



OOPS!

OntOlogy Pitfall Scanner! and linguistic resources

María Poveda

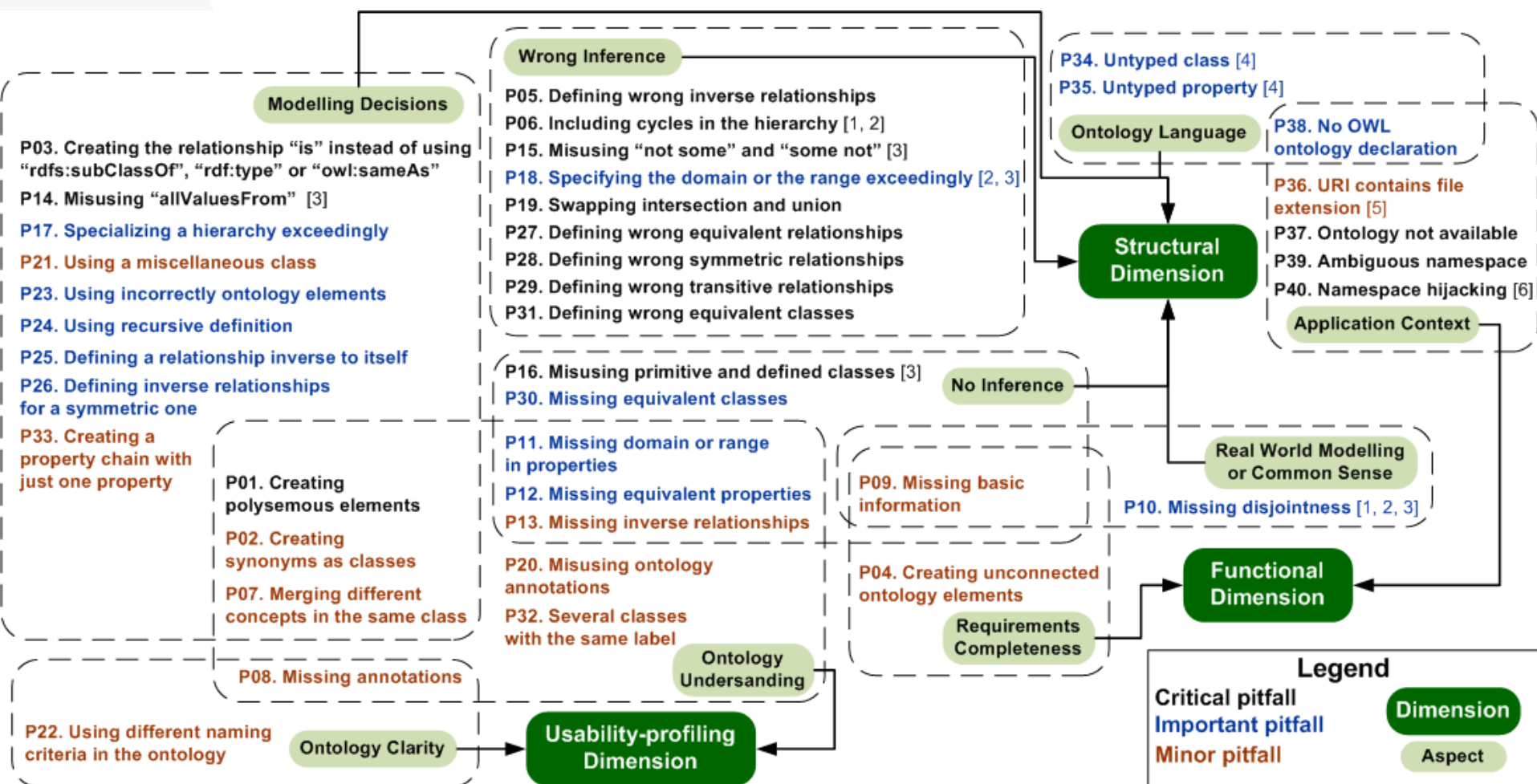
email: mpoveda@fi.upm.es



- <http://www.oeg-upm.net/oops>
- Web-based
- Independent of any ontology development environment
- Detect PITFALLS (**potential** errors or problems when modelling ontologies)
 - Looking for architectural patterns (topology based)
 - *Analysing lexical information*
 - Seeking particular ontology characteristics
- OWL ontologies (for now)

- Pitfalls **could represent** or **lead to** an error.
- Pitfalls are **not necessarily** errors. For example, pitfalls **might not** represent an error depending on:
 - Modelling decisions.
 - Context or scope of the ontology.
 - Ontology requirements.
- In addition not all the pitfalls are equally important.

Pitfall Catalogue classification by dimensions



- [1] Gómez-Pérez, A. "Ontology Evaluation". Handbook on Ontologies. S. Staab and R. Studer Editors. Springer. International Handbooks on Information Systems. Pp: 251-274. 2004.
- [2] Noy, N.F., McGuinness. D. L. "Ontology development 101: A guide to creating your first ontology." Technical Report SMI-2001-0880, Stanford Medical Informatics. 2001.
- [3] Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C. "Owl pizzas: Practical experience of teaching owl-dl: Common errors and common patterns". In Proc. of EKAW 2004, pp: 63-81. Springer. 2004.
- [4] Hogan, A., Harth, A., Passant, A., Decker, S., Polleres, A. Weaving the Pedantic Web. Linked Data on the Web Workshop LDOW2010 at WWW2010 (2010).
- [5] Archer, P., Goedertier, S., and Loutas, N. D7.1.3 – Study on persistent URIs, with identification of best practices and recommendations on the topic for the MSs and the EC. Deliverable. 2012.
- [6] Heath, T., Bizer, C.: Linked data: Evolving the Web into a global data space (1st edition). Morgan & Claypool (2011).

OOPS! - Web user interface - Homepage



Ontology Pitfall Scanner!

OOPS! (Ontology Pitfall Scanner!) helps you to detect some of the most common pitfalls appearing when developing ontologies.

To try it, enter a URI or paste an OWL document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

Scanner by URI:

Scanner by URI

Example: http://data.semanticweb.org/ns/swc/swc_2009-05-09.rdf

If you just include the RDF code here, the following Pitfalls will not be checked: P36. URI contains file extension, P37. Ontology not available, P40. Namespace hijacking

Scanner by direct input:

Scanner by RDF

[Go to advanced evaluation](#)

News!

Now you can **integrate OOPS!** pitfall detection **with your own developments and tools** simply by invoking the [OOPS! RESTful Web Service](#).

Detecting common pitfalls in ontologies

Modelling ontologies has become one of the main topics of research within ontological engineering because of the difficulties it involves. Developers must tackle a wide range of difficulties and handicaps when modelling ontologies that can imply the appearance of anomalies or errors in ontologies. Therefore, it is important to evaluate the ontologies in order to detect those potential problems.

In this sense, OOPS! helps you to detect some of the most common pitfalls appearing within ontology developments. For example, OOPS! warns you when:

- The domain or range of a relationship is defined as the intersection of two or more classes. This warning could avoid reasoning problems in case those classes could not share instances.
- No naming convention is used in the identifiers of the ontology elements. In this case the maintainability, the accessibility and the clarity of the ontology could be improve.
- A cycle between two classes in the hierarchy is included in the ontology. Detecting this situation could avoid modelling and reasoning problems.
- And many other problems described in [the catalogue](#).

Please, help us making OOPS! better. [Feedback](#) is more than welcome and you can also [suggest new pitfalls!](#)

Input area

Advanced options

General information

Feedback

Documentation






Publications

Web Service

OOPS! - Web user interface - Results

Evaluation results

[Expand All] | [Collapse All]

Pitfall name	Pitfall frequency	Importance Level
Results for P04: Creating unconnected ontology elements.	11 cases	Minor 
Results for P05: Defining wrong inverse relationships.	2 cases	Critical 
Two relationships are defined as inverse relations when they are not necessarily. For example, something is sold or something is bought; in this case, the relationships "isSoldIn" and "isBoughtIn" are not inverse.		
<ul style="list-style-type: none">This pitfall appears in the following elements:<ul style="list-style-type: none">> http://data.semanticweb.org/ns/swc/ontology#relatedToEvent may not be inverse of http://data.semanticweb.org/ns/swc/ontology#hasRelatedDocument> http://data.semanticweb.org/ns/swc/ontology#hasRelatedDocument may not be inverse of http://data.semanticweb.org/ns/swc/ontology#relatedToEvent		
Results for P08: Missing annotations.	156 cases	Minor 
Results for P11: Missing domain or range in properties.	83 cases	Important 
Results for P12: Missing equivalent properties.	8 cases	Important 
When an ontology is imported into another, classes that are duplicated in both ontologies are normally defined as equivalent classes. However, the ontology developer misses the definition of equivalent properties in those cases of duplicated relationships and attributes. For example, the classes "CITY" and "City" in two different ontologies are defined as equivalent classes; however, relationships "hasMember" and "has-Member" in two different ontologies are not defined as equivalent relations.		
<ul style="list-style-type: none">The following relations could be defined as equivalent:<ul style="list-style-type: none">> http://swrc.ontoware.org/ontology#member, http://xmlns.com/foaf/0.1/member> http://purl.org/dc/terms/isPartOf, http://data.semanticweb.org/ns/swc/ontology#isPartOfThe following attributes could be defined as equivalent:<ul style="list-style-type: none">> http://swrc.ontoware.org/ontology#title, http://xmlns.com/foaf/0.1/title> http://purl.org/dc/elements/1.1/source, http://swrc.ontoware.org/ontology#source> http://www.w3.org/2002/12/cal/ical#location, http://swrc.ontoware.org/ontology#location		

Importance Level

Pitfall description

Ontology elements affected

Pitfall Catalogue classification by importance level

CRITICAL (1)

P01. Creating polysemous elements

P03. Creating the relationship “is” instead of using "rdfs:subClassOf", "rdf:type" or "owl:sameAs"

P05. Defining wrong inverse relationships

P06. Including cycles in the hierarchy

P14. Misusing "owl:allValuesFrom"

P15. Misusing “not some” and “some not”

P16. Misusing primitive and defined classes

P19. Swapping intersection and union

P27. Defining wrong equivalent relationships

P28. Defining wrong symmetric relationships

P29. Defining wrong transitive relationships

P31. Defining wrong equivalent classes

P37. Ontology not available

P39. Ambiguous namespace

P40. Namespace hijacking

LEGEND

Implemented pitfall

Not implemented pitfall

IMPORTANT (2)

P10. Missing disjointness

P11. Missing domain or range in properties

P12. Missing equivalent properties

P17. Specializing a hierarchy exceedingly

P18. Specifying the domain or range exceedingly

P23. Using incorrectly ontology elements

P24. Using recursive definition

P25. Defining a relationship inverse to itself

P26. Defining inverse relationships for a symmetric one

P30. Missing equivalent classes

P34. Untyped class

P35. Untyped property

P38. No OWL ontology declaration

MINOR (3)

P02. Creating synonyms as classes

P04. Creating unconnected ontology elements

P07. Merging different concepts in the same class

P08. Missing annotations

P09. Missing basic information

P13. Missing inverse relationships

P20. Misusing ontology annotations

P21. Using a miscellaneous class

P22. Using different naming criteria in the ontology

P32. Several classes with the same label

P33. Creating a property chain with just one property

P36. URI contains file extension

Pitfall Catalogue classification by technique and cardinality

<div>Cardinality</div> <div>Technique</div>	0..1 Appears at most once	0..N Could appear more than once
Structural pattern based	P10 <div>P03</div>	P02, P04, P05, P06, P08, P11, P13, P19, P24, P25, P26, P27, P28, P29, P33, P34, P35 <div>P12, P20, P30, P31, P32</div>
Lexical content analysis	<div>P03</div> P22	<div>P12, P20, P30, P31, P32</div> P07, P21
Other characteristics	P36, P37, P38, P39	P40

P05. Defining wrong inverse relationships

P30. Missing equivalent classes

P37. Ontology not available

P07. Merging different concepts in the same class

OOPS! and linguistic resources

Improvement of existing detections

- P03. Creating the relationship “is” instead of using "rdfs:subClassOf", "rdf:type" or "owl:sameAs"
- P05. Defining wrong inverse relationships
- P07. Merging different concepts in the same class
- P13. Missing inverse relationships
- P21. Using a miscellaneous class

**Lexico-syntactic
patterns**

- P02. Creating synonyms as classes
- P30. Missing equivalent classes
- P31. Defining wrong equivalent classes

Linguistic resources

- P12. Missing equivalent properties

**Linguistic resources
+
Lexico-syntactic patterns**

Automation of pitfalls not yet detected

- P01. Creating polysemous elements
- P09. Missing basic information

Linguistic resources

1. M. Poveda-Villalón, M.C. Suárez-Figueroa, A. Gómez-Pérez. *Validating ontologies with OOPS!*. Accepted as long research paper at the 18th International Conference on Knowledge Engineering and Knowledge Management. 8 - 12 October 2012, Galway, Ireland.
2. M.C. Suárez-Figueroa, M. Kamel, M. Poveda-Villalón. *Benefits of Natural Language Techniques in Ontology Evaluation: the OOPS! Case*. Accepted as short paper at 10th International Conference on Terminology and Artificial Intelligence (TIA 2013). 28-30 October 2013, Paris, France.



**Thank you for
your attention!**