



INRIA
SOPHIA ANTIPOLIS



Corese

olivier.corby@inria.fr

Corese/KGRAM

Corese: Conceptual Resource Search Engine

KGRAM: Knowledge Graph Abstract Machine

- RDF/S & SPARQL
- KGRAM: SPARQL 1.1 Interpreter
- Java
- Design by Wimmics team @ Inria/I3S
- <http://wimmics.inria.fr/corese>

Goodies

- SPARQL within XSLT, XPATH, SQL
- Path variable, length, enumeration
- Approximate search
- SPARQL Template Transformation
- SPARQL endpoint
- Pragma
- Inference Rules Engine
- Query several graphs
- Distributed Federated Query

Corese API : Create a Graph

```
import fr.inria.edelweiss.kgraph.core.*;
```

```
Graph g = Graph.create();
```

```
// with RDFS entailments:
```

```
Graph g = Graph.create(true);
```

Corese API : Load RDF/S & Rule

```
Load ld = Load.create(g);
```

```
try {  
    ld.loadWE("/home/kb/schema.rdfs");  
    ld.loadWE("/home/kb/rule.rul");  
    ld.loadWE("/home/kb/data.rdf");  
    ld.loadWE("/home/kb/graph.ttl");  
} catch (LoadException e) {  
}
```

Corese API : Query

```
QueryProcess exec = QueryProcess.create(g) ;
```

```
String query =
```

```
"select * where {?x rdfs:label ?n}" ;
```

```
try {
```

```
    Mappings map = exec.query(query) ;
```

```
}
```

```
catch (EngineException e) {}
```

Corese API : Debug Query

```
String query =  
"select debug * where {?x rdfs:label ?n}" ;
```

Corese API : Result

```
Mappings map = exec.query(q);

for (Mapping m : map) {
    IDatatype dt = (IDatatype) m.getValue("?x");
    dt.stringValue();
    dt.intValue();
    dt.doubleValue();
    dt.booleanValue();
    dt.getDatatypeURI();
}
```

Corese API : Pretty Print Result

```
ResultFormat f = ResultFormat.create(map);  
  
System.out.println(f);  
  
// XML Result (select where)  
// RDF Result (construct where)
```

Corese API : Pretty Print Graph

```
TripleFormat f = TripleFormat.create(g);  
  
System.out.println(f);  
  
// Triple Format
```

Inference Rules

```
ld.loadWE("data.rul");
```

```
RuleEngine re = ld.getRuleEngine();
```

```
re.process();
```

```
g.addEngine(re);
```

```
g.process();
```

Corese HTTP Server

```
java -jar corese-server.jar
```

```
http://localhost:8080/
```