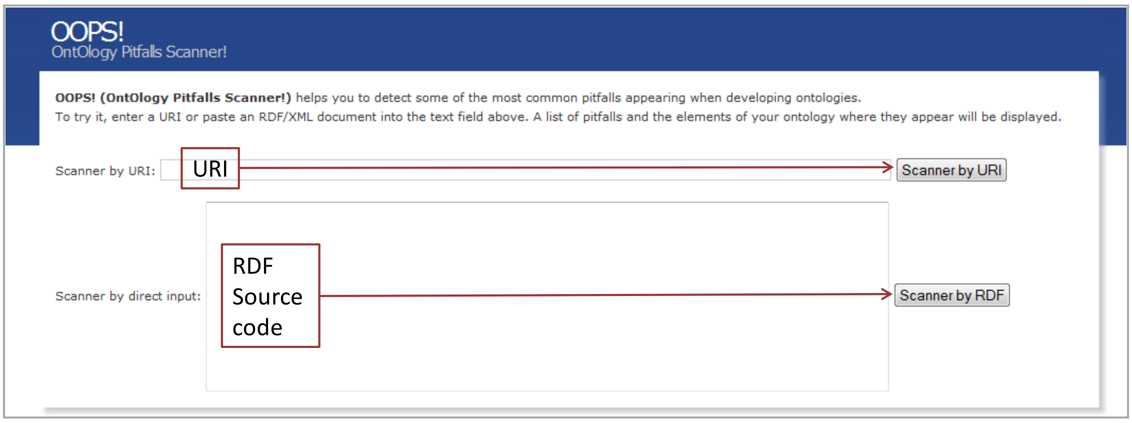
OOPS! User guide

Getting started

OOPS! is a web application that helps you to detect some of the most common pitfalls appearing when developing ontologies.

To use it, you just need to enter the URI or the RDF code of the ontology you would like to analyze and click on the corresponding button as Figure 1 shows.

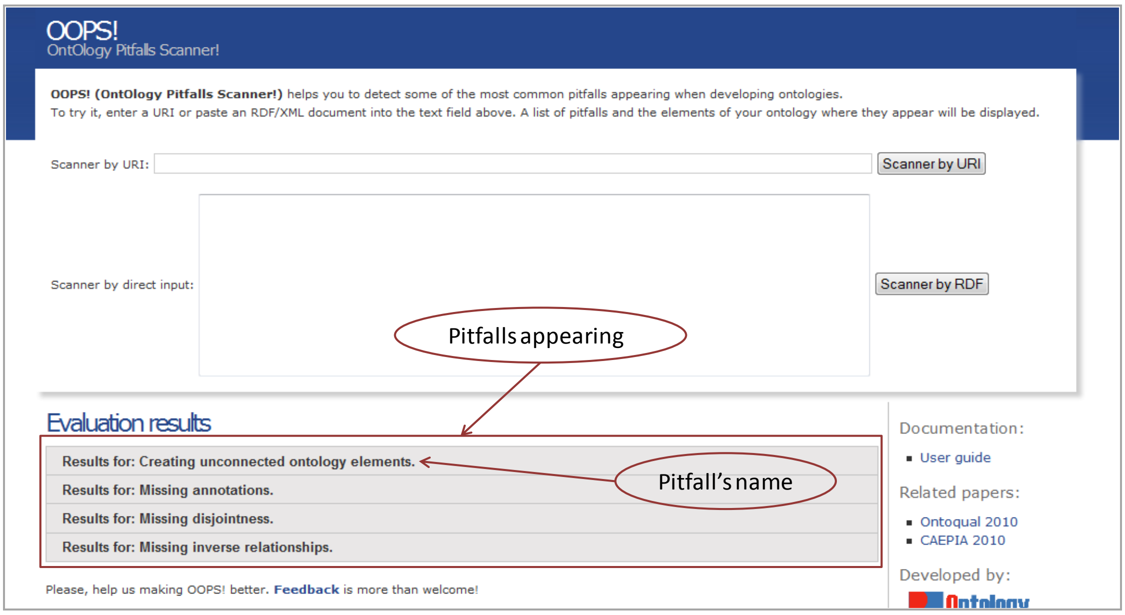

OOPS!
OntOlogy Pitfalls Scanner!

OOPS! (OntOlogy Pitfalls Scanner!) helps you to detect some of the most common pitfalls appearing when developing ontologies.
To try it, enter a URI or paste an RDF/XML document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

Figure 1. Enter the ontology to be analyzed

Showing the results

Once the ontology is analyzed the results are displayed as Figure 2 shows. The evaluation results contain a list of appearing pitfalls that will be expanded by clicking in each grey rectangle.


OOPS!
OntOlogy Pitfalls Scanner!

OOPS! (OntOlogy Pitfalls Scanner!) helps you to detect some of the most common pitfalls appearing when developing ontologies.
To try it, enter a URI or paste an RDF/XML document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

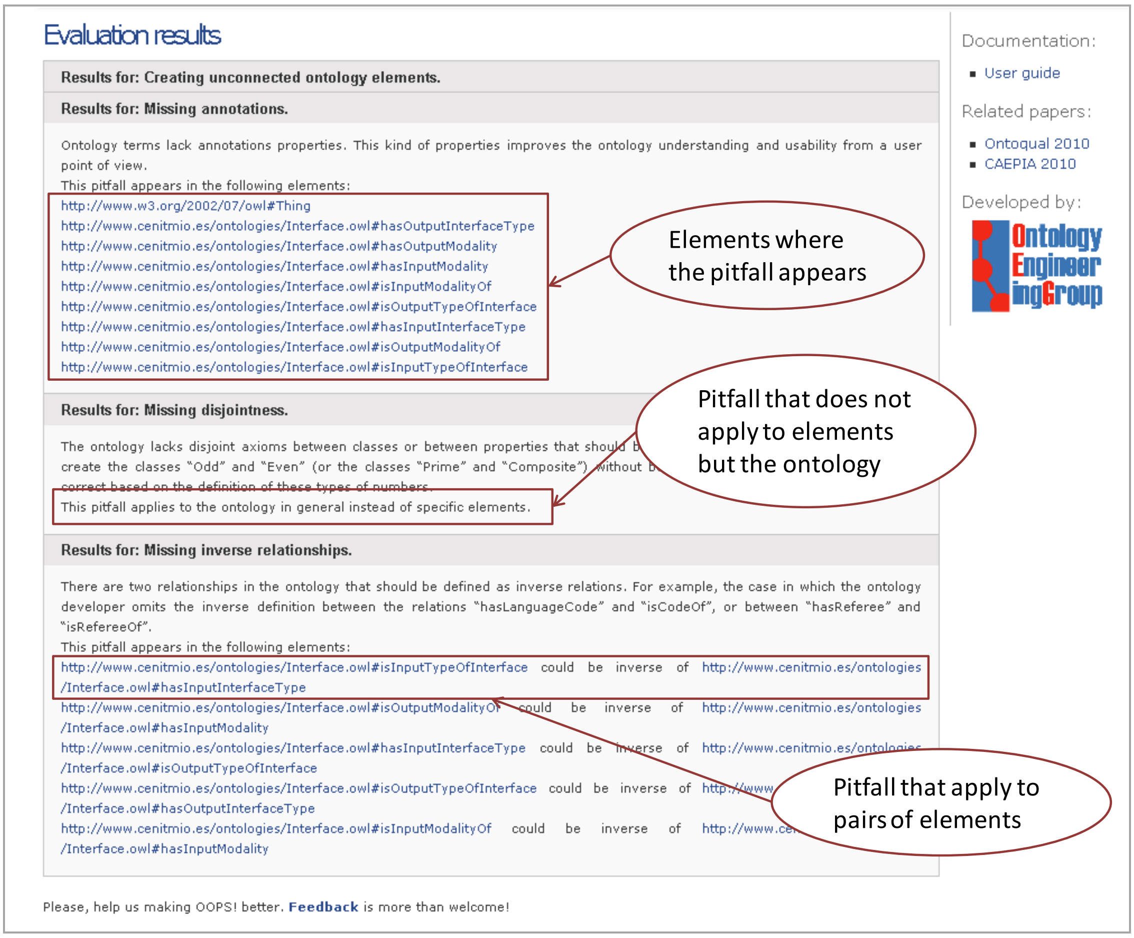
Evaluation results

Results for: Creating unconnected ontology elements.
Results for: Missing annotations.
Results for: Missing disjointness.
Results for: Missing inverse relationships.

Please, help us making OOPS! better. Feedback is more than welcome!

Figure 2. Results collapsed

When the results are expanded a brief description and example of each pitfall will be shown. Also, it will be indicated, as it is shown in Figure 3, the elements, or pair of elements, where the pitfall appears or whether the pitfall affects to the whole ontology instead of particular elements.



Evaluation results

Results for: Creating unconnected ontology elements.

Results for: Missing annotations.

Ontology terms lack annotations properties. This kind of properties improves the ontology understanding and usability from a user point of view.
This pitfall appears in the following elements:
http://www.w3.org/2002/07/owl#Thing
http://www.cenitmio.es/ontologies/Interface.owl#hasOutputInterfaceType
http://www.cenitmio.es/ontologies/Interface.owl#hasOutputModality
http://www.cenitmio.es/ontologies/Interface.owl#hasInputModality
http://www.cenitmio.es/ontologies/Interface.owl#isInputModalityOf
http://www.cenitmio.es/ontologies/Interface.owl#isOutputTypeOfInterface
http://www.cenitmio.es/ontologies/Interface.owl#hasInputInterfaceType
http://www.cenitmio.es/ontologies/Interface.owl#isOutputModalityOf
http://www.cenitmio.es/ontologies/Interface.owl#isInputTypeOfInterface

Results for: Missing disjointness.

The ontology lacks disjoint axioms between classes or between properties that should b
create the classes "Odd" and "Even" (or the classes "Prime" and "Composite") without b
correct based on the definition of these types of numbers.
This pitfall applies to the ontology in general instead of specific elements.

Results for: Missing inverse relationships.

There are two relationships in the ontology that should be defined as inverse relations. For example, the case in which the ontology developer omits the inverse definition between the relations "hasLanguageCode" and "isCodeOf", or between "hasReferee" and "isRefereeOf".
This pitfall appears in the following elements:
http://www.cenitmio.es/ontologies/Interface.owl#isInputTypeOfInterface could be inverse of http://www.cenitmio.es/ontologies/Interface.owl#hasInputInterfaceType
http://www.cenitmio.es/ontologies/Interface.owl#isOutputModalityOf could be inverse of http://www.cenitmio.es/ontologies/Interface.owl#hasInputModality
http://www.cenitmio.es/ontologies/Interface.owl#hasInputInterfaceType could be inverse of http://www.cenitmio.es/ontologies/Interface.owl#isOutputTypeOfInterface
http://www.cenitmio.es/ontologies/Interface.owl#isOutputTypeOfInterface could be inverse of http://www
/Interface.owl#hasOutputInterfaceType
http://www.cenitmio.es/ontologies/Interface.owl#isInputModalityOf could be inverse of http://www.ce
/Interface.owl#hasInputModality

Please, help us making OOPS! better. **Feedback** is more than welcome!

Documentation:
- User guide

Related papers:
- Ontoqual 2010
- CAEPIA 2010

Developed by:

Elements where the pitfall appears

Pitfall that does not apply to elements but the ontology

Pitfall that apply to pairs of elements

Figure 3. Results expanded

The result given by OOPS! suggests how the ontology elements could be modified to improve the ontology quality. However, not all of the pitfalls identified should be interpreted as factual errors but as suggestions that **must** be manually revised in some cases. This manual revision is two-fold, it is intended to find cases where a) OOPS! suggests an occurrence of a pitfall that is not an error, and b) there is an error that has not been detected by OOPS!. Concretely the following pitfalls must be checked manually;

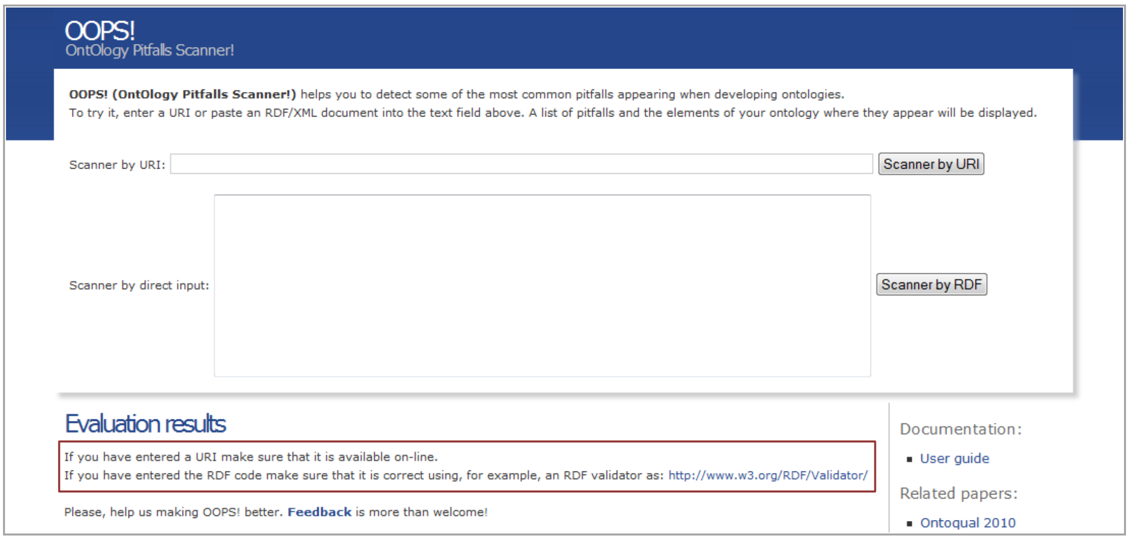
- Creating the relationship "is" instead of using ''rdfs:subClassOf'', ''rdf:type'' or ''owl:sameAs''
- Creating unconnected ontology elements
- Defining wrong inverse relationships
- Merging different concepts in the same class
- Missing equivalent properties
- Missing inverse relationships
- Swapping intersection and union
- Using a miscellaneous class
- Using recursive definition
- Defining wrong equivalent relationships

As OOPS! is not intended to be an ontology editor, those modifications within the ontology steaming from the results given by OOPS! must be carried out using an ontology editor, for example, NeOn Toolkit[1], Protégé[2] or TopBraidComposer[3].

Once, the developers have modified the ontology it is recommended to evaluate it with OOPS! again, as some modifications can help to discover errors that were hidden because of the lack of information or to generate new ones. For example, some pitfalls are detected by comparing domains and ranges in properties, if they are not defined the pitfalls cannot be identified. In this sense, to repair the pitfall "Missing domain or range in properties" allows OOPS! to find other pitfalls, for instance, "Defining symmetric relationships that do not have same domain and range".

If there has been any errors

In case the URI does not point to a correct ontology or the RDF code is not valid, the information displayed in Figure 4 will be shown. If you cannot find any problem with the URI or the RDF, please send to us an email explaining the case and we will find out what was wrong within OOPS!.



OOPS!
OntOlogy Pitfalls Scanner!

OOPS! (OntOlogy Pitfalls Scanner!) helps you to detect some of the most common pitfalls appearing when developing ontologies.
To try it, enter a URI or paste an RDF/XML document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed.

Evaluation results

If you have entered a URI make sure that it is available on-line.
If you have entered the RDF code make sure that it is correct using, for example, an RDF validator as: http://www.w3.org/RDF/Validator/

Please, help us making OOPS! better. **Feedback** is more than welcome!

Figure 4. Error in URI or RDF code

[1] http://neon-toolkit.org/wiki/Main_Page
[2] http://protege.stanford.edu/
[3] http://www.topquadrant.com/products/TB_Composer.html