OntologySummit2013: Thursday 2013-01-31
Summit Theme: **"Ontology Evaluation Across the Ontology Lifecycle"**
Summit Track Title: **Track-A: Intrinsic Aspects of Ontology Evaluation**
Session Topic: **Intrinsic Aspects of Ontology Evaluation: Practice and Theory**

A PITFALL CATALOGUE AND OOPS!: AN APPROACH TO ONTOLOGY VALIDATION

María Poveda-Villalón, Mari Carmen Suárez-Figueroa and Asunción Gómez-Pérez
Ontology Engineering Group. Departamento de Inteligencia Artificial.
Facultad de Informática, Universidad Politécnica de Madrid.
Campus de Montegancedo s/n.
28660 Boadilla del Monte. Madrid. Spain
{**mpoveda**, mcsuarez, asun}@fi.upm.es

Table of Contents

- **Introduction**
- Pitfall Catalogue
- OOPS! (OntOlogy Pitfall Scanner!)
- Conclusions and Future Work

Introduction (i)

Methodologies (e.g: Methontology [1, 2], On-To-Knowledge [3], DILIGENT [4], and the NeOn Methodology [5]) that support the **ontology development** transformed the art of building ontologies into an **engineering activity**.

The correct application of such **methodologies benefits** the **ontology quality**.

However

Developers must tackle a wide range of **difficulties** and **handicaps** when **modelling ontologies**.

These difficulties can imply the appearance of **anomalies** or **worst practices** in ontologies.

Ontology evaluation (checking the technical quality of an ontology against a frame of reference) is a **crucial activity** in ontology engineering projects.

[1] Gómez-Pérez, A., Fernández-López, M., Corcho, O. Ontological Engineering. November 2003. Springer Verlag. Advanced Information and Knowledge Processing series. ISBN 1-85233-551-3.

[2] M. Fernández-López, A. Gómez-Pérez, N. Juristo. METHONTOLOGY: From Ontological Art Towards Ontological Engineering. 1997. Spring Symposium on Ontological Engineering of AAAI. Stanford University, California, pp 33–40.

[3] S. Staab, H.P. Schnurr, R. Studer, Y. Sure. Knowledge Processes and Ontologies. IEEE Intelligent Systems 16(1):26–34. (2001).

[4] H. S. Pinto, C. Tempich, S. Staab. *DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies*. In Ramón López de Mantaras and Lorenza Saitta, Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), August 22nd - 27th, pp. 393--397. IOS Press, Valencia, Spain, August 2004. ISBN: 1-58603-452-9. ISSN: 0922-6389.

[5] M.C. Suárez-Figueroa. Doctoral Thesis: NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse. Spain. Universidad Politécnica de Madrid. June 2010.

Introduction (ii)

A lot of work has been done in ontology evaluation:

- generic quality evaluation frameworks [1, 2, 3, 4, 5],
- methods based on the final (re)use of the ontology [6]
- quality models based on features, criteria and metrics [7, 8]
- tools: ODEclean, ODEval, XDTools, OntoCheck, EyeBall, MoKi, etc.

However

Ontology evaluation is still largely neglected by developers and practitioners

Maybe because of?

(a) the current **time-consumin**g and **tedious** nature of evaluating the quality of an ontology

(b) the **lack** of **awareness** of the necessity for evaluating ontologies we are producing and publishing throughout the web.

[1] Welty, C.A., and Guarino, N. Supporting ontological analysis of taxonomic relationships. In Data & Knowledge Engineering. vol 39, pp 51-74. 2001

[2] Duque-Ramos, A., Uriel López, J. T. Fernández-Breis, Robert Stevens. *Towards an SQUaRE-based Quality Evaluation Framework for Ontologies*. OntoQual 2010 - Workshop on Ontology Quality at EKAW 2010) ISBN: ISSN 1613-0073. Pages: 13-24. 15 October 2010. Lisbon, Portugal.

[3] Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann J. *Modelling Ontology Evaluation* and *Validation.* Proceedings of ESWC2006, number 4011 in LNCS, Budva. 2006.

[4] Gómez-Pérez, A. *Ontology Evaluation*. Handbook on Ontologies. S. Staab and R. Studer Editors. Springer. International Handbooks on Information Systems. Pp: 251-274. 2004.

[5] Strasunskas, D., Tomassen, S.L.: *The role of ontology in enhancing semantic searches: the EvOQS framework and its initial validation*. Int. J. Knowledge and Learning, Vol. 4, No. 4, pp. 398-414.

[6] Suárez-Figueroa, M.C. Doctoral Thesis: NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse. Spain. Universidad Politécnica de Madrid. June 2010.

[7] Flemming, A.. Assessing the quality of a Linked Data source. Proposal. http://www2.informatik.hu-berlin.de/~flemming/Proposal.pdf

[8] Burton-Jones, A., Storey, V.C., and Sugumaran, V., and Ahluwalia, P. A Semiotic Metrics Suite for Assessing the Quality of Ontologies. Data and Knowledge Engineering, (55:1) 2005, pp. 84-102.

Introduction (iii)

Our objective is

- To **ease** the activity of **ontology evaluation** (mainly people who are not ontological engineers)
- To **reduce time** and **effort** in **ontology evaluation**
- Mainly **focused** on **technology transfer** in **enterprises**

For doing so we have

(a) **Created** a **catalogue** contained potential errors we have seen in other ontologies and other authors' work

(b) Established mechanisms to keep this **catalogue updated** and **maintained**

(c) **Automated** the detection of several errors

(d) **Provided** a web-based **user interface**

Table of Contents

- Introduction
- **Pitfall Catalogue**
- OOPS! (OntOlogy Pitfall Scanner!)
- Conclusions and Future Work

Pitfall Catalogue - Origin and Maintenance

Origin

- **Manual inspection** of 26 ontologies (students from the master on artificial intelligence at UPM)
- **Well known problems** described by other authors [Gómez-Pérez, 2004; Noy and McGuinness, 2001; Rector, et al., 2004]
- **First** pitfall catalogue **version** published [Poveda-Villalón, et al., 2010]
- **24** pitfalls described

Maintenance

- Including new pitfalls:
  - ○ **Discovered** while manually analyzing ontologies
  - ○ **Proposed by users** (http://www.oeg-upm.net/oops/submissions.jsp)
- **Current** version contains **35** pitfalls
- **11** new pitfalls (P25-P35)
- **Survey** on ontology pitfall importance

(https://docs.google.com/spreadsheet/viewform?formkey=dFBqT1N1a3dHQWZ2SjJOeG41OTliaXc6MQ#gid=0)

Gómez-Pérez, A. "Ontology Evaluation". Handbook on Ontologies. S. Staab and R. Studer Editors. Springer. International Handbooks on Information Systems. Pp: 251-274. 2004.

Noy, N.F., McGuinness. D. L. "Ontology development 101: A guide to creating your first ontology." Technical Report SMI-2001-0880, Standford Medical Informatics. 2001.

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.

M. Poveda-Villalón, M.C. Suárez-Figueroa, A. Gómez-Pérez. *A Double Classification of Common Pitfalls in Ontologies.* OntoQual 2010 - Workshop on Ontology Quality at EKAW 2010. Proceedings of the Workshop on Ontology Quality - OntoQual 2010

Pitfall Catalogue so far

Human understanding

- P1. Creating polysemous elements
- P2. Creating synonyms as classes
- P7. Merging different concepts in the same class
- P8. Missing annotations
- P11. Missing domain or range in properties
- P12. Missing equivalent properties
- P13. Missing inverse relationships
- P19. Swapping intersection and union
- P20. Misusing ontology annotations
- P22. Using different naming criteria in the ontology
- P30. Missing equivalent classes
- P32. Several classes with the same label

Logical consistency

- P5. Defining wrong inverse relationships
- P6. Including cycles in the hierarchy
- P14. Misusing "owl:allValuesFrom"
- P15. Misusing “not some” and “some not”
- P18. Specifying too much the domain or the range
- 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
- P33. Creating a property chain with just one property

Real world representation

- P5. Defining wrong inverse relationships
- P9. Missing basic information
- P10. Missing disjointness
- P27. Defining wrong equivalent relationships
- P28. Defining wrong symmetric relationships
- P29. Defining wrong transitive relationships

Modelling issues

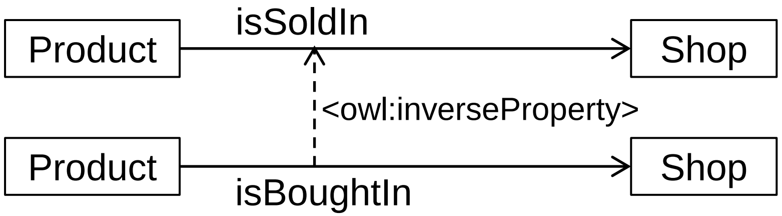
- P2. Creating synonyms as classes
- P3. Creating the relationship “is” instead of using "rdfs:subClassOf", "rdf:type" or "owl:sameAs"
- P4. Creating unconnected ontology elements
- P5. Defining wrong inverse relationships
- P6. Including cycles in the hierarchy
- P7. Merging different concepts in the same class
- P10. Missing disjointness
- P17. Specializing too much a hierarchy
- P11. Missing domain or range in properties
- P12. Missing equivalent properties
- P13. Missing inverse relationships
- P14. Misusing "owl:allValuesFrom"
- P15. Misusing “not some” and “some not”
- P18. Specifying too much the domain or the range
- P19. Swapping intersection and union
- P21. Using a miscellaneous class
- P23. Using incorrectly ontology elements
- P24. Using recursive definition
- P25. Defining a relationship inverse to itself
- P26. Defining inverse relationships for a symmetric one
- P27. Defining wrong equivalent relationships
- P28. Defining wrong symmetric relationships
- P29. Defining wrong transitive relationships
- P30. Missing equivalent classes
- P31. Defining wrong equivalent classes
- P32. Several classes with the same label
- P33. Creating a property chain with just one property

Ontology language specification

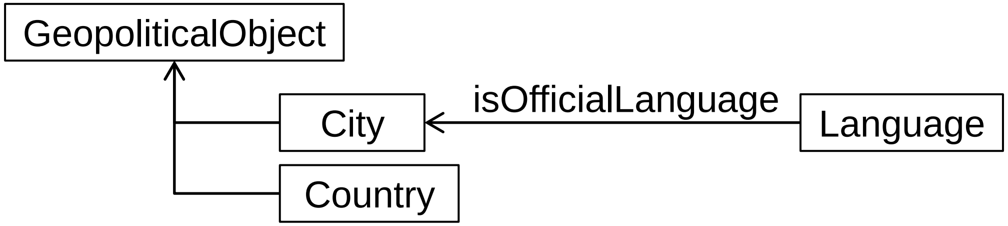
- P34. Untyped class
- P35. Untyped property

Examples

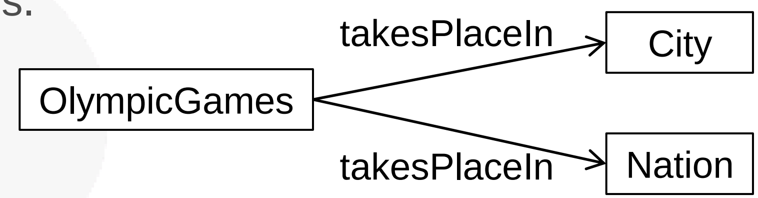
- **P5. Defining wrong inverse relationships:** two relationships are defined as inverse relations when they are not necessarily.



- **P18. Specifying too much the domain or the range:** not to find a domain or a range that is general enough.



- **P19. Swapping intersection and union:** the ranges and/or domains of the properties (relationships and attributes) are defined by intersecting several classes in cases in which the ranges and/or domains should be the union of such classes.



Pitfall Catalogue (important) Notes

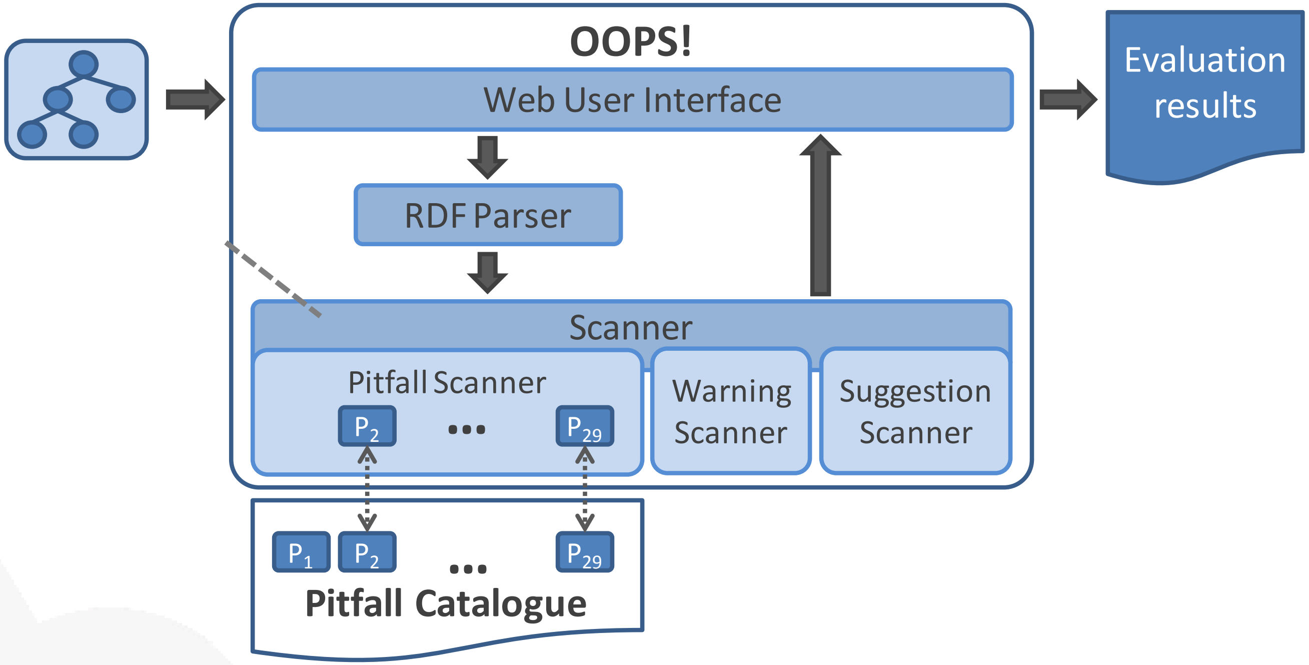
- Pitfalls **could represent** or **lead to** an error.
- Pitfalls are **not necessarily** errors. For example, pitfalls **might not** represent an error depending on:
  - Modelling decisions.
  - Context or scope of the ontology.
  - Ontology requirements.
- In addition not all the pitfalls are equally important.
- **Ongoing work**: associate an **indicator** to each pitfall according to their **possible** negative consequences
  - **Survey** on ontology pitfalls importance: https://docs.google.com/spreadsheet/viewform?formkey=dFBqT1N1a3dHQWZ2SjJOeG41OTliaXc6MQ#gid=0

Table of Contents

- Introduction
- Pitfall Catalogue
- **OOPS! (OntOlogy Pitfall Scanner!)**
- Conclusions and Future Work

OOPS! - How it is internally organized (i)

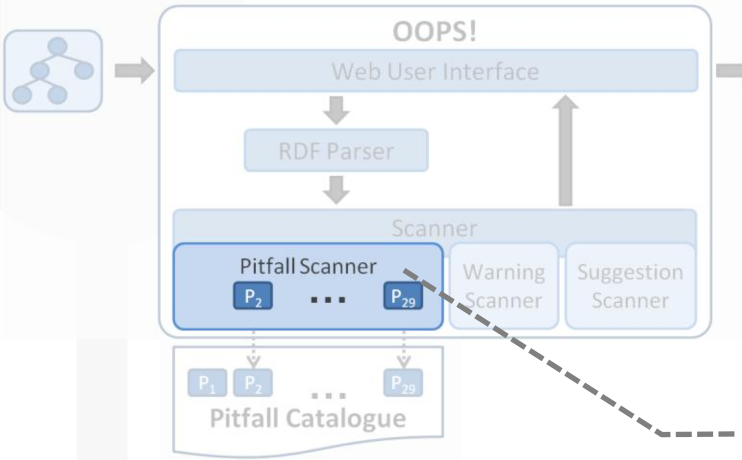
- **Web-based** tool
- Available at **http://www.oeg-upm.net/oops**
- Ontology development environment **independent**
- **No installation** process required



Jena API: http://jena.sourceforge.net/
Java EE: http://www.oracle.com/technetwork/java/javaee/overview/index.html
HTML: http://www.w3.org/html/wg/

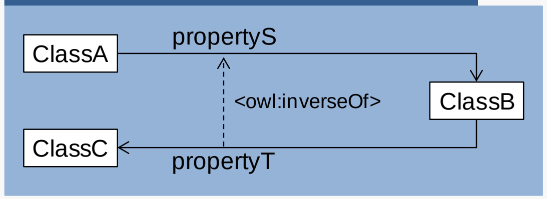
jQuery: http://jquery.com/
JSP: http://www.oracle.com/technetwork/java/javaee/jsp/index.html
CSS: http://www.w3.org/Style/CSS/

OOPS! - How it is internally organized (ii)

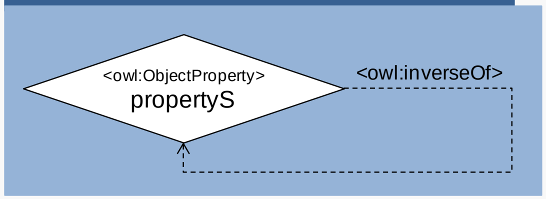


- **21** pitfalls **implemented** out of 35 included in the catalogue
- 1 Java class per pitfall implementation
- **Detection** automated in 2 ways:
  - **Checking general characteristics** of the ontology (P3, P7, P12, P20, P21, and P22). Eg: P 22. Using more than one naming convention.
  - Looking for **patterns** (P2, P4, **P5**, **P6**, P8, P10, P11, P13, P19, P24, **P25**, P26, P27, **P28**, and P29). Eg: P5: Defining wrong inverse relationships

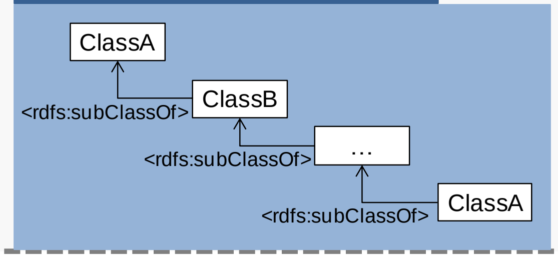
P5. Defining wrong inverse relationships



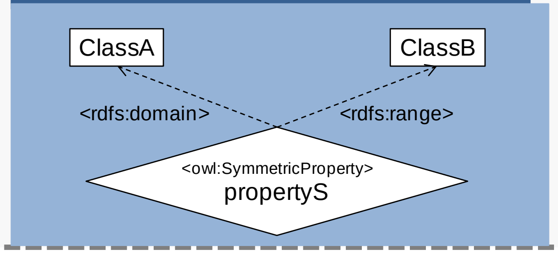
P25. Defining a relationship inverse to itself



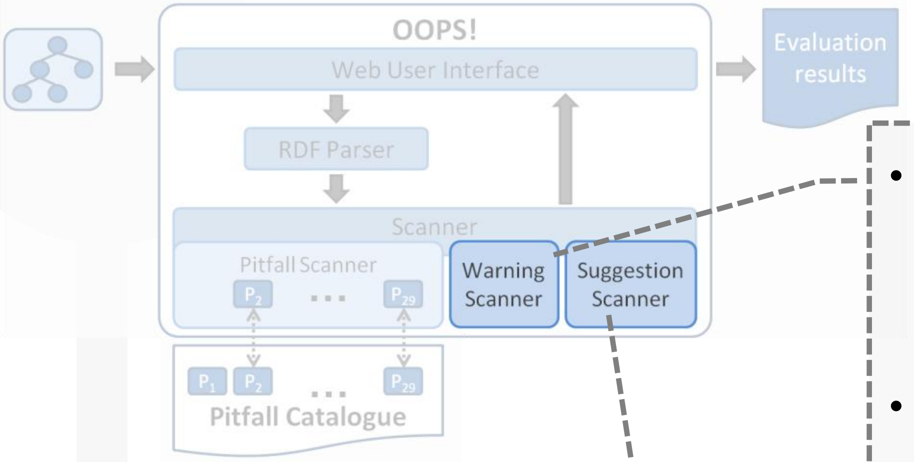
P6. Including cycles in the hierarchy



P28. Defining wrong symmetric relationships



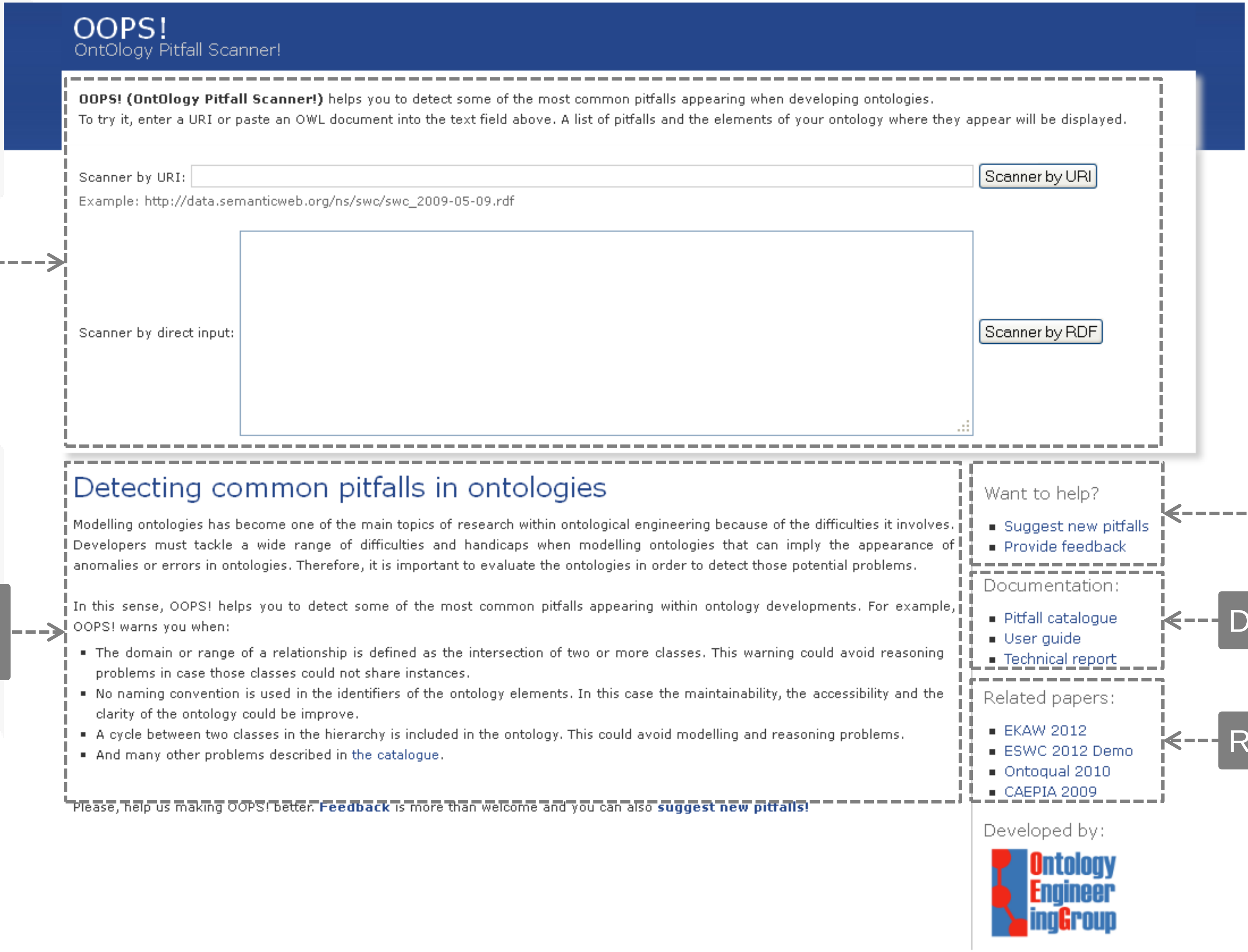
OOPS! - How it is internally organized (iii)



- Identifies cases where a **class** or **property** is **not defined** as such by means of the corresponding **OWL primitive**.
- It is spotted **during** the execution of the "**Pitfall Scanner" module**.
- **Only** the classes and relationships **related to** the **other pitfalls** detection are **flag up**.

- Looks for **properties** with **equal domain and range** axioms and proposes them as potential symmetric or transitive properties.

OOPS! - How it works (i)



OOPS!
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

Scanner by direct input: [Scanner by RDF]

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. This 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!**

Want to help?

- Suggest new pitfalls
- Provide feedback

Documentation:

- Pitfall catalogue
- User guide
- Technical report

Related papers:

- EKAW 2012
- ESWC 2012 Demo
- Ontoqual 2010
- CAEPIA 2009

Developed by:

Ontology Engineering Group

Ontology input area

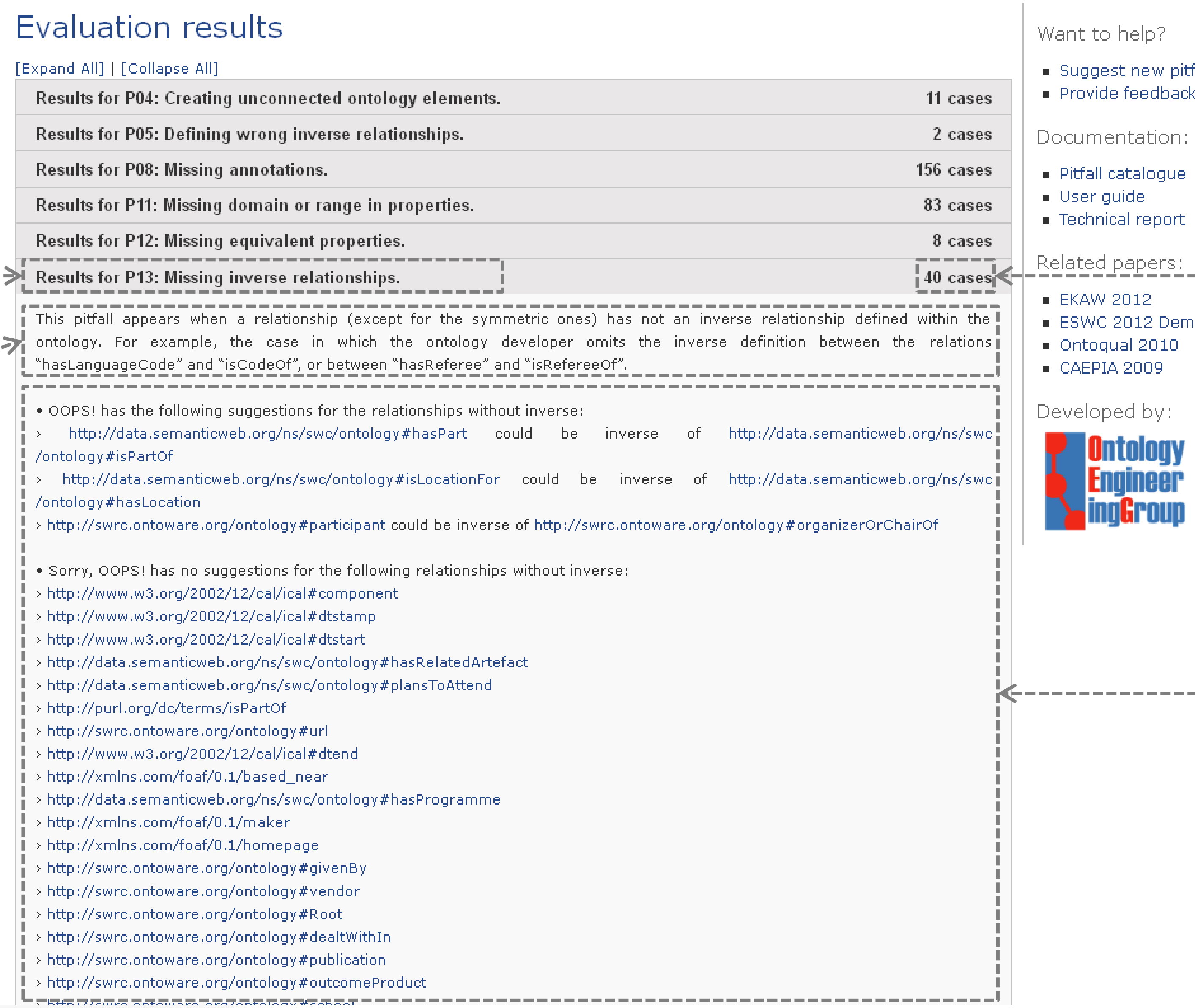
Brief description

Suggestions & feedback

Documentation

Related papers

OOPS! - How it works (ii)



Evaluation results

[Expand All] | [Collapse All]

Results for P04: Creating unconnected ontology elements. 11 cases

Results for P05: Defining wrong inverse relationships. 2 cases

Results for P08: Missing annotations. 156 cases

Results for P11: Missing domain or range in properties. 83 cases

Results for P12: Missing equivalent properties. 8 cases

Results for P13: Missing inverse relationships. 40 cases

This pitfall appears when a relationship (except for the symmetric ones) has not an inverse relationship defined within the ontology. For example, the case in which the ontology developer omits the inverse definition between the relations "hasLanguageCode" and "isCodeOf", or between "hasReferee" and "isRefereeOf".

- OOPS! has the following suggestions for the relationships without inverse:
  - › http://data.semanticweb.org/ns/swc/ontology#hasPart could be inverse of http://data.semanticweb.org/ns/swc/ontology#isPartOf
  - › http://data.semanticweb.org/ns/swc/ontology#isLocationFor could be inverse of http://data.semanticweb.org/ns/swc/ontology#hasLocation
  - › http://swrc.ontoware.org/ontology#participant could be inverse of http://swrc.ontoware.org/ontology#organizerOrChairOf

- Sorry, OOPS! has no suggestions for the following relationships without inverse:
  - › http://www.w3.org/2002/12/cal/ical#component
  - › http://www.w3.org/2002/12/cal/ical#dtstamp
  - › http://www.w3.org/2002/12/cal/ical#dtstart
  - › http://data.semanticweb.org/ns/swc/ontology#hasRelatedArtefact
  - › http://data.semanticweb.org/ns/swc/ontology#plansToAttend
  - › http://purl.org/dc/terms/isPartOf
  - › http://swrc.ontoware.org/ontology#url
  - › http://www.w3.org/2002/12/cal/ical#dtend
  - › http://xmlns.com/foaf/0.1/based_near
  - › http://data.semanticweb.org/ns/swc/ontology#hasProgramme
  - › http://xmlns.com/foaf/0.1/maker
  - › http://xmlns.com/foaf/0.1/homepage
  - › http://swrc.ontoware.org/ontology#givenBy
  - › http://swrc.ontoware.org/ontology#vendor
  - › http://swrc.ontoware.org/ontology#Root
  - › http://swrc.ontoware.org/ontology#dealtWithIn
  - › http://swrc.ontoware.org/ontology#publication
  - › http://swrc.ontoware.org/ontology#outcomeProduct

Want to help?

- Suggest new pitfalls
- Provide feedback

Documentation:

- Pitfall catalogue
- User guide
- Technical report

Related papers:

- EKAW 2012
- ESWC 2012 Demo
- Ontoqual 2010
- CAEPIA 2009

Developed by:

Pitfall name

Pitfall description

Pitfall frequency

Ontology elements affected

Example generated using the ontology http://data.semanticweb.org/ns/swc/swc_2009-05-09.rdf

Table of Contents

- Introduction
- Pitfall Catalogue
- OOPS! (OntOlogy Pitfall Scanner!)
- **Conclusions and Future Work**

Conclusions and Future Work (i)

Conclusions	
Catalogue	• Currently **35** pitfalls including other authors' work • **Maintained** and open to users/experts/practitioners point of view (**collaborative**)
OOPS! OntOlogy Pitfall Scanner!	• It is **freely available** to users on the Web: http://www.oeg-upm.net/oops ○ is fully **independent** of any ontology **development environment .** ○ works with **main web browsers** (Firefox, Chrome, Safari and IE). ○ does **not** involve **installation** process. • Everyone can test it, provide **feedback**, **suggest new pitfalls** to be included in the catalogue and implemented into the tool. ○ easy to use ○ broadly used • **>800** executions • **>300** different ontologies • from 14th November 2011 to 29th January 2013 • feedback from a number of users by emails and feedback form

For further details see: M. Poveda-Villalón, M.C. Suárez-Figueroa, A. Gómez-Pérez. *Validating ontologies with OOPS!*. 18th International Conference on Knowledge Engineering and Knowledge Management (EKAW2012). 8 - 12 October 2012, Galway, Ireland. ISBN:978-3-642-33875-5

Conclusions and Future Work (ii)

Future Work	
Catalogue	• Continuous maintenance • To associate an **indicator** to each pitfall according to their **possible** negative consequences (coming soon) • To include **guidelines** about how **to solve** each pitfall
OOPS! OntOlogy Pitfall Scanner!	• To create an specialized version of **OOPS! for Linked Data** use case ○ Developers are often domain experts ○ No or little ontology knowledge support ○ Little time/resources to evaluate the vocabularies ○ Mainly lightweight vocabularies (lack of axioms) ○ Specialized requirements for web ontologies (e.g: derreferenciability) • To create an **access point** to more complex ontology evaluation techniques ○ Up to now we point to another papers ○ To incorporate other tools into OOPS! (e.g: eyeball) ○ To list and classify methods and tools ▪ To point to other methods (e.g: OntoClean) or tools (e.g: ontocheck) • **Web Services** (coming soon, under testing) • To allow pitfalls **definition** following a **formal language**, according with their particular quality criteria

OntologySummit2013: Thursday 2013-01-31
Summit Theme: **"Ontology Evaluation Across the Ontology Lifecycle"**
Summit Track Title: **Track-A: Intrinsic Aspects of Ontology Evaluation**
Session Topic: **Intrinsic Aspects of Ontology Evaluation: Practice and Theory**

A PITFALL CATALOGUE AND OOPS!: AN APPROACH TO ONTOLOGY VALIDATION

María Poveda-Villalón, Mari Carmen Suárez-Figueroa and Asunción Gómez-Pérez
Ontology Engineering Group. Departamento de Inteligencia Artificial.
Facultad de Informática, Universidad Politécnica de Madrid.
Campus de Montegancedo s/n.
28660 Boadilla del Monte. Madrid. Spain
{**mpoveda**, mcsuarez, asun}@fi.upm.es