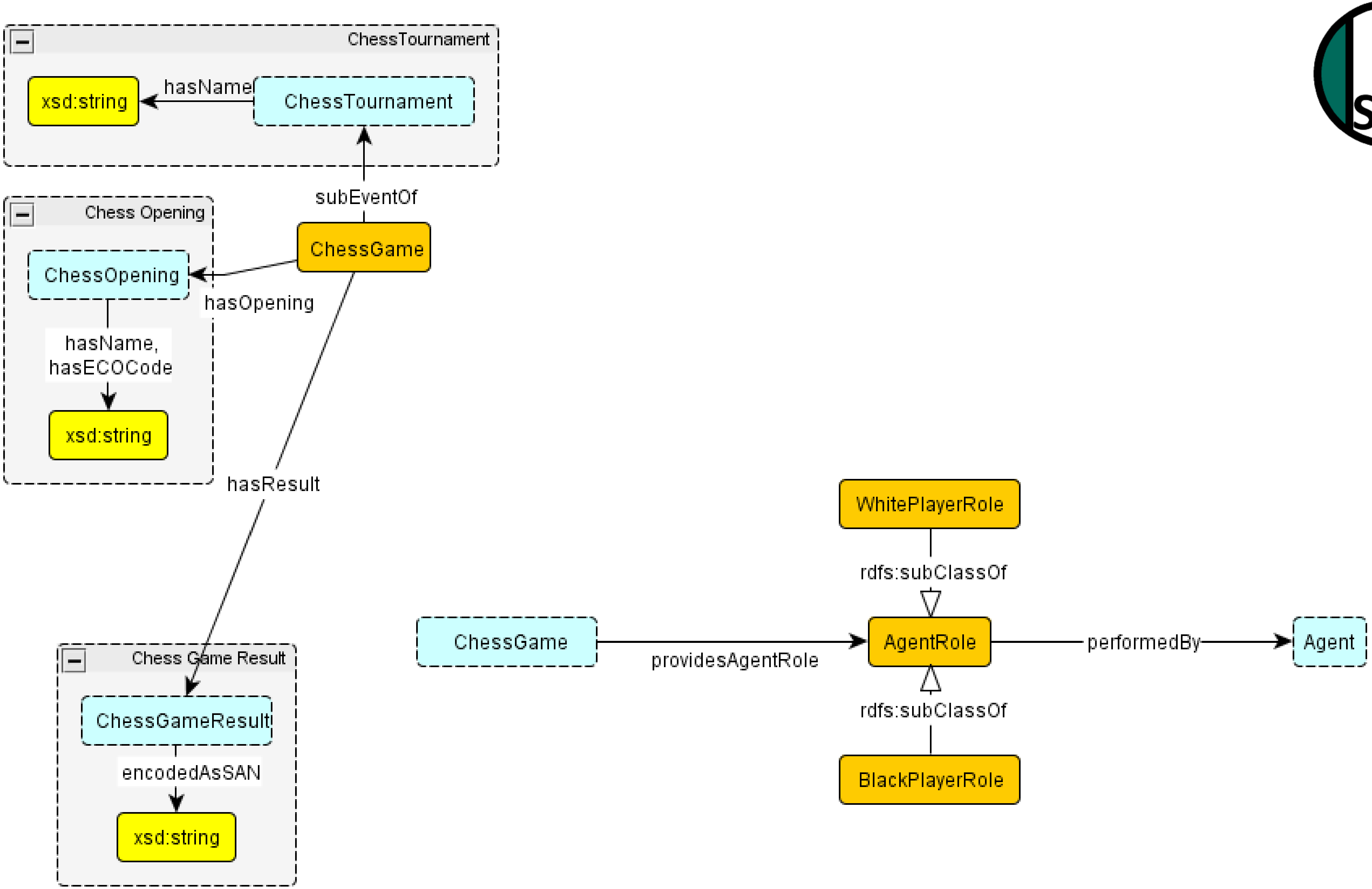
☐ CODP entity *Person* is a sub-entity of existing ontology entity *Person*

IsPersonRoleOf

☑ CODP entity *isPersonRoleOf* is a sub-entity of existing ontology entity *isRoleOf*

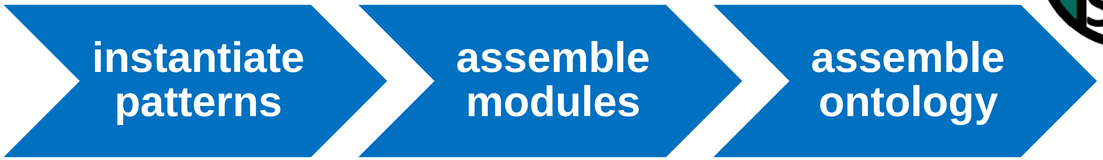
☐ CODP entity *isPersonRoleOf* is a sub-entity of existing ontology entity *hasRole*

Graphical mock-up



Assembly

Ontology assembly:



instantiate patterns

assemble modules

assemble ontology

- From developed modules.
- Plus additional modifications/additions as needed.
- Modules should be identifiable from the OWL file.
- Used patterns should be identifiable from the OWL file.

- Strong tool support needed. Could be graphical, but needs to work on axiom level.

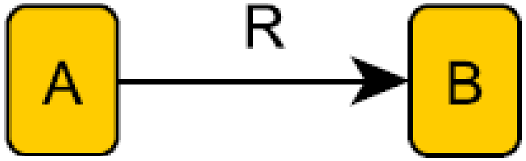
Axiomatization support

- OWLAx Protégé plug-in [ISWC2016 demo]
  - Start with schema diagram.
  - Quick and easy addition of most common axioms using check-box selection.

- ROWL Protégé plug-in [ISWC2016 demo]
  - Rule-based interface for adding complex OWL axioms.
  - Evaluated [ESWC2017 paper] showing that it improves modeling efficiency.

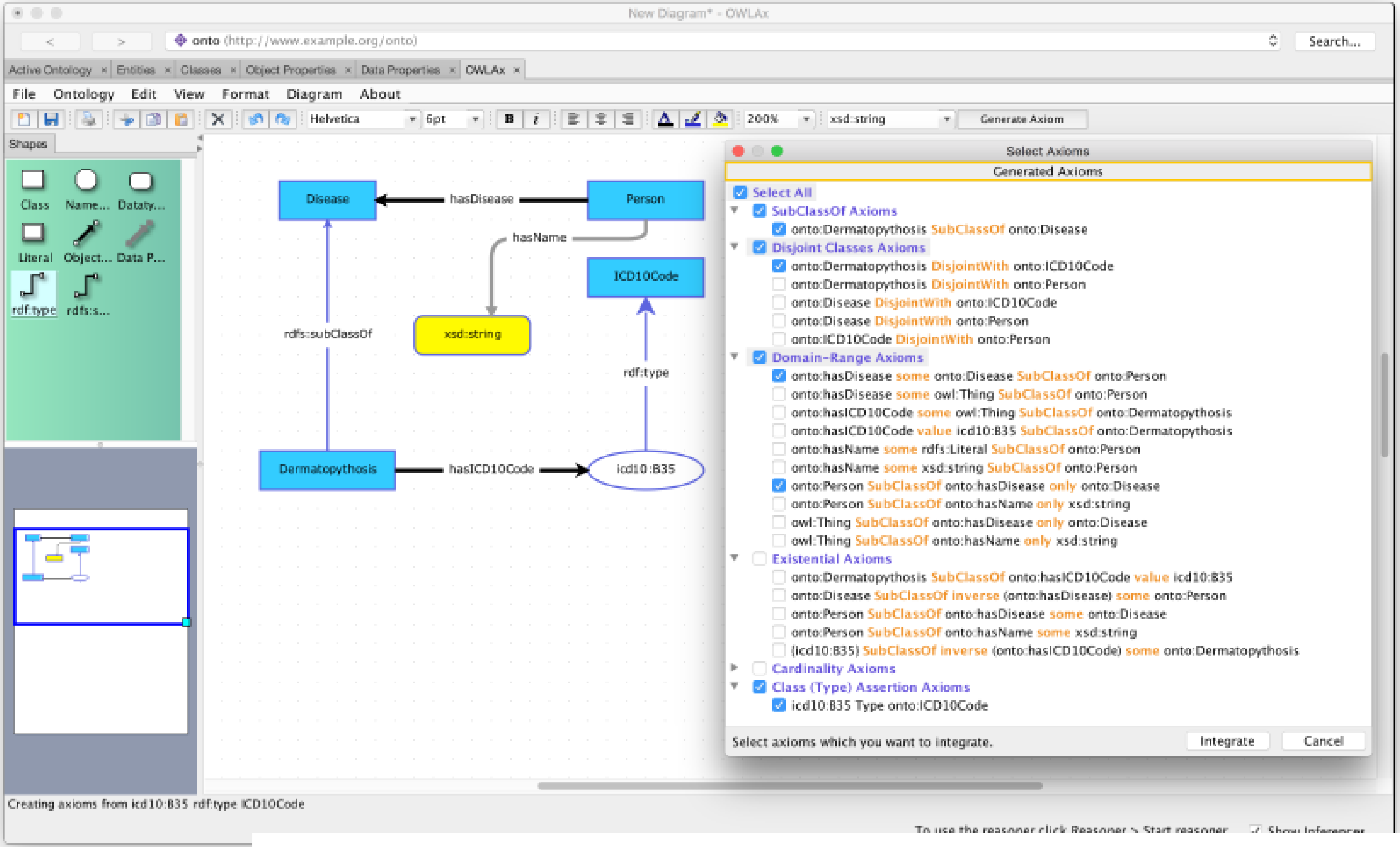
Axioms – Systematically

1. $A \sqcap B \sqsubseteq \bot$
2. $\exists R.\top \sqsubseteq A$
3. $\exists R.B \sqsubseteq A$
4. $\top \sqsubseteq \forall R.B$
5. $A \sqsubseteq \forall R.B$
6. $A \sqsubseteq R.B$
7. $B \sqsubseteq R^{-}.A$
8. $\top \sqsubseteq {\leq}1 R.\top$
9. $\top \sqsubseteq {\leq}1 R.B$
10. $A \sqsubseteq {\leq}1 R.\top$
11. $A \sqsubseteq {\leq}1 R.B$
12. $\top \sqsubseteq {\leq}1 R^{-}.\top$
13. $\top \sqsubseteq {\leq}1 R^{-}.A$
14. $B \sqsubseteq {\leq}1 R^{-}.\top$
15. $B \sqsubseteq {\leq}1 R^{-}.A$



1. A DisjointWith B (disjointness)
2. R some `owl:Thing` SubClassOf A (domain)
3. R some B SubClassOf A (scoped domain)
4. `owl:Thing` SubClassOf R only B (range)
5. A SubClassOf R only B (scoped range)
6. A SubClassOf R some B (existential)
7. B SubClassOf inverse R some A (inverse existential)
8. `owl:Thing` SubClassOf R max 1 `owl:Thing` (functionality)
9. `owl:Thing` SubClassOf R max 1 B (qualified functionality)
10. A SubClassOf R max 1 `owl:Thing` (scoped functionality)
11. A SubClassOf R max 1 B (qualified scoped functionality)
12. `owl:Thing` SubClassOf inverse R max 1 `owl:Thing` (inverse functionality)
13. `owl:Thing` SubClassOf inverse R max 1 A (inverse qualified functionality)
14. B SubClassOf inverse R max 1 `owl:Thing` (inverse scoped functionality)
15. B SubClassOf inverse R max 1 A (inverse qualified scoped functionality)

OWLAx Protégé plug-in



Select Axioms

Generated Axioms

- ☑ Select All
- ☑ SubClassOf Axioms
  - ☑ onto:Dermatopythosis SubClassOf onto:Disease
- ☑ Disjoint Classes Axioms
  - ☑ onto:Dermatopythosis DisjointWith onto:ICD10Code
  - □ onto:Dermatopythosis DisjointWith onto:Person
  - □ onto:Disease DisjointWith onto:ICD10Code
  - □ onto:Disease DisjointWith onto:Person
  - □ onto:ICD10Code DisjointWith onto:Person
- ☑ Domain-Range Axioms
  - ☑ onto:hasDisease some onto:Disease SubClassOf onto:Person
  - □ onto:hasDisease some owl:Thing SubClassOf onto:Person
  - □ onto:hasICD10Code some owl:Thing SubClassOf onto:Dermatopythosis
  - □ onto:hasICD10Code value icd10:B35 SubClassOf onto:Dermatopythosis
  - □ onto:hasName some rdfs:Literal SubClassOf onto:Person
  - □ onto:hasName some xsd:string SubClassOf onto:Person
  - ☑ onto:Person SubClassOf onto:hasDisease only onto:Disease
  - □ onto:Person SubClassOf onto:hasName only xsd:string
  - □ owl:Thing SubClassOf onto:hasDisease only onto:Disease
  - □ owl:Thing SubClassOf onto:hasName only xsd:string
- □ Existential Axioms
  - □ onto:Dermatopythosis SubClassOf onto:hasICD10Code value icd10:B35
  - □ onto:Disease SubClassOf inverse (onto:hasDisease) some onto:Person
  - □ onto:Person SubClassOf onto:hasDisease some onto:Disease
  - □ onto:Person SubClassOf onto:hasName some xsd:string
  - □ {icd10:B35} SubClassOf inverse (onto:hasICD10Code) some onto:Dermatopythosis
- □ Cardinality Axioms
- ☑ Class (Type) Assertion Axioms
  - ☑ icd10:B35 Type onto:ICD10Code

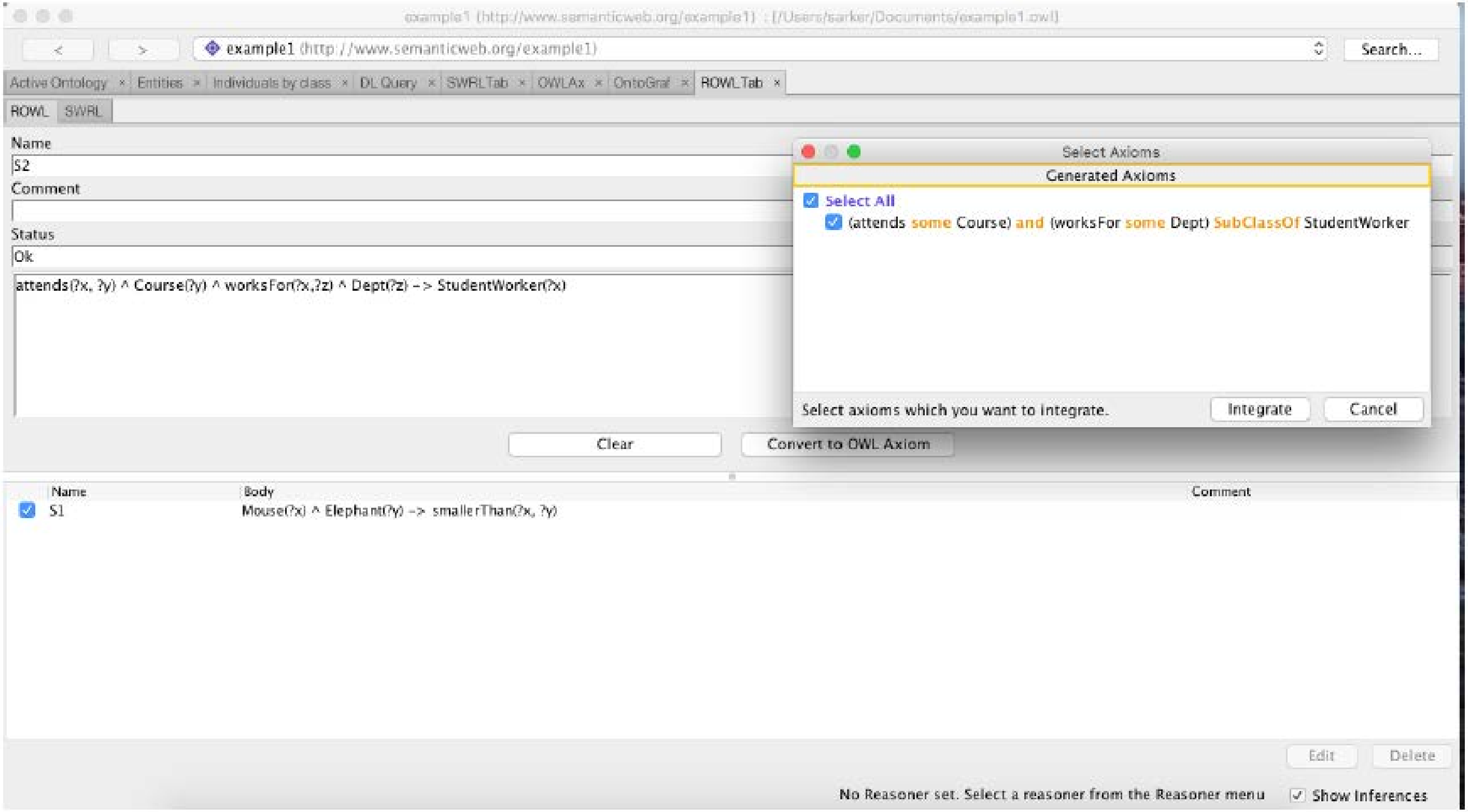
Select axioms which you want to integrate.

Integrate Cancel

Creating axioms from icd10:B35 rdf:type ICD10Code

- In: Proc. ISWC 2016 poster & demos
  http://dase.cs.wright.edu/content/ontology-axiomatization-support

ROWL Protégé plug-in



Name

S2

Comment

Status

Ok

attends(?x, ?y) ^ Course(?y) ^ worksFor(?x,?z) ^ Dept(?z) -> StudentWorker(?x)

Generated Axioms

☑ Select All

☑ (attends some Course) and (worksFor some Dept) SubClassOf StudentWorker

Select axioms which you want to integrate.

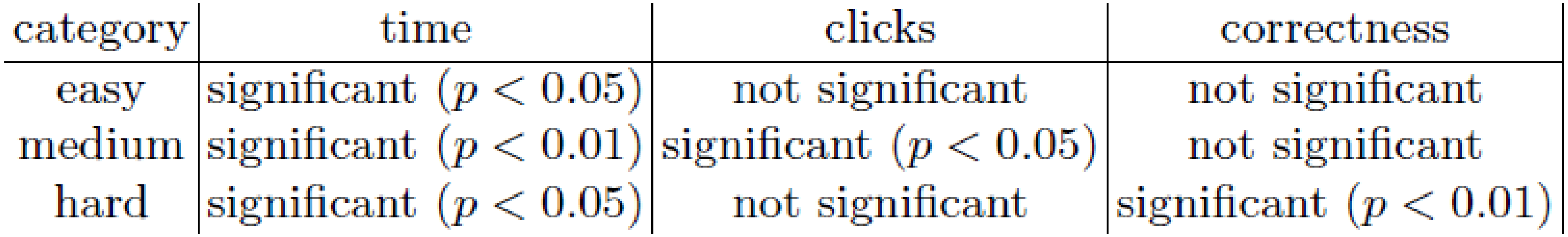
	Name	Body	Comment
☑	S1	Mouse(?x) ^ Elephant(?y) -> smallerThan(?x, ?y)	

No Reasoner set. Select a reasoner from the Reasoner menu ☑ Show Inferences

http://dase.cs.wright.edu/content/rowl

ROWL Evaluation

- **The hypotheses for time and for correctness (hard questions) were confirmed. For correctness (medium questions) the hypothesis was rejected.**



category	time	clicks	correctness
easy	significant ($p < 0.05$)	not significant	not significant
medium	significant ($p < 0.01$)	significant ($p < 0.05$)	not significant
hard	significant ($p < 0.05$)	not significant	significant ($p < 0.01$)

It appears that medium modeling problems (with some role restrictions) can be done correctly with the standard Protégé interface by this type of user, although more time is needed than when using ROWLTab.

It appears that hard problems (requiring rolification) cannot really be solved using the standard Protégé interface, and the unsuccessful solution attempts in addition require more time.

Keeping track

- Keeping track of modules and patterns within an ontology, and of their origins.

- Simple proposed solution: OPLa (Ontology Pattern Language):

  Pascal Hitzler, Aldo Gangemi, Krzystof Janowicz, Adila A. Krisnadhi, Valentina Presutti, Towards a simple but useful ontology design pattern representation language. In: Proceedings WOP 2017.

  See the presentation on this later today.

What is missing?

- High-quality (well-documented) sets of ODPs.
  Perhaps we need to get away from loose collections of ODPs, and rather start talking (and developing) "ODP suites“ which consist of uniformly modeled ODPs.

- Instantiation and composition tools (like Hammar's prototype, but on steroids). They require an ODP language, and graphical support.

- Good, well-written (textbook-style) tutorials tailored to the tools and suites.

Thanks!

References

Hitzler, Krötzsch, Rudolph, Foundations of Semantic Web Technologies, CRC/Chapman & Hall, 2010

Adila Krisnadhi, Pascal Hitzler, Modeling With Ontology Design Patterns: Chess Games As a Worked Example. In: Pascal Hitzler, Aldo Gangemi, Krzysztof Janowicz, Adila Krisnadhi, Valentina Presutti (eds.), Ontology Engineering with Ontology Design Patterns: Foundations and Applications. Studies on the Semantic Web Vol. 25, IOS Press/AKA Verlag, 2016. Chapter 1, pp. 3-22.

Adila Krisnadhi, Nazifa Karima, Pascal Hitzler, Reihaneh Amini, Michelle Cheatham, Víctor Rodríguez-Doncel, Krzysztof Janowicz, Ontology Design Patterns for Linked Data Publishing. In: Pascal Hitzler, Aldo Gangemi, Krzysztof Janowicz, Adila Krisnadhi, Valentina Presutti (eds.), Ontology Engineering with Ontology Design Patterns: Foundations and Applications. Studies on the Semantic Web Vol. 25, IOS Press/AKA Verlag, 2016. Chapter 10, pp. 201-232.

References

A. Gangemi. Ontology design patterns for semantic web content. In Y. Gil et al. (eds), The Semantic Web - ISWC 2005 – 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings, volume 3729 of Lecture Notes in Computer Science, pages 262-276. Springer, 2005

Eva Blomqvist, Pascal Hitzler, Krzysztof Janowicz, Adila Krisnadhi, Thomas Narock, Monika Solanki, Considerations regarding Ontology Design Patterns. Semantic Web 7 (1) 1-7.

Adila A. Krisnadhi, Yingjie Hu, Krzysztof Janowicz, Pascal Hitzler, Robert Arko, Suzanne Carbotte, Cynthia Chandler, Michelle Cheatham, Douglas Fils, Tim Finin, Peng Ji, Matthew Jones, Nazifa Karima, Audrey Mickle, Tom Narock, Margaret O'Brien, Lisa Raymond, Adam Shepherd, Mark Schildhauer, Peter Wiebe, The GeoLink Modular Oceanography Ontology. In: Marcelo Arenas, Óscar Corcho, Elena Simperl, Markus Strohmaier, Mathieu d'Aquin, Kavitha Srinivas, Paul T. Groth, Michel Dumontier, Jeff Heflin, Krishnaprasad Thirunarayan, Steffen Staab (eds.), The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II. Lecture Notes in Computer Science 9367, Springer, Heidelberg, 2015, 301-309.

References

Víctor Rodríguez-Doncel, Adila A. Krisnadhi, Pascal Hitzler, Michelle Cheatham, Nazifa Karima, Reihaneh Amini, Pattern-Based Linked Data Publication: The Linked Chess Dataset Case. In: Olaf Hartig, Juan Sequeda, Aidan Hogan (eds.), Proceedings of the 6th International Workshop on Consuming Linked Data co-located with 14th International Semantic Web Conference (ISWC 2105), Bethlehem, Pennsylvania, US, October 12th, 2015. CEUR Workshop Proceedings 1426, CEUR-WS.org, 2015.

Adila Krisnadhi, Ontology Pattern-Based Data Integration. Dissertation, Department of Computer Science and Engineering, Wright State University, 2015.

References

Md. Kamruzzaman Sarker, David Carral, Adila A. Krisnadhi, Pascal Hitzler, Modeling OWL with Rules: The ROWL Protege Plugin. In: Takahiro Kawamura, Heiko Paulheim (eds.), Proceedings of the ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 19, 2016. CEUR Workshop Proceedings 1690, CEUR-WS.org 2016.

Md. Kamruzzaman Sarker, Adila A. Krisnadhi, Pascal Hitzler, OWLAx: A Protege Plugin to Support Ontology Axiomatization through Diagramming. In: Takahiro Kawamura, Heiko Paulheim (eds.), Proceedings of the ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 19, 2016. CEUR Workshop Proceedings 1690, CEUR-WS.org 2016.

Md Kamruzzaman Sarker, Adila A. Krisnadhi, David Carral, Pascal Hitzler, Rule-based OWL Modeling with ROWLTab Protege Plugin. In: E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler, O. Hartig (eds.), The Semantic Web. 14th International Conference, ESWC 2017, Portoroz, Slovenia, May 28 - June 1, 2017, Proceedings. Lecture Notes in Computer Science Vol. 10249, Springer, Heidelberg, 2017, pp. 419-433.

Karl Hammar, Content Ontology Design Patterns: Qualities, Methods, and Tools. Dissertation. Linköping University, Sweden 2017