Ontology Pre-Processor Language (OPPL)

http://oppl.sourceforge.net/

Mikel Egaña (mikel.eganaaranguren@cs.man.ac.uk)
Robert Stevens (robert.stevens@manchester.ac.uk)
Bio-Health Informatics Group (BHIG), University of Manchester

Erick Antezana (erant@psb.ugent.be)
Department of Plant Systems Biology, VIB, Gent, Belgium
Department of Molecular Genetics, Ghent University, Gent, Belgium

OWLED 2008 DC

Introduction to OPPL

"Macro" language for manipulating content of OWL ontologies

Motivation: authors' needs when creating/maintaining bio-ontologies (Use cases)

Nearly API-level functionality with minimum programming/OWL knowledge

Manchester OWL Syntax + **SELECT**, **ADD**, **REMOVE**

Introduction to OPPL

"Macro" language for manipulating content of OWL ontologies

Motivation: authors' needs when creating/maintaining bio-ontologies (Use cases)

Nearly API-level functionality with minimum programming/OWL knowledge

Manchester OWL Syntax + **SELECT**, **ADD**, **REMOVE**

Introduction to OPPL

"Macro" language for manipulating content of OWL ontologies

Motivation: authors' needs when creating/maintaining bio-ontologies (Use cases)

Nearly API-level functionality with minimum programming/OWL knowledge

Manchester OWL Syntax + **SELECT**, **ADD**, **REMOVE**

Introduction to OPPL

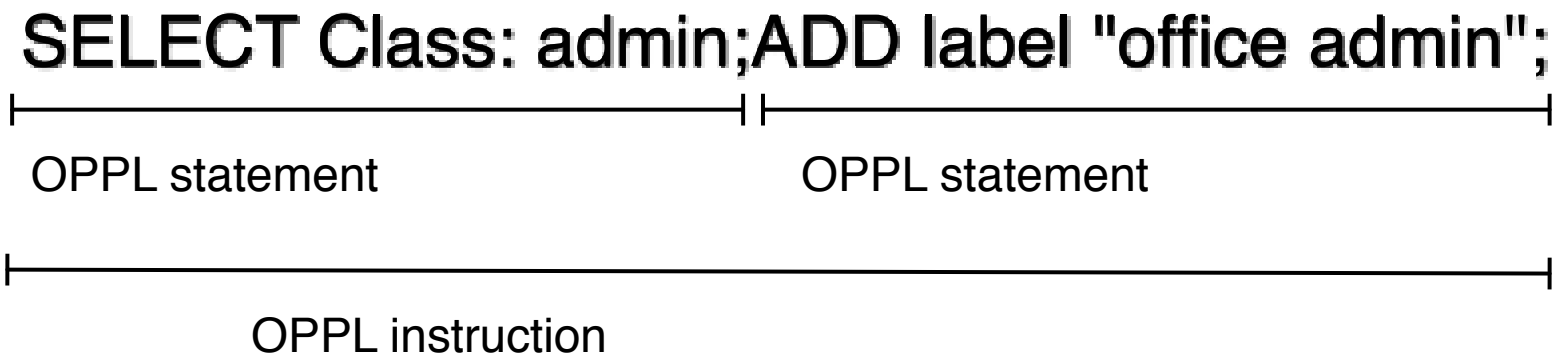
"Macro" language for manipulating content of OWL ontologies

Motivation: authors' needs when creating/maintaining bio-ontologies (Use cases)

Nearly API-level functionality with minimum programming/OWL knowledge

Manchester OWL Syntax + **`SELECT`**, **`ADD`**, **`REMOVE`**

OPPL instruction



OPPL instruction examples

```
ADD Class:  undergraduate;
```

```
REMOVE Class:  undergraduate;
```

```
SELECT equivalentTo participates_in only
(intellectual_dinner and party);
ADD label "professor";
REMOVE subClassOf lives_on only (not campus);
```

OPPL instruction examples

```
ADD Class:  undergraduate;
```

```
REMOVE Class:  undergraduate;
```

```
SELECT equivalentTo participates_in only
(intellectual_dinner and party);
ADD label "professor";
REMOVE subClassOf lives_on only (not campus);
```

OPPL instruction examples

```
ADD Class:  undergraduate;
```

```
REMOVE Class:  undergraduate;
```

```
SELECT equivalentTo participates_in only
(intellectual_dinner and party);
ADD label "professor";
REMOVE subClassOf lives_on only (not campus);
```

OPPL instruction examples

```
SELECT subClassOf MoleOfSalt;
ADD subClassOf hasDensity value 0.0;
```

```
SELECT inverse participates_in;
ADD range student;
```

```
ADD Class:  professor;
ADD label "staff";
ADD equivalentTo participates_in only
(intellectual_dinner and party);
```

OPPL instruction examples

```
SELECT subClassOf MoleOfSalt;
ADD subClassOf hasDensity value 0.0;
```

```
SELECT inverse participates_in;
ADD range student;
```

```
ADD Class:  professor;
ADD label "staff";
ADD equivalentTo participates_in only
(intellectual_dinner and party);
```

OPPL instruction examples

```
SELECT subClassOf MoleOfSalt;
ADD subClassOf hasDensity value 0.0;
```

```
SELECT inverse participates_in;
ADD range student;
```

```
ADD Class:  professor;
ADD label "staff";
ADD equivalentTo participates_in only
(intellectual_dinner and party);
```

OPPL instruction examples

```
disjointWith, differentFrom, sameAs, type,
descendantOf, ancestorOf, subPropertyOf, ...
```

```
http://oppl.sourceforge.net/test.oppl
```

OPPL instruction examples

```
disjointWith, differentFrom, sameAs, type,
descendantOf, ancestorOf, subPropertyOf, ...
```

```
http://oppl.sourceforge.net/test.oppl
```

OPPL extra instructions

```
SELECT_PRIMITIVE descendantOf student;

SELECT_DEFINED descendantOf student;
```

```
SELECT descendantOf person;
ADD disjointWithSiblings;
```

```
SELECT assertedSubClassOf participates_in some
sport;
```

OPPL extra instructions

```
SELECT_PRIMITIVE descendantOf student;

SELECT_DEFINED descendantOf student;
```

```
SELECT descendantOf person;
ADD disjointWithSiblings;
```

```
SELECT assertedSubClassOf participates_in some
sport;
```

OPPL extra instructions

```
SELECT_PRIMITIVE descendantOf student;

SELECT_DEFINED descendantOf student;
```

```
SELECT descendantOf person;
ADD disjointWithSiblings;
```

```
SELECT assertedSubClassOf participates_in some
sport;
```

OPPL software

`http://oppl.sourceforge.net/`

`OPPLInstructionManager`
Java library for processing OPPL instructions (LGPL)
OWL API, Pellet, FaCT++, DIG

OPPL reference implementation: OPPL instructions in flat file

```
java -jar oppl.jar
pellet ins.oppl ont.owl new_ont.owl
```

OPPL software

`http://oppl.sourceforge.net/`

`OPPLInstructionManager`
Java library for processing OPPL instructions (LGPL)
OWL API, Pellet, FaCT++, DIG

OPPL reference implementation: OPPL instructions in flat file

```
java -jar oppl.jar
pellet ins.oppl ont.owl new_ont.owl
```

OPPL software

`http://oppl.sourceforge.net/`

`OPPLInstructionManager`
Java library for processing OPPL instructions (LGPL)
OWL API, Pellet, FaCT++, DIG

OPPL reference implementation: OPPL instructions in flat file

```
java -jar oppl.jar
pellet ins.oppl ont.owl new_ont.owl
```

Use case 1: Bio-ontology axiomatic enrichment

Gene Ontology Next Generation

`http://www.gong.manchester.ac.uk/`

```
alanine:sodium symporter activity

—— Axiomatic enrichment based in rdfs:label ——

EquivalentTo:  symporter activity
and transports only (alanine or sodium)
```

Use case 1: Bio-ontology axiomatic enrichment

Gene Ontology Next Generation
`http://www.gong.manchester.ac.uk/`

```
alanine:sodium symporter activity

—— Axiomatic enrichment based in rdfs:label ——

EquivalentTo:  symporter activity
and transports only (alanine or sodium)
```

Use case 1: Bio-ontology axiomatic enrichment

rdfs:label: `alanine:sodium symporter activity`
rdfs:label: `glycine:potassium symporter activity`
rdfs:label: `valine:sodium symporter activity`
...

```
SELECT label "(.+):(.+) (symporter activity)";
ADD equivalentTo symporter_activity and
transports only (<1> or <2>);
```

Use case 2: Cell Cycle Ontology

`http://www.cellcycleontology.org/`

Gather knowledge about the cell cycle in 5 ontologies

Ontologies created anew each pipeline execution

Impossible to add new axioms by hand to 5 ontologies:
- Axioms overwritten each time the pipeline is executed
- Many axioms in different places
- Ontologies too big

Use case 2: Cell Cycle Ontology

`http://www.cellcycleontology.org/`

Gather knowledge about the cell cycle in 5 ontologies

Ontologies created anew each pipeline execution

Impossible to add new axioms by hand to 5 ontologies:
- Axioms overwritten each time the pipeline is executed
- Many axioms in different places
- Ontologies too big

Use case 2: Cell Cycle Ontology

Add axioms automatically each time the pipeline is executed

Explicit development

Querying capabilities

Flexible development

Ontology Design Patterns
http://odps.sourceforge.net

Use case 2: Cell Cycle Ontology

Add axioms automatically each time the pipeline is executed

Explicit development

Querying capabilities

Flexible development

Ontology Design Patterns
http://odps.sourceforge.net

Use case 2: Cell Cycle Ontology

Add axioms automatically each time the pipeline is executed

Explicit development

Querying capabilities

Flexible development

Ontology Design Patterns
http://odps.sourceforge.net

Use case 2: Cell Cycle Ontology

Add axioms automatically each time the pipeline is executed

Explicit development

Querying capabilities

Flexible development

Ontology Design Patterns
http://odps.sourceforge.net

Use case 2: Cell Cycle Ontology

Add axioms automatically each time the pipeline is executed

Explicit development

Querying capabilities

Flexible development

Ontology Design Patterns
`http://odps.sourceforge.net`

Use case 2: Cell Cycle Ontology

```
# Create object property inmediately_precedes

ADD ObjectProperty: inmediately_precedes;ADD functional;
ADD subPropertyOf precedes;ADD inverse inmediately_preceded_by;ADD domain
CCO_U0000002;ADD range CCO_U0000002;

# Meiotic cell cycle: G1 -> S -> G2 -> M

SELECT Class: CCO_P0000327;ADD subClassOf inmediately_preceded_by some
CCO_P0000325;ADD subClassOf inmediately_precedes some CCO_P0000326;

# Query 1: Proteins acting in the mitotic S phase (At)

ADD Class: query_1;ADD subClassOf query;REMOVE subClassOf Thing;
ADD comment "Proteins acting in the mitotic S phase";

SELECT subClassOf participates_in some (CCO_P0000014 or (part_of some
CCO_P0000014));ADD subClassOf query_1;
```

Future of OPPL

Syntax closer to OWL?

```
ADD Class:  professor
ADD professor subClassOf Thing
```

Loops, conditional control, subroutines, ...

Variables?

```
SELECT subClassOf part_of some ?x;
ADD equivalentTo part_of only ?x;
```

```
SET professor label "new label";
```

Future of OPPL

Syntax closer to OWL?

```
ADD Class:  professor
ADD professor subClassOf Thing
```

Loops, conditional control, subroutines, ...

Variables?

```
SELECT subClassOf part_of some ?x;
ADD equivalentTo part_of only ?x;
```

```
SET professor label "new label";
```

Future of OPPL

Syntax closer to OWL?

```
ADD Class:  professor
ADD professor subClassOf Thing
```

Loops, conditional control, subroutines, ...

Variables?

```
SELECT subClassOf part_of some ?x;
ADD equivalentTo part_of only ?x;
```

```
SET professor label "new label";
```

Future of OPPL

Syntax closer to OWL?

```
ADD Class:  professor
ADD professor subClassOf Thing
```

Loops, conditional control, subroutines, ...

Variables?

```
SELECT subClassOf part_of some ?x;
ADD equivalentTo part_of only ?x;
```

```
SET professor label "new label";
```

Future of OPPL

- Formal grammar
- Protégé plugin (autocomplete, syntax validation, logs, ...)

Future of OPPL

- Formal grammar
- Protégé plugin (autocomplete, syntax validation, logs, ...)

Acknowledgements

Mikel Egaña: University of Manchester and EPSRC

Erick Antezana: European Science Foundation (ESF) for the activity entitled Frontiers of Functional Genomics

`http://oppl.sourceforge.net/`