MOLTO

A company of Sirma Group
www.sirma.com

ontotext

WP4. Grammar-Ontology Interoperability

ontotext

Milen Chechev

2nd MOLTO Review
Barcelona, 20.03.2012

The goal of WP4

The objectives of WP4 are

- Research and development of two-way grammar-ontology interoperability bridging the gap between natural language and formal knowledge;
- Infrastructure for knowledge modeling, semantic indexing and retrieval;
- Modelling and alignment of structured data sources;
- Alignment of ontologies with the grammar derived models.

Current State – WP4

- Already completed
  - D4.1. Knowledge Representation Infrastructure
  - D4.2. Data Models and Alignment
  - D4.3. Grammar-Ontology Interoperability
- KRI prototype – http://molto.ontotext.com

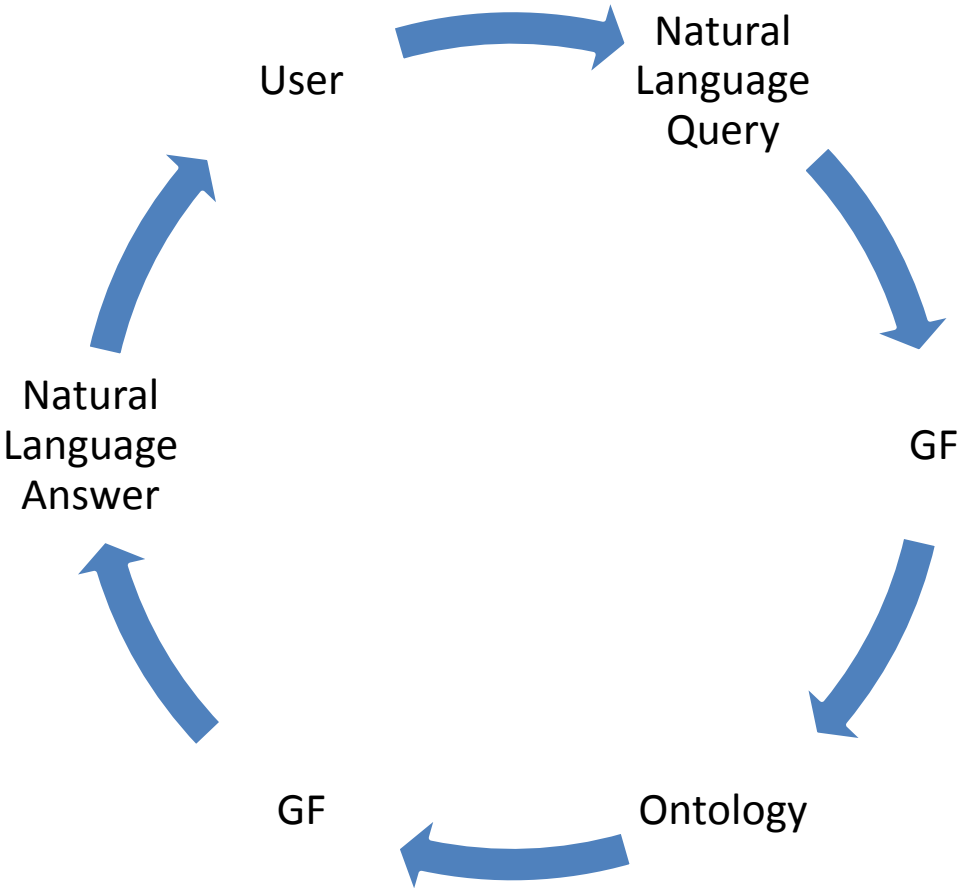
D4.1 – Knowledge Representation Infrastructure

- OWLIM - a semantic repository that stores all structured data such as ontologies, background knowledge, etc., and provides SPARQL query mechanism and reasoning;
- RDFDB - an API that provides a remote access to the stored structured data via JMS;
- PROTON Ontology - a light-weight upper-level ontology, which defines about 300 classes and 100 properties, covering most of the upper-level concepts, necessary for semantic annotation, indexing and retrieval;
- KRI Web UI a UI that accesses OWLIM through the RDFDB layer. The web UI gives the user the possibility to browse the ontologies and the database, to execute SPARQL queries, etc.

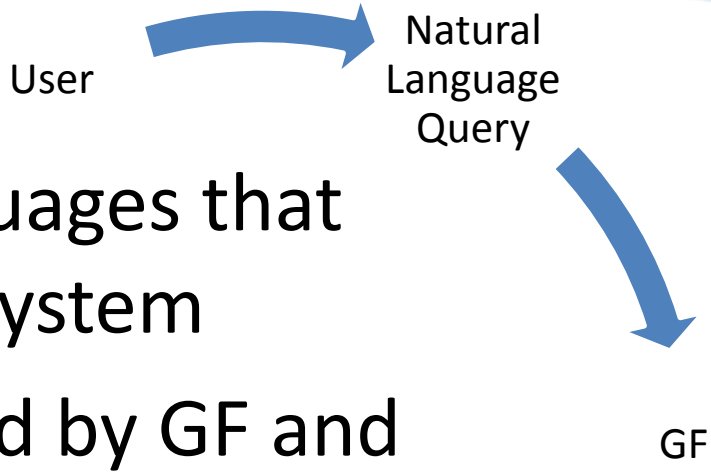
D4.2 – Data Models and Alignment

- Linked Open Data
- Reason-able view
  - DBpedia - the RDF-ized version of Wikipedia, describing more than 3.5 million things.
  - Geonames - a geographic database that covers 6 million of the most significant geographical features on Earth.
  - PROTON - an upper-level ontology, 542 entity classes and 183 properties.
  - WKB – general information about People, Locations and Organizations.
  - WordNet

D4.3 Grammar-Ontology Interoperability



Steps(1)



- User write a query at some of the natural languages that are supported from the system
- The NL query is processed by GF and transformed to GF Abstract Representation

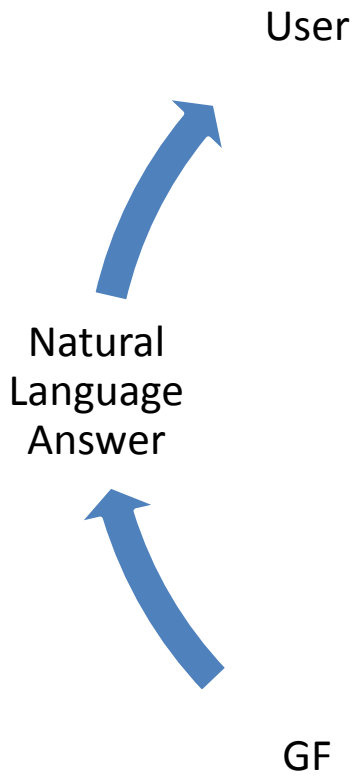
Steps(2)

- Map GF abstract representation to the SPARQL query
- Use the SPARQL query in the semantic repository and retrieve the results as RDF tripples

Steps(3)

- Process the RDF tripple results and transform them to GF Abstract Representation.

Steps(4)



- Use GF to process the GF Abstract Representation and obtain NL results

Challenges

- How to build the GF grammar that process the NL queries?
- How to map the GF abstract representation to SPARQL query?
- How to build from the retrieved triples GF abstract representation.
- How to build GF grammar for the results?

GF Query grammar

- Steps:
  - Select the knowledge domain
  - Select the possible queries
  - Make GF abstract grammar
  - Make GF concrete grammar for each supported language.

GF – SPARQL

- Semi-automated approach
- Use set of rules that transform the GF abstract representation to SPARQL query

RDF Triples - GF

- Semi-automated approach
- Use set of rules to build a GF abstract representation from the RDF tripples
- Use predefined order for the predicates.

GF Answer grammar

- Automatically build from the ontology.
- Manually corrected and refined.

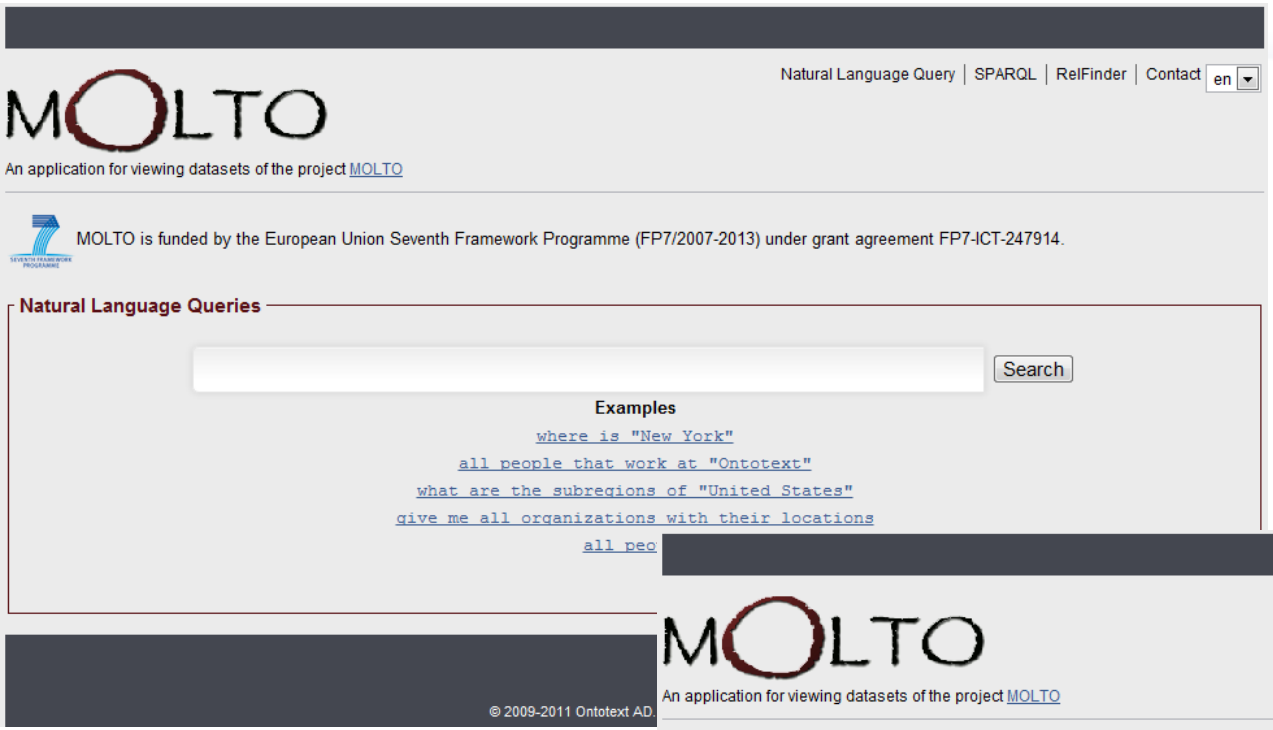
Example:GF Answer grammar

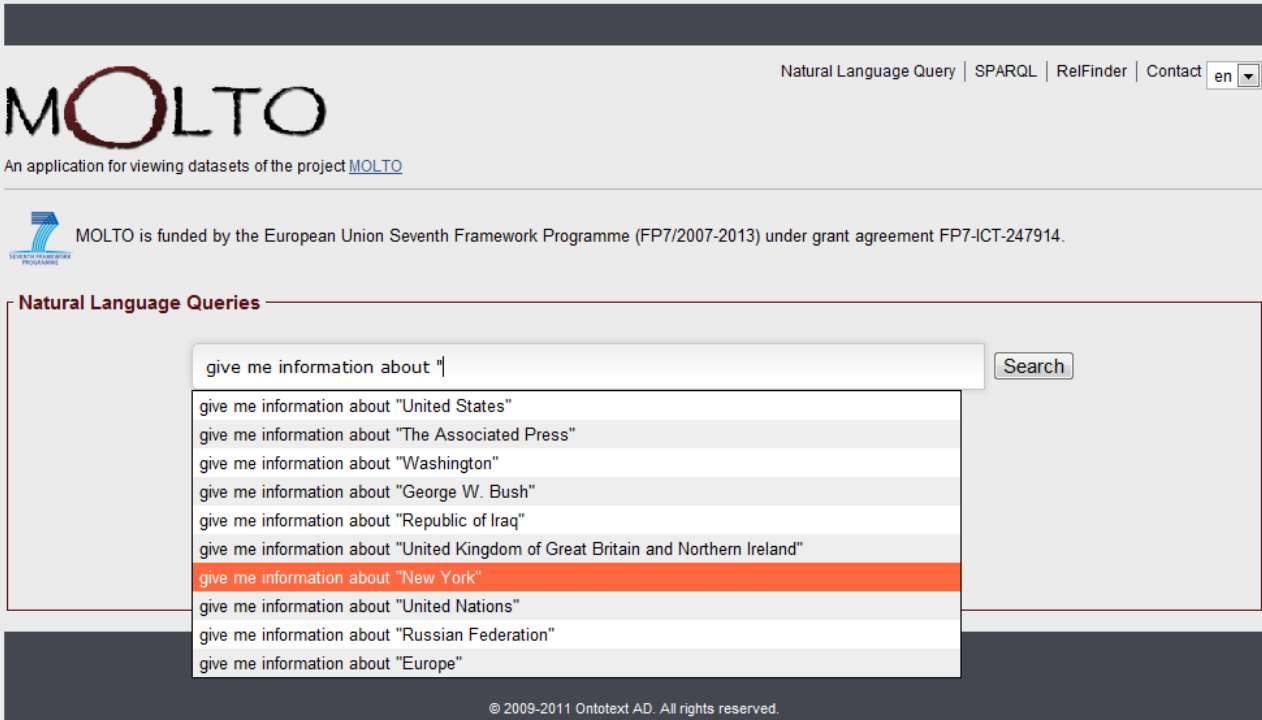
```
abstract Wkbx = {
flags startcat = Phrase;
cat
 Phrase; Bank; Continent; City; University;
Fun
InfoBank : Bank ->Phrase;
InfoContinent : Continent ->Phrase;
InfoCity : City ->Phrase;
InfoUniversity : University ->Phrase;
Bank_T_147 : Bank;
Bank_T_148 : Bank;
Continent_T_1 : Continent;
Continent_T_2 : Continent;
City_T_1 : City;
University_T_1 : University;
locatedInBankCity : Bank  -> City -> Phrase ;
locatedInUniversityCity : University  -> City -> Phrase ;
}
```

Example:GF Answer grammar(2)

```
concrete WkbEng of Wkb =
open MorphoEng, ResEng, ParadigmsEng, MakeStructuralEng, SyntaxEng in {
lincat Phrase = Cl;
 Bank = NP;
 Continent= NP;
 City= NP;
 University = NP;
lin Bank_T_1 = mkNP( mkN "Bank DSK");
 Bank_T_2 = mkNP( mkN "First International Bank");
 Continent_T_1 = mkNP( mkN "Europe");
 Continent_T_2 = mkNP( mkN "Asia");
 City_T_1 = mkNP( mkN "Sofia");
 University_T_1 = mkNP( mkN "MIT");
InfoBank x = mkCl x (mkN "bank");
InfoCity x = mkCl x (mkN "city");
InfoContinent x = mkCl x (mkN "continent");
InfoUniversity x = mkCl x (mkN "university");
locatedInBankCity x y = mkCl x  (mkVP (passiveVP (mkV2  (mkV "locate") )) (mkAdv (mkPrep "in") y));
locatedInUniversityCity x y = mkCl x  (mkVP (passiveVP (mkV2  (mkV "locate") )) (mkAdv (mkPrep "in") y));
```

http://molto.ontotext.com



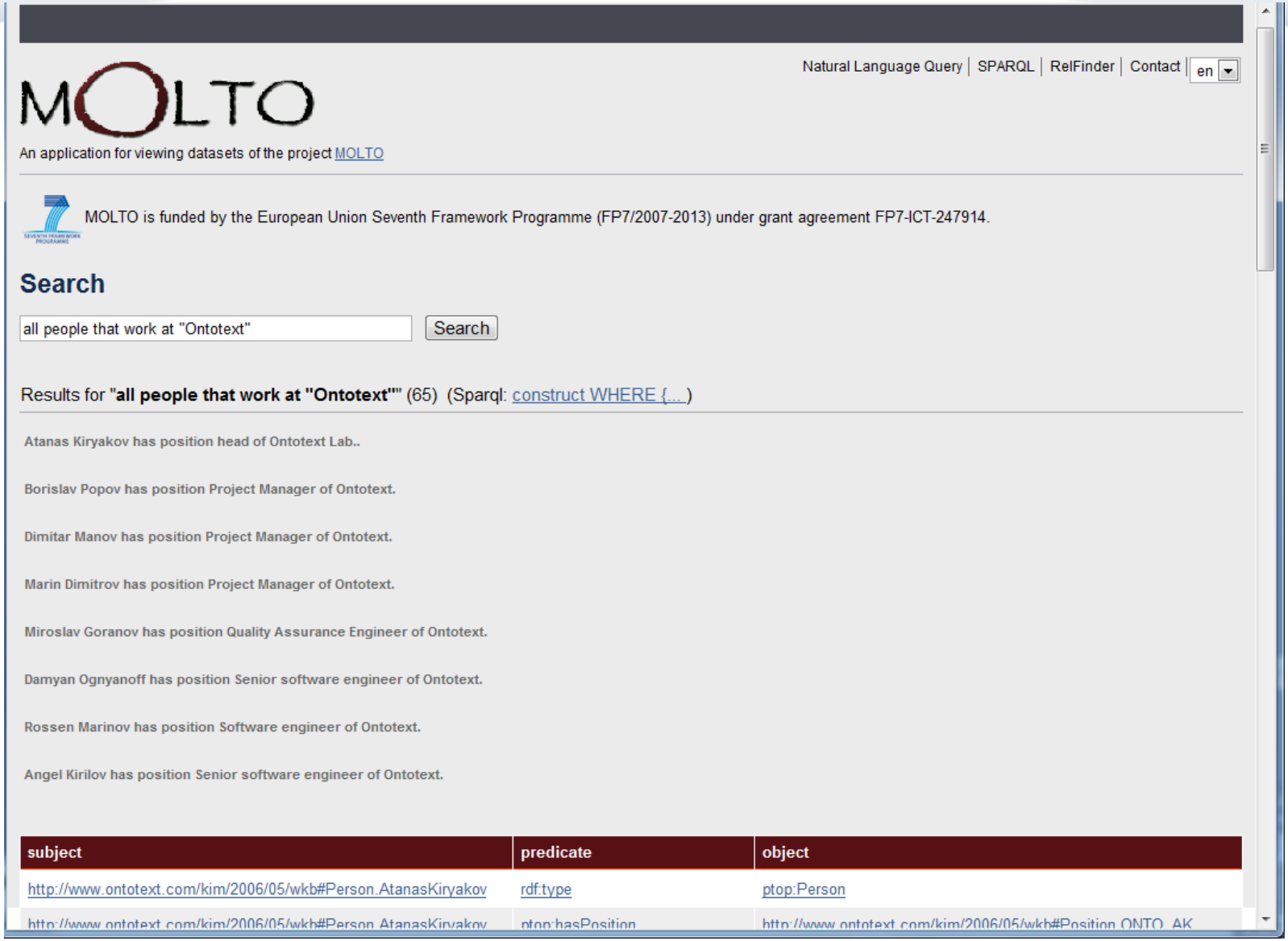


MOLTO

A company of Sirma Group
www.sirma.com

ontotext

Prototype(2)



MOLTO

Natural Language Query | SPARQL | RelFinder | Contact | en

An application for viewing datasets of the project MOLTO

MOLTO is funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement FP7-ICT-247914.

Search

all people that work at "Ontotext" Search

Results for **"all people that work at "Ontotext""** (65) (Sparql: construct WHERE {...})

Atanas Kiryakov has position head of Ontotext Lab..

Borislav Popov has position Project Manager of Ontotext.

Dimitar Manov has position Project Manager of Ontotext.

Marin Dimitrov has position Project Manager of Ontotext.

Miroslav Goranov has position Quality Assurance Engineer of Ontotext.

Damyan Ognyanoff has position Senior software engineer of Ontotext.

Rossen Marinov has position Software engineer of Ontotext.

Angel Kirilov has position Senior software engineer of Ontotext.

subject	predicate	object
http://www.ontotext.com/kim/2006/05/wkb#Person.AtanasKiryakov	rdf:type	ptop:Person
http://www.ontotext.com/kim/2006/05/wkb#Person.AtanasKiryakov	ptop:hasPosition	http://www.ontotext.com/kim/2006/05/wkb#Position.ONTO_AK

Evaluation

- Tripples to NL

$$Recall = \frac{\#number\ of\ retrieved\ right\ results}{\#number\ of\ all\ right\ results} = 95\%$$

$$Precision = \frac{\#number\ of\ retrieved\ right\ results}{\#number\ of\ all\ retrieved\ results} = 76\%$$

- NL Quality

$$Precision = \frac{\#number\ of\ right\ sentances}{\#number\ of\ all\ sentances} = 76\%$$

Thank you for your attention!

Questions

?