An API for ontology alignment

Jérôme Euzenat

Jerome.Euzenat@inrialpes.fr

Who did it

Basically done at INRIA with some contributions from University of Montreal.

Related to Knowledge web.

Based on Java, OWL-API (for the implementation).

Conceptual work

The API relies on the work done in Knowledge web

[Alignment algebra to be based on categorical work].

The implementations can be based on other work (we have an aligner, OLA, based on our work with UoM).

Functionality and limitation

Reads, generates, manipulates alignments;

Limitation of the API:

- Does not provides you with the alignments;

Limitation of the implementation:

- Currently limited to level 0;
- Currently limited to OWL-API.

Architecture

- An API around objects;
- Fully embeddable;
- Can be extended and customized;

Outline

1 Format
2 API
3 Examples of the API use
4 Availability

What's in an alignment (container)

- Level indicator (0, 1, 2+);
- Description of the alignment: 1:1, 1:*, etc.;
- Pair of ontology URIs;
- Set of pairs;

What else?

generating algorithm, formal properties of the alignment, any kind of annotation…

Format

- Format in RDF, in fact RDF/XML;
- Defined in DTD, OWL and RDFS;
- Entities are identified by URIs.

RDF/XML Syntax

```
<Alignment>
  <xml>yes</xml>
  <level>0</level>
  <type>**</type>
  <uri1>http://ebiquity.umbc.edu/v2.1/ontology/publication.owl#
       publication</uri1>
   <uri2>file://localhost/Volumes/Phata/JAVA/ontoalign/rdf/edu.mit
       .visus.bibtex.owl</uri2>
  <map>
    <Cell>
      <entity1
   rdf:resource='http://ebiquity.umbc.edu/v2.1/ontology/publication.o
   wl#volume'/>
      <entity2
   rdf:resource='file://localhost/Volumes/Phata/JAVA/ontoalign/rdf/ed
   u.mit.visus.bibtex.owl#hasVolume'/>
      <measure
   rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.8</measure
   >
      <relation>=</relation>
    </Cell>
```


…

What is an alignment for?

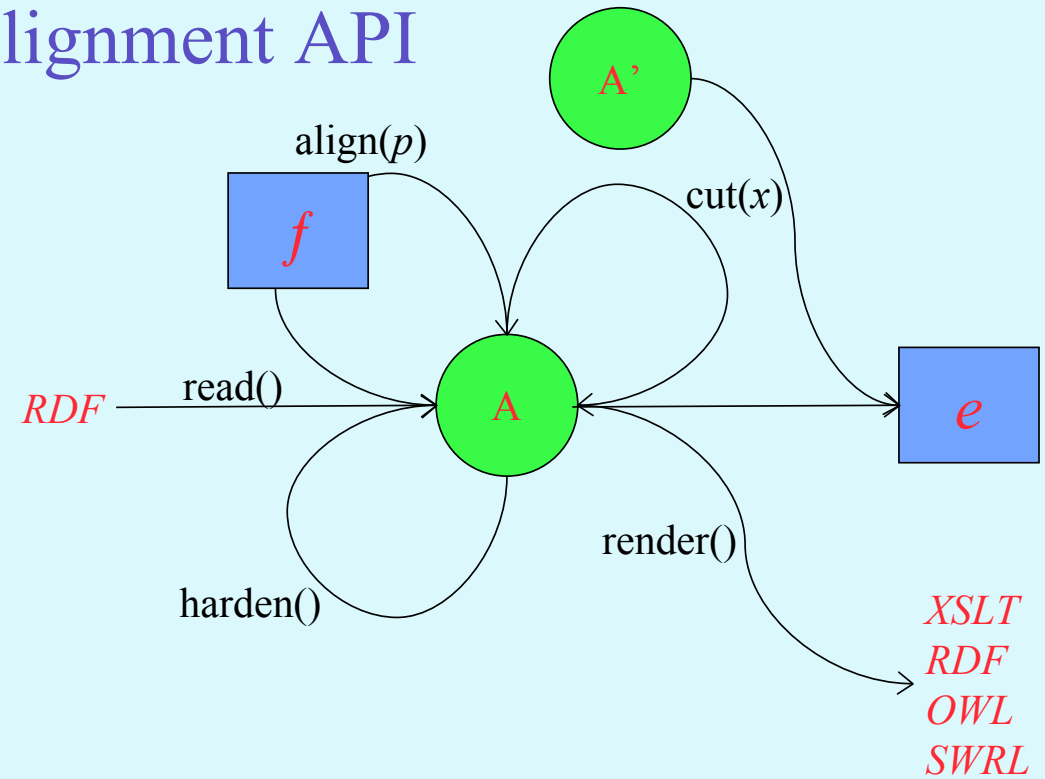
- Storing, finding, and floating around;
- Piping alignments algorithms (improving an existing alignment);
- Manipulating (thresholding and hardening);
- Generating processing output (transformations, axioms, rules);
- Comparing alignments.

API

Set of Java interfaces:

- Alignment
  align(A,p), *cut*(t), *harden*(), *render*(s,v)
- Cell;
- Relation;
- Evaluator
  eval(p), *write*()

Alignment API



API implementation

- Grounded on the OWL-API + [0 1] measures;
- Base implementation of the interfaces with all useful facilities;
- Library of sample aligners;
- Library of renderers (XSLT, SWRL, OWL, C-OWL…);
- Couple of evaluators (P/R);
- Parser.

Examples of the API use

Demonstration

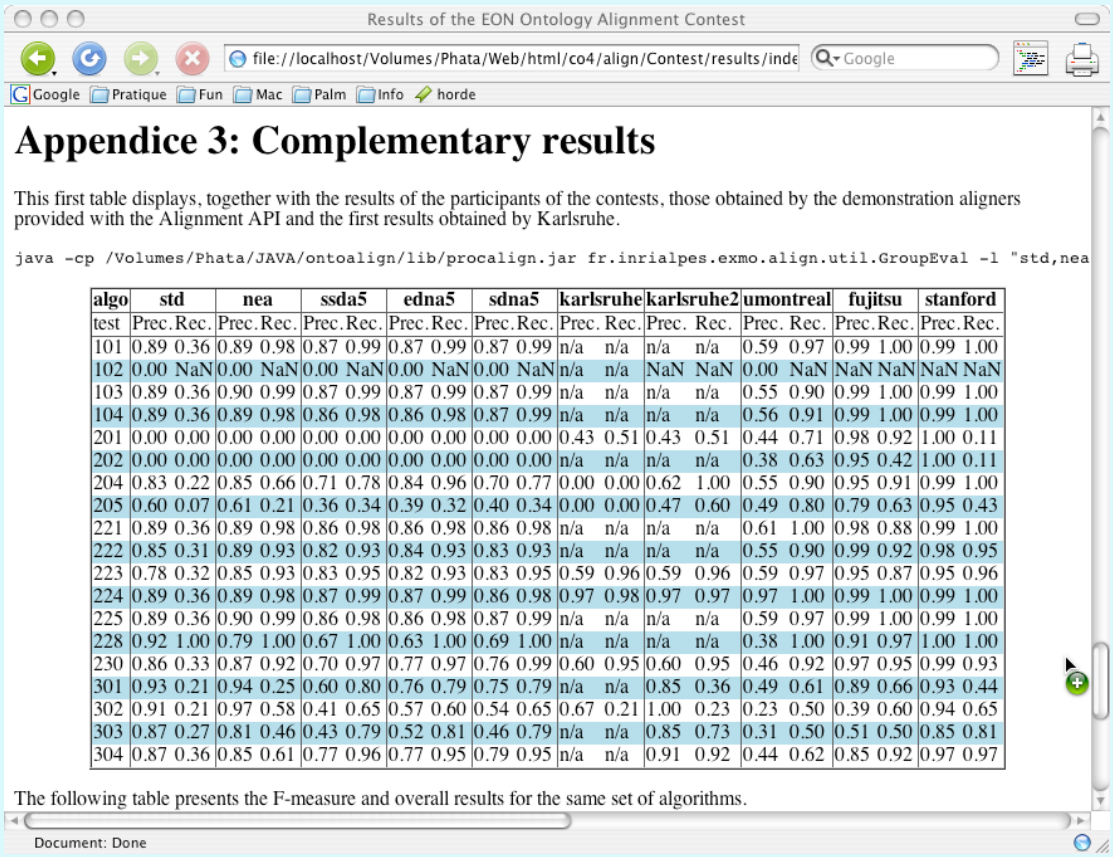
- Use as command line;
- Show how to embed (skeleton example);
- Show how to extend (create new object);
- Show Anna's JSP.

Example

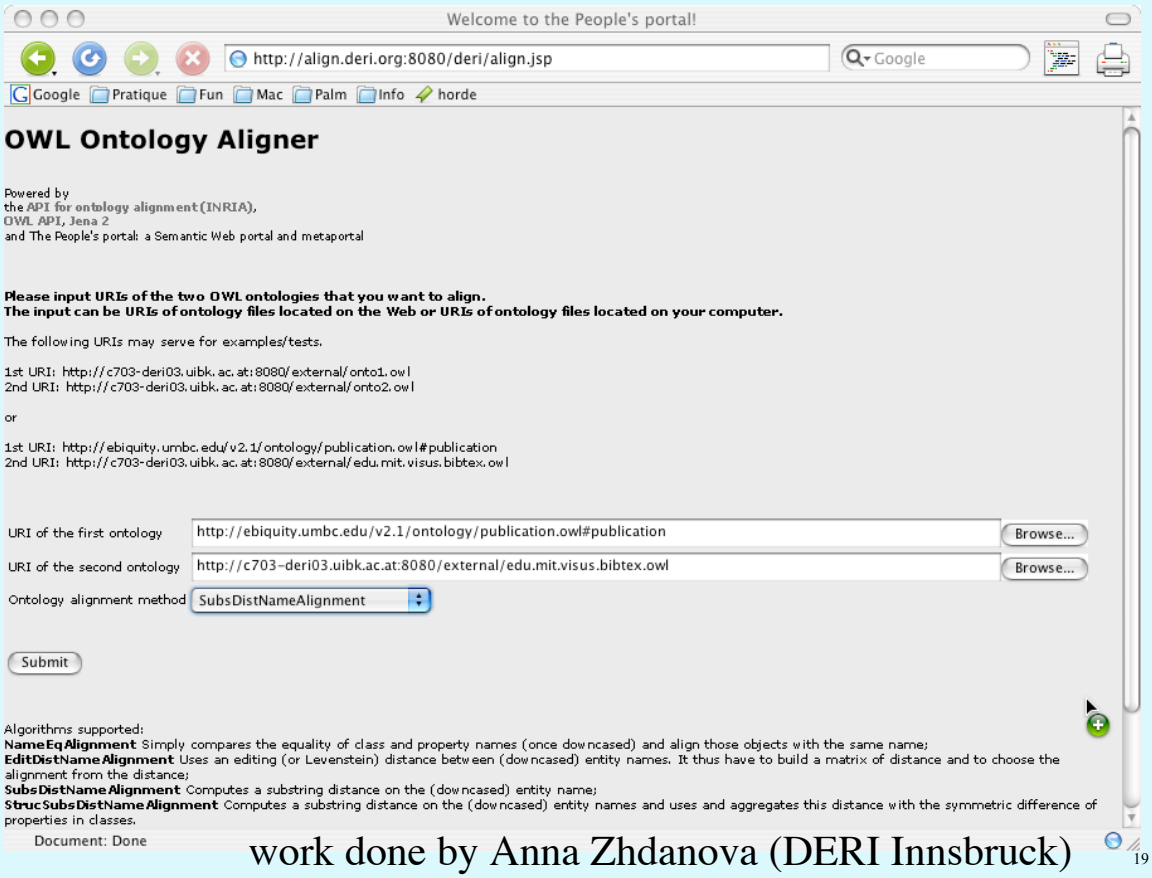
I have two ontologies
I align the property names + the properties
    see what I got to do in the API
I align classes and the class names
    see what I got to do in the API or the command line
I threshold
    see what I got to do in the API or the command line
I compare with something
    again…
I render in some format
    show it, proove it!
See the result of the contest actually

Example of API use

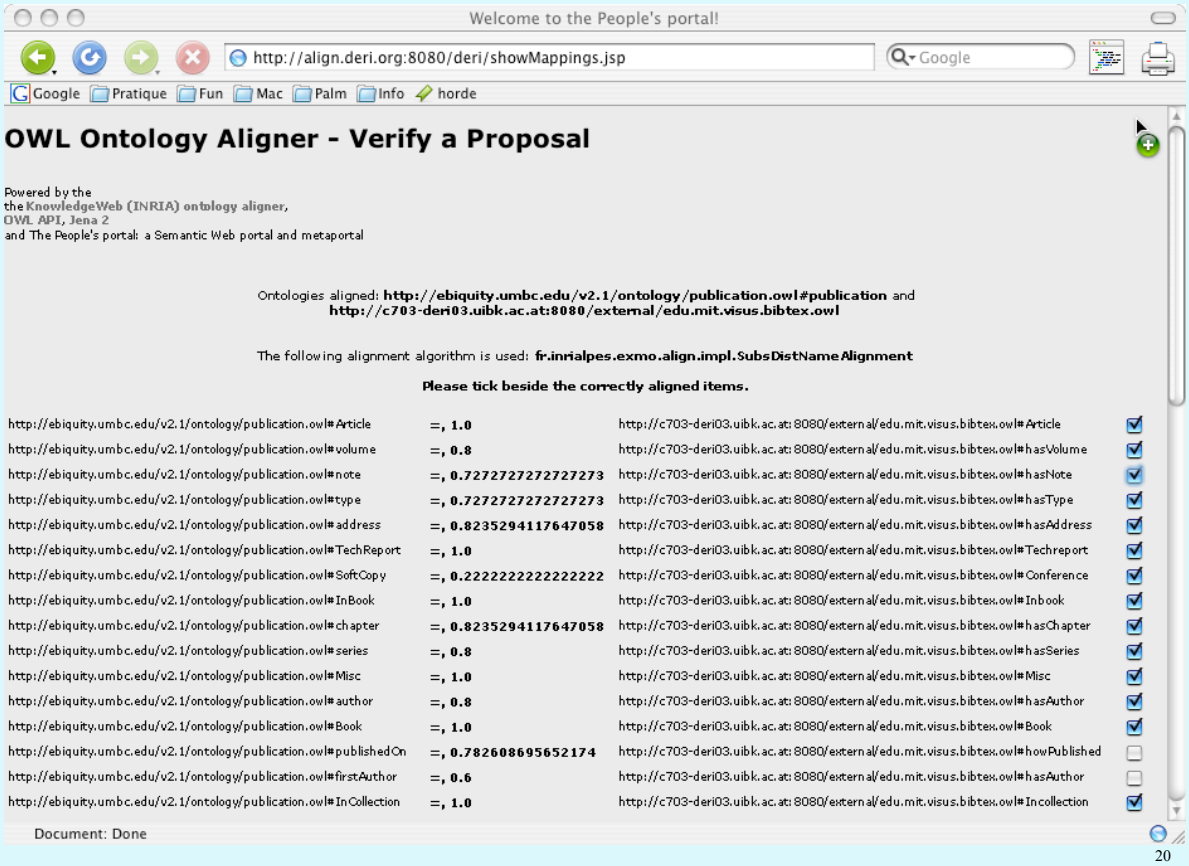
```
OWLOntology O1 = loadOntology(…);
OWLOntology O2 = loadOntology(…);
Alignment A1 =
  new SubsDistNameAlignment(O1, O2);
Alignment A2 =
  new PropSubsDistAlignment(O1,O2);
Alignment A3 =
  new NameAndPropertyAlignment(O1,O2);
A1.align(); A1.cut("prop",.5);
A2.align(); A3.align(A2);
Evaluator E = new PRecEvaluator(A1, A3);
E.eval(A1,A3);
if ( E.getPrecision() > .6)
  A3.render(…,SWRLRendererVisitor);
```

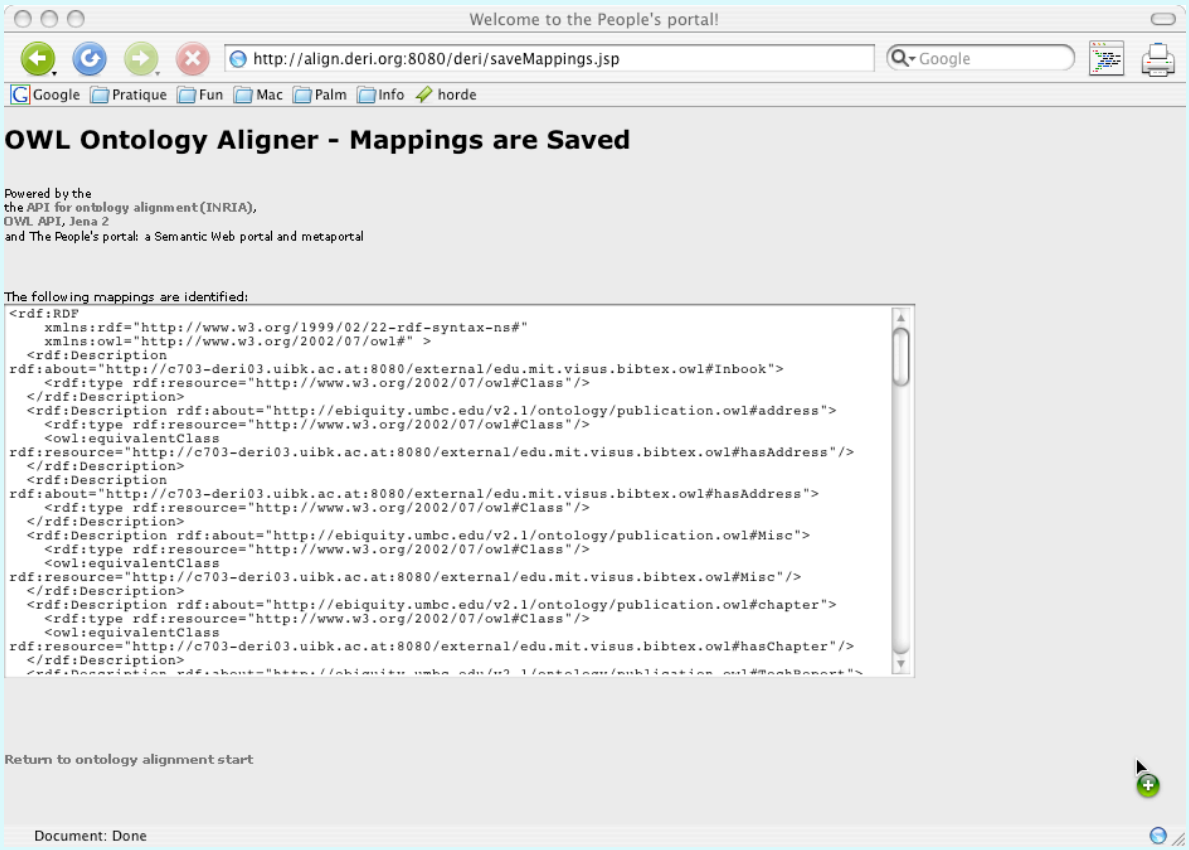



algo	std		nea		ssda5		edna5		sdna5		karlsruhe		karlsruhe2		umontreal		fujitsu		stanford	
test	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
101	0.89	0.36	0.89	0.98	0.87	0.99	0.87	0.99	0.87	0.99	n/a	n/a	n/a	n/a	0.59	0.97	0.99	1.00	0.99	1.00
102	0.00	NaN	0.00	NaN	0.00	NaN	0.00	NaN	0.00	NaN	n/a	n/a	NaN	NaN	0.00	NaN	NaN	NaN	NaN	NaN
103	0.89	0.36	0.90	0.99	0.87	0.99	0.87	0.99	0.87	0.99	n/a	n/a	n/a	n/a	0.55	0.90	0.99	1.00	0.99	1.00
104	0.89	0.36	0.89	0.98	0.86	0.98	0.86	0.98	0.87	0.99	n/a	n/a	n/a	n/a	0.56	0.91	0.99	1.00	0.99	1.00
201	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.43	0.51	0.43	0.51	0.44	0.71	0.98	0.92	1.00	0.11
202	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	n/a	n/a	n/a	n/a	0.38	0.63	0.95	0.42	1.00	0.11
204	0.83	0.22	0.85	0.66	0.71	0.78	0.84	0.96	0.70	0.77	0.00	0.00	0.62	1.00	0.55	0.90	0.95	0.91	0.99	1.00
205	0.60	0.07	0.61	0.21	0.36	0.34	0.39	0.32	0.40	0.34	0.00	0.00	0.47	0.60	0.49	0.80	0.79	0.63	0.95	0.43
221	0.89	0.36	0.89	0.98	0.86	0.98	0.86	0.98	0.86	0.98	n/a	n/a	n/a	n/a	0.61	1.00	0.98	0.88	0.99	1.00
222	0.85	0.31	0.89	0.93	0.82	0.93	0.84	0.93	0.83	0.93	n/a	n/a	n/a	n/a	0.55	0.90	0.99	0.92	0.98	0.95
223	0.78	0.32	0.85	0.93	0.83	0.95	0.82	0.93	0.83	0.95	0.59	0.96	0.59	0.96	0.59	0.97	0.95	0.87	0.95	0.96
224	0.89	0.36	0.89	0.98	0.87	0.99	0.87	0.99	0.86	0.98	0.97	0.98	0.97	0.97	0.97	1.00	0.99	1.00	0.99	1.00
225	0.89	0.36	0.90	0.99	0.86	0.98	0.86	0.98	0.87	0.99	n/a	n/a	n/a	n/a	0.59	0.97	0.99	1.00	0.99	1.00
228	0.92	1.00	0.79	1.00	0.67	1.00	0.63	1.00	0.69	1.00	n/a	n/a	n/a	n/a	0.38	1.00	0.91	0.97	1.00	1.00
230	0.86	0.33	0.87	0.92	0.70	0.97	0.77	0.97	0.76	0.99	0.60	0.95	0.60	0.95	0.46	0.92	0.97	0.95	0.99	0.93
301	0.93	0.21	0.94	0.25	0.60	0.80	0.76	0.79	0.75	0.79	n/a	n/a	0.85	0.36	0.49	0.61	0.89	0.66	0.93	0.44
302	0.91	0.21	0.97	0.58	0.41	0.65	0.57	0.60	0.54	0.65	0.67	0.21	1.00	0.23	0.23	0.50	0.39	0.60	0.94	0.65
303	0.87	0.27	0.81	0.46	0.43	0.79	0.52	0.81	0.46	0.79	n/a	n/a	0.85	0.73	0.31	0.50	0.51	0.50	0.85	0.81
304	0.87	0.36	0.85	0.61	0.77	0.96	0.77	0.95	0.79	0.95	n/a	n/a	0.91	0.92	0.44	0.62	0.85	0.92	0.97	0.97



work done by Anna Zhdanova (DERI Innsbruck)





Summary

We have:

- Proposed a general format for expressing alignments among ontologies;
- Produced an API and an implementation;
- Provide a number of services and example.

Focusing on usability and reusability.

Current state of the alignment API

- API as a set of Java interfaces (Alignment, Cell, Relation, Evaluator…);
- Unique rendering format in RDF/XML described by DTD, RDF(S) and OWL ontology;
- Implementation with the OWL API (available with many examples);
- Used in the EON contest.

Current status

- Available under LGPL;
- Stable, still under development (implementation);

  http://co4.inrialpes.fr/align
- CVS archive open for read-only, write access possible: ask me;
- Few documentation, no tests currently;
- Used in the EON contest and several projects.

Future plan

Improve this! This is by no way a definitive solution…

Build a complete alignment evaluation tool on top of this API.

We need to ear from you what is wrong, what is needed, how it is useful.

http://co4.inrialpes.fr/align

?

http://www.inrialpes.fr/exmo
Jerome.Euzenat@inrialpes.fr

Definition 29 (Triangular norm). *A triangular norm T is a function from $D \times D \to D$ (with D a set ordered by $\leq$ and provided with an upper bound $\top$) satifying:*

$$
\begin{aligned}
T(x, \top) &= x && (boundary\ condition)\\
x \leq y \implies T(x, z) &\leq T(y, z) && (monotonicity)\\
T(x, y) &= T(y, x) && (commutativity)\\
T(x, T(y, z)) &= T(T(x, y), z) && (associativity)
\end{aligned}
$$