



# 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>





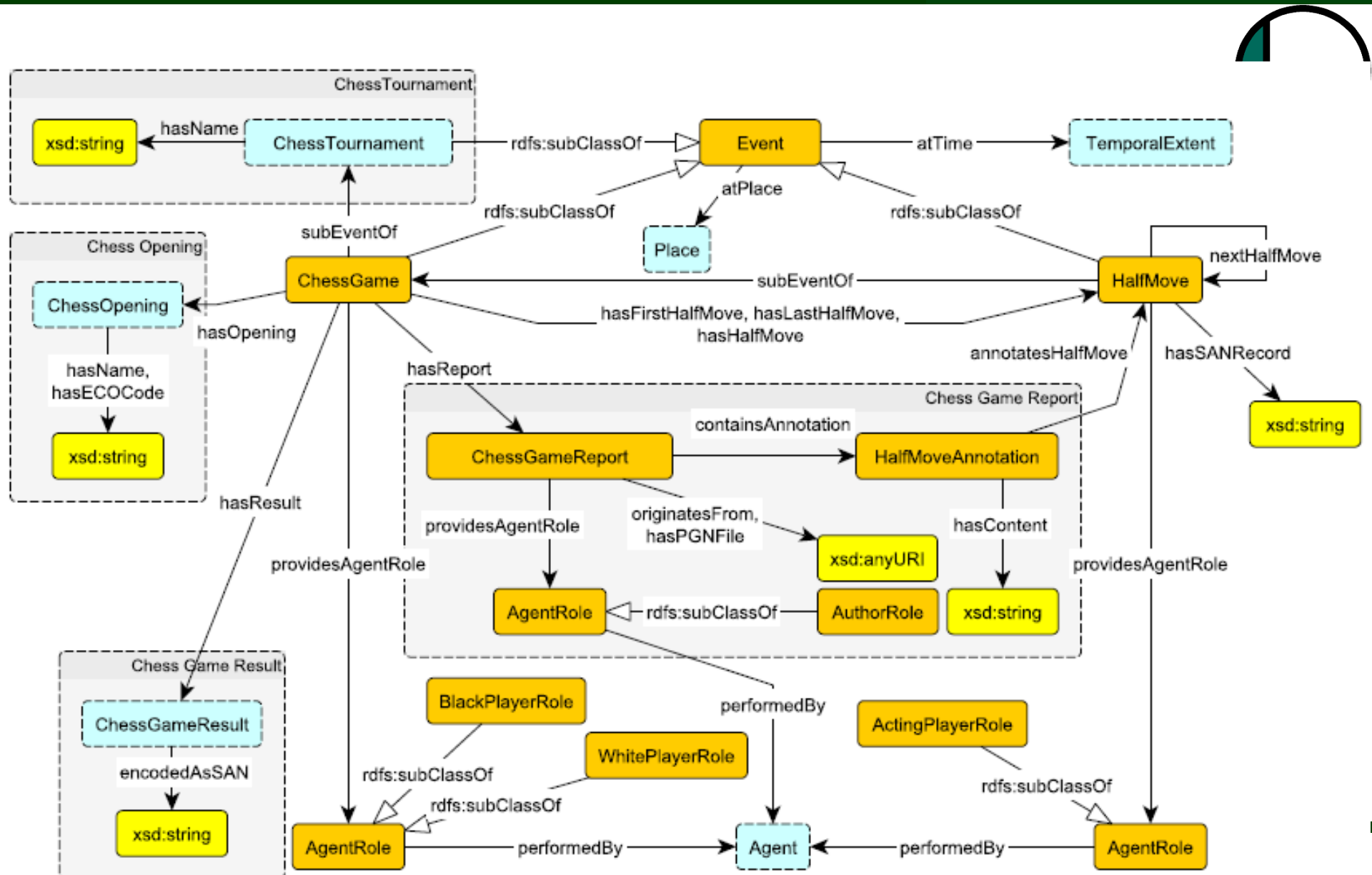
**A module is a part of an ontology which corresponds to a conceptual part of the domain which can be summarized under one (key) notion within the topic domain.**

**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.**





Problem  
Decomposition

Ontology  
Assembly

identify  
modules

identify  
patterns

instantiate  
patterns

assemble  
modules

assemble  
ontology



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”, ...



identify  
modules

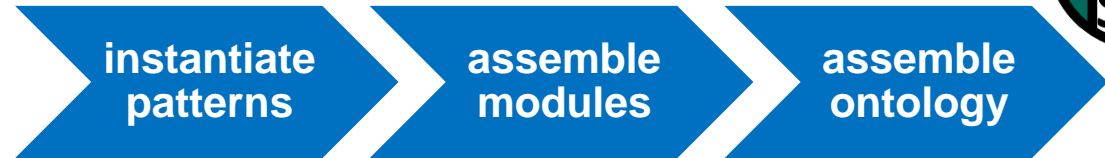
identify  
patterns

## 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

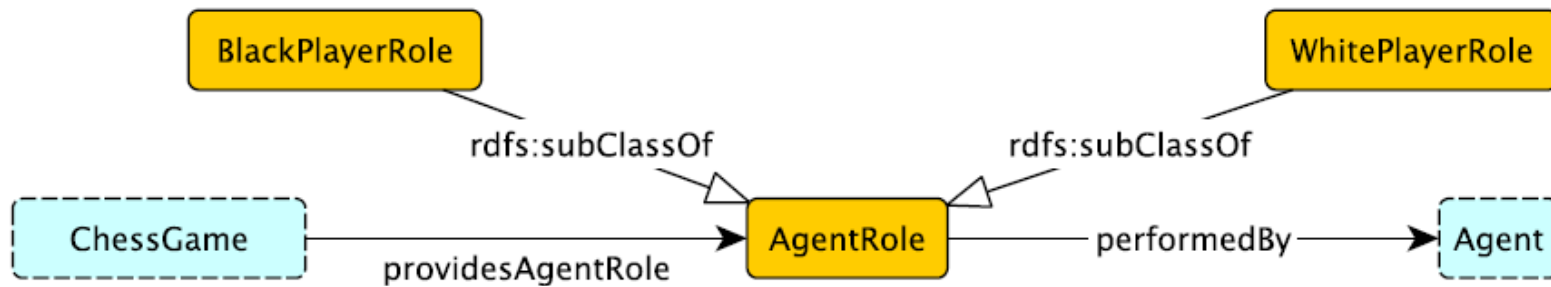
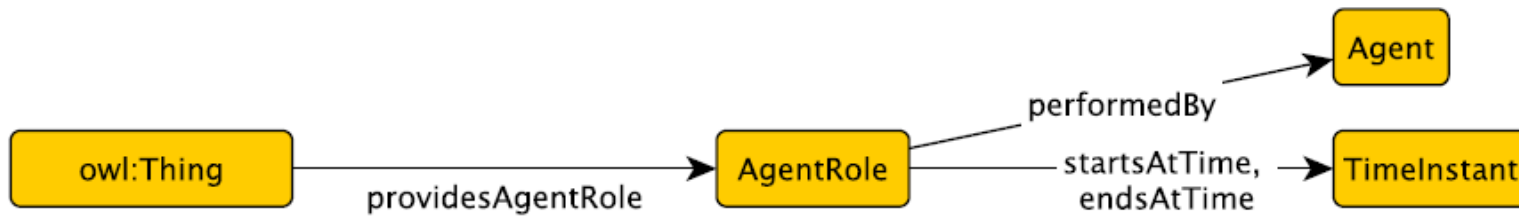


Pattern instantiation:



- 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]

The screenshot displays the XD plug-in interface with several components:

- ODP Selector:** Includes an ODP Category Selector (set to 'Any'), an ODP Search field with 'Query:' and 'Search'/'Reset' buttons, and a Results list with categories like 'ActingFor', 'Activity Pattern', 'Affordance', 'agent role', and 'Airline'.
- ODP Details:** Shows 'Use this Pattern' and 'Pattern Description' tabs. The 'WebVOWL Visualisation' tab displays a graphical representation of an ODP with nodes for 'or:Role' and 'or:Object' connected by 'or:isRoleOf' and 'class:isClassifiedBy' relationships.
- CODP Instantiation Wizard:** A dialog box with 'CODP Instantiation' and 'CODP Visualisation' tabs. It prompts the user to provide labels for ODP entities. The 'Classes' section shows 'Role' mapped to 'PersonRole' and 'Agent' to 'Person'. The 'Object Properties' section shows 'has role' mapped to 'hasPersonRole' and 'is role of' mapped to 'isPersonRoleOf'. The 'Datatype Properties' section shows 'topDataProperty' mapped to 'topDataProperty'.

# XD plug-in

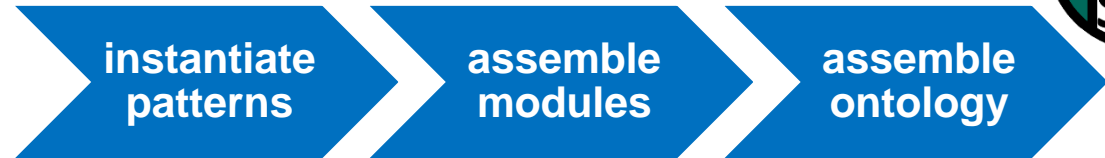
[Hammar]

The screenshot shows the XD plug-in interface. On the left is the **ODP Selector** panel with an **ODP Category Selector** set to 'Any', an **ODP Search** field with a 'Query:' label, and a **Results list** containing various categories like 'ActingFor', 'Affordance', and 'agent role'. The main area is the **ODP Details** panel, which includes a **Pattern Description** tab for 'WebVOWL Visualisation' and a **Graphical representation** section. This section displays a graph with two nodes: 'or:Role' (top) and 'or:Object' (bottom). The 'or:Role' node has a property 'or:isRoleOf : or:Object'. The 'or:Object' node has properties 'class:isClassifiedBy : or:Role' and 'or:hasRole : or:Role'. A blue arrow labeled 'class:isClassifiedBy' points from the 'or:Object' node to the 'or:Role' node.

The screenshot shows the **CODP Instantiation Wizard** dialog box. It has two tabs: **CODP Instantiation** and **CODP Visualisation**. Below the tabs is a **Generate preview** button. Underneath are two more tabs: **Axiom Preview** and **VOWL Preview**. The main area displays a graph with three nodes: 'PersonRole' (top), 'Person [RBJWDI2HaQ...]' (bottom), and 'Thing' (left). The 'PersonRole' node is connected to the 'Person' node by two edges labeled 'hasF' and 'isPersonR...'. The 'Thing' node is connected to the 'Person' node by two edges labeled 'Rckovb7...' and 'R99cZDs...'. At the bottom of the dialog are **Back**, **Finish**, and **Next** buttons.



## Module assembly:



- From instantiated patterns.
- Plus additional modifications/additions as needed.
- Modules may contain other modules.
- Provenance information (which pattern was used and 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]

The screenshot displays the XD plug-in interface. At the top, there are several tabs: WebProtege, test, Simplified editor, Advanced editor, Classes, Properties, Individuals, Changes By Entity, Project Dashboard, Design Patterns, and Visualization. The ODP Selector panel on the left includes an ODP Category Selector set to 'Any', an ODP Search field with a 'Query:' label, and a 'Results list' table. The ODP Details panel on the right shows a 'Graphical representation' of a pattern with two nodes: 'or:Role' (top) and 'or:Object' (bottom). The 'or:Role' node has a property 'or:isRoleOf :or:Object'. The 'or:Object' node has properties 'class:isClassifiedBy :or:Role' and 'or:hasRole :or:Role'. A blue arrow labeled 'or:isRoleOf' points from the 'or:Object' node to the 'or:Role' node. The CODP Instantiation Wizard dialog box is open in the foreground, showing the 'CODP Visualisation' tab. It contains a list of suggested entity alignments with checkboxes for selection.

**ODP Selector**

ODP Category Selector: Any

ODP Search: Query: [ ] Search Reset

**Results list**

Name
ActingFor
Activity Pattern
Affordance
agent role
Airline
AOS AGROVOC Con model
AquaticResourceObs
AquaticResources
Bag
BasicPlan
BasicPlanExecution
Born Digital Archives
CatchRecord
ChessGame
Classification

**ODP Details**

Use this Pattern: WebVOWL Visualisation

**Graphical representation**

or:Role (or:isRoleOf :or:Object)

or:Object (class:isClassifiedBy :or:Role, or:hasRole :or:Role)

**CODP Instantiation Wizard**

CODP Instantiation | CODP Visualisation

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**

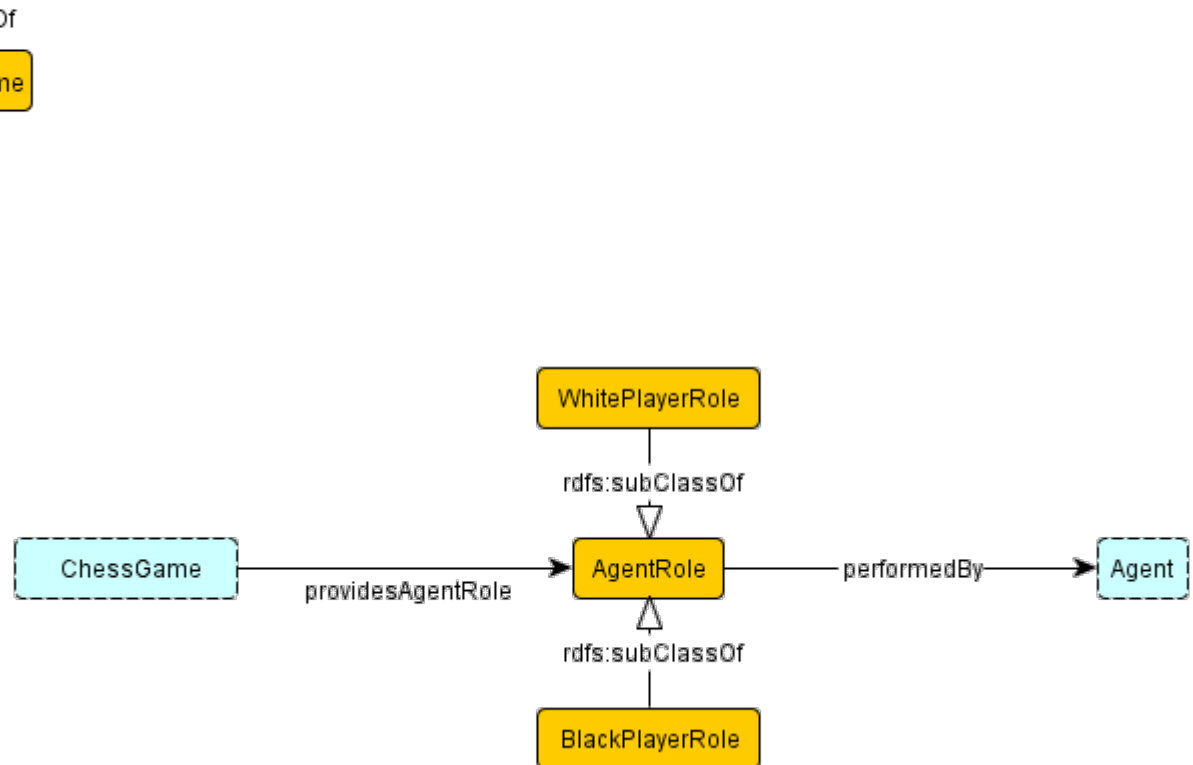
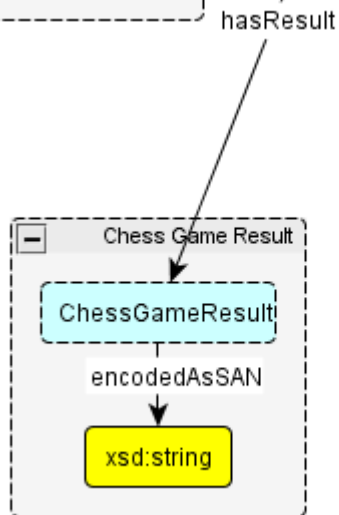
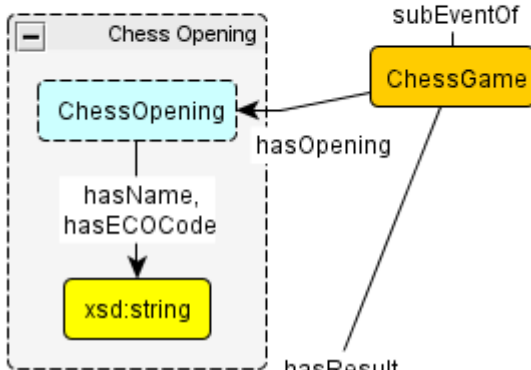
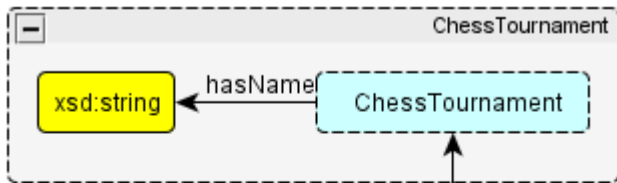
- CODP entity *Person* is equivalent to existing ontology entity *Person*
- CODP entity *Person* is a super-entity of existing ontology entity *Person*
- CODP entity *Person* is a sub-entity of existing ontology entity *Person*

**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*

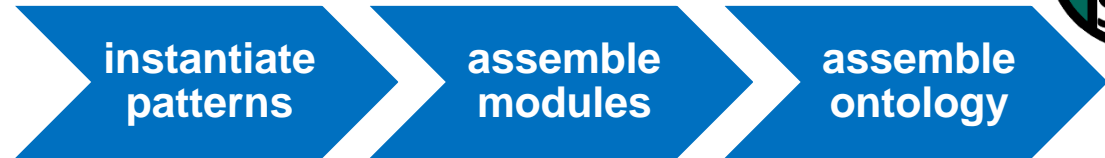
Back Finish Next

# Graphical mock-up





## Ontology assembly:



- 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.**



- **OWL<sub>Ax</sub> 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.

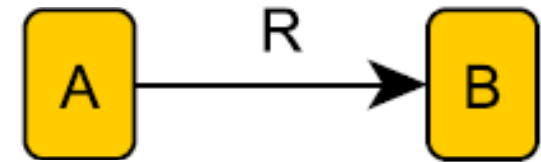
# Axioms – Systematically



1.  $A \sqcap B \sqsubseteq \perp$
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



The screenshot displays the OWLAx Protégé plug-in interface. The main window shows an ontology diagram with the following elements:

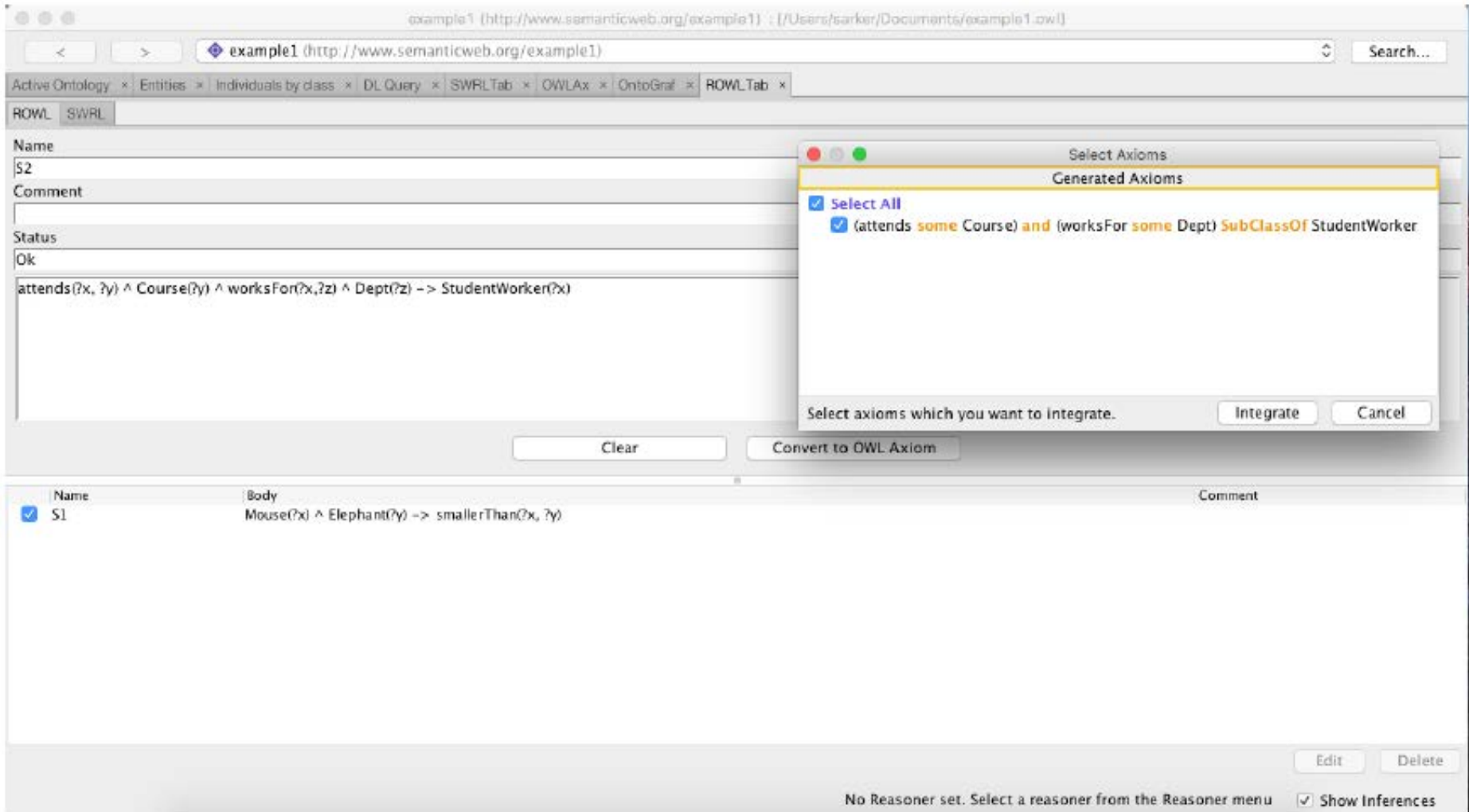
- Classes:** Disease, Person, ICD10Code, Dermatopythosis.
- Instances:** icd10:B35 (circled).
- Properties:** hasDisease, hasName, hasICD10Code, rdfs:subClassOf, rdf:type.
- Types:** xsd:string (yellow box).

The 'Select Axioms' dialog box is open, showing the following categories and selected items:

- Select All** (checked)
- SubClassOf Axioms** (checked)
  - onto:Dermatopythosis **SubClassOf** onto:Disease
- Disjoint Classes Axioms** (checked)
  - 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** (checked)
  - 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** (unchecked)
  - 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** (unchecked)
- Class (Type) Assertion Axioms** (checked)
  - icd10:B35 Type onto:ICD10Code

Buttons: Integrate, Cancel

# ROWL Protégé plug-in



example1 (http://www.semanticweb.org/example1) : [Users/sarker/Documents/example1.owl]

example1 (http://www.semanticweb.org/example1)

Active Ontology x Entities x Individuals by class x DL Query x SWRLTab x OWLAX x OntoGraf x ROWLTab x

ROWL SWRL

Name  
S2

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

Status  
Ok

Clear Convert to OWL Axiom

Select Axioms

Generated Axioms

Select All

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

Select axioms which you want to integrate. Integrate Cancel

Name	Body	Comment
<input checked="" type="checkbox"/> S1	Mouse(?x) ^ Elephant(?y) -> smallerThan(?x, ?y)	

Edit Delete

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

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

- 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 of modules and patterns within an ontology, and of their origins.
- Simple proposed solution: OPLa (Ontology Pattern Language):



**Pascal Hitzler, Aldo Gangemi, Krzysztof 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!**



**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.**

**A. Gangemi. Ontology design patterns for semantic web content. In Y. Gil et al. (eds), The Semantic Web - ISWC 2005 – 4<sup>th</sup> 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 - 14<sup>th</sup> 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.**





**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.**



**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**