knowledgeweb
realizing the semantic web

Interoperability of Protégé using RDF(S) as Interchange Language

Protégé Conference 2006
24th July 2006

Raúl García Castro
Asunción Gómez Pérez
{rgarcia, asun}@fi.upm.es

Ontology Engineering Group

Index

- **Motivation**
- RDF(S) interoperability benchmarking
- Definition of the benchmark suites
- Protégé's interoperability results
- Recommendations
- Future work

Why benchmarking interoperability?

Ontology developers have **different**:

- Geographical locations
- Expertise
- Tools

Ontology Development Tools have **different**:

- Underlying representation formalisms
- Modelling components inside the same formalism
- Ways of interchanging ontologies:
  - Direct access (APIs)
  - Ontology repositories
  - HTTP servers
  - …

Current situation:

- Unknown interoperability:
  - Information added / lost
- Some experiments show low interoperability (EON 2003)
- Slower uptake by users in academia and industry

How to improve interoperability?

Benchmarking the interoperability of ontology development tools

Benchmarking of several tools

Evaluation of several tools

Evaluation
- Desired quality attributes
- Weaknesses

- Comparative analysis
- Recomendations on tools (users)

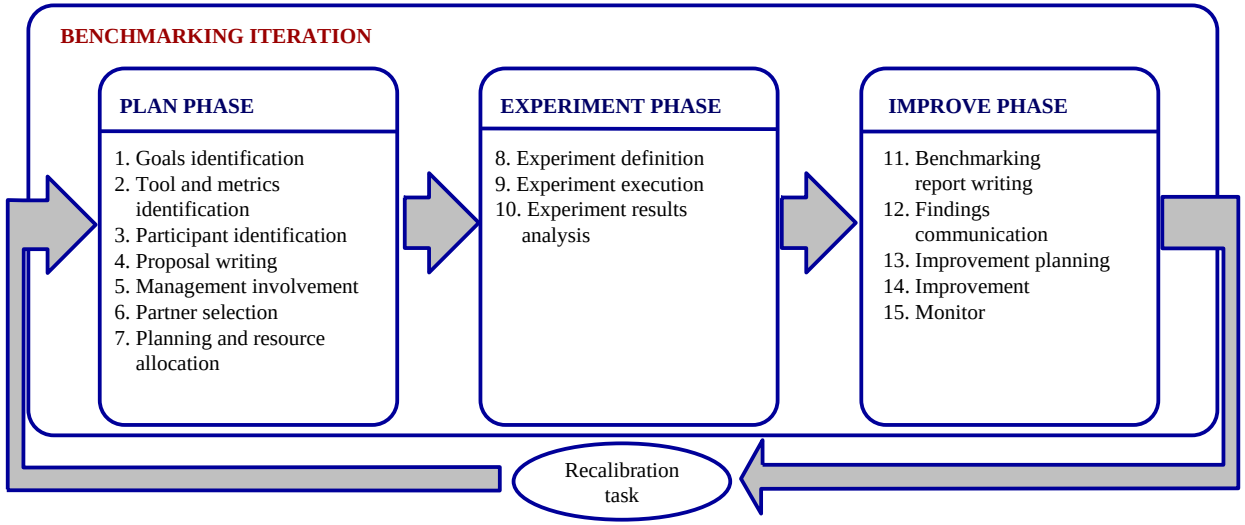
- Continuous improvement
- Recommendations on practices (developers)
- Best practices

Our goal: Support the industrial applicability of ontology tools

Index

- Motivation
- **RDF(S) interoperability benchmarking**
- Definition of the benchmark suites
- Protégé's interoperability results
- Recommendations
- Future work

RDF(S) interoperability benchmarking



GOALS: Assess and improve the interoperability of ontology development tools using RDF(S) as interchange language

Participation

All the benchmarking information is publicly accesible in a web page:

http://knowledgeweb.semanticweb.org/benchmarking_interoperability/

Participation open to any organization

Direct contact with developers of free and commercial tools able of importing/exporting RDF(S).

Announcement through a call for participation:

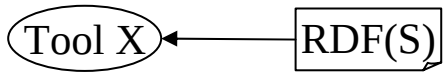
- Semantic Web lists
- Ontology development tool lists

RDF repositories

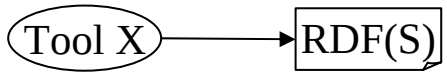
Tool	KAON Tool Suite	*OntoStudio*	protégé (RDF backend)	WebODE	CORESE	semantic web framework Jena	openRDF.org Sesame
Developer	UKARL	Ontoprise	Stanford	UPM	INRIA	HP	Aduna
Participant	UKARL	Ontoprise	UPM	UPM	INRIA	UPM	UPM

Evaluation phases

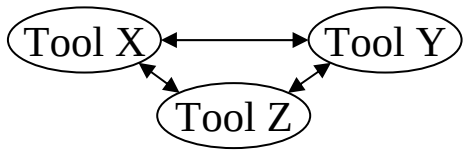
- Evaluation phase 1
  - RDF(S) Import Benchmark Suite



  - RDF(S) Export Benchmark Suite



- Evaluation phase 2
  - RDF(S) Interoperability Benchmark Suite



Index

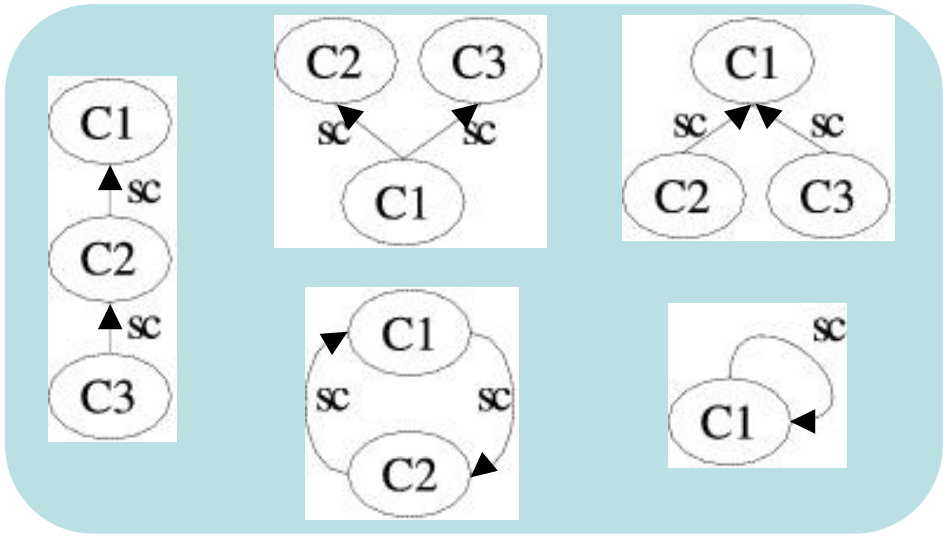
- Motivation
- RDF(S) interoperability benchmarking
- **Definition of the benchmark suites**
- Protégé's interoperability results
- Recommendations
- Future work

RDF(S) benchmark suites definition

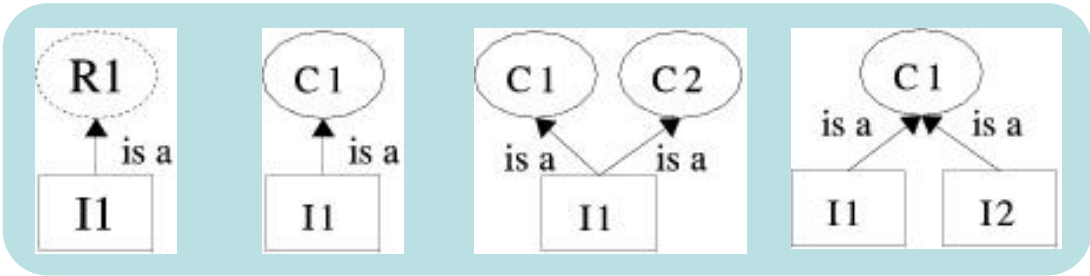
Type of benchmarks	Import	Export	Interoperability
Knowledge Model dependant: • Single components • Two components with a property • More that two components that usually appear together	X	X	X
Syntax dependant: • RDF/XML syntax variants	X		
Component naming dependant: • Character restrictions (RDF, URI, XML)		X	X
To be used by:	Any tool with RDF(S) importer	Any tool with RDF(S) exporter	Any tool with RDF(S) importer and exporter

Example benchmarks

Class hierarchies



Instances

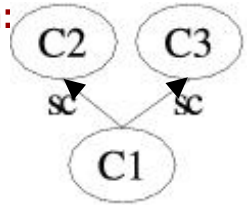


Import benchmark

Identifier: I09

Description: Import one class that is subclass of several classes

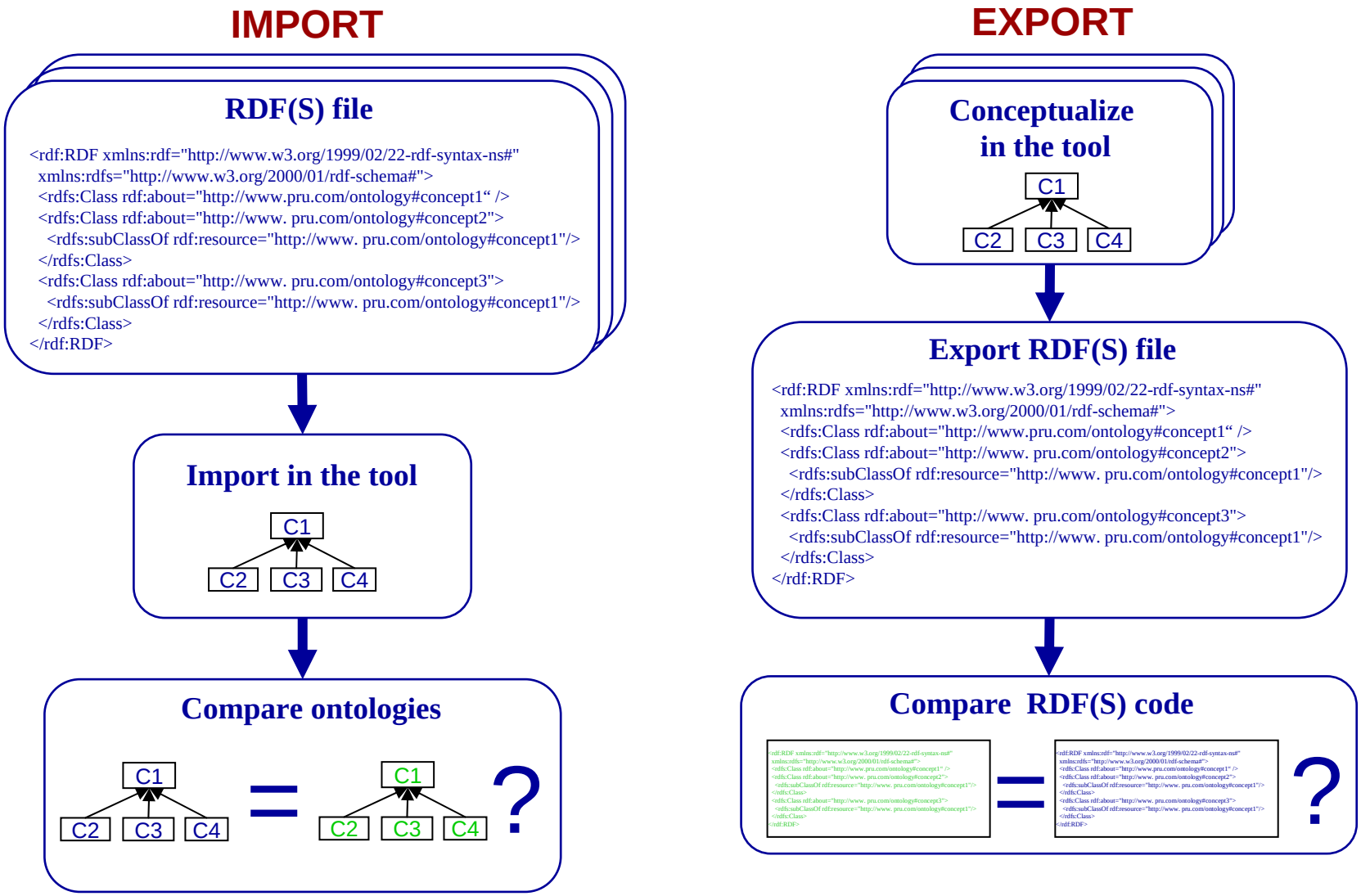
Graphical representation:



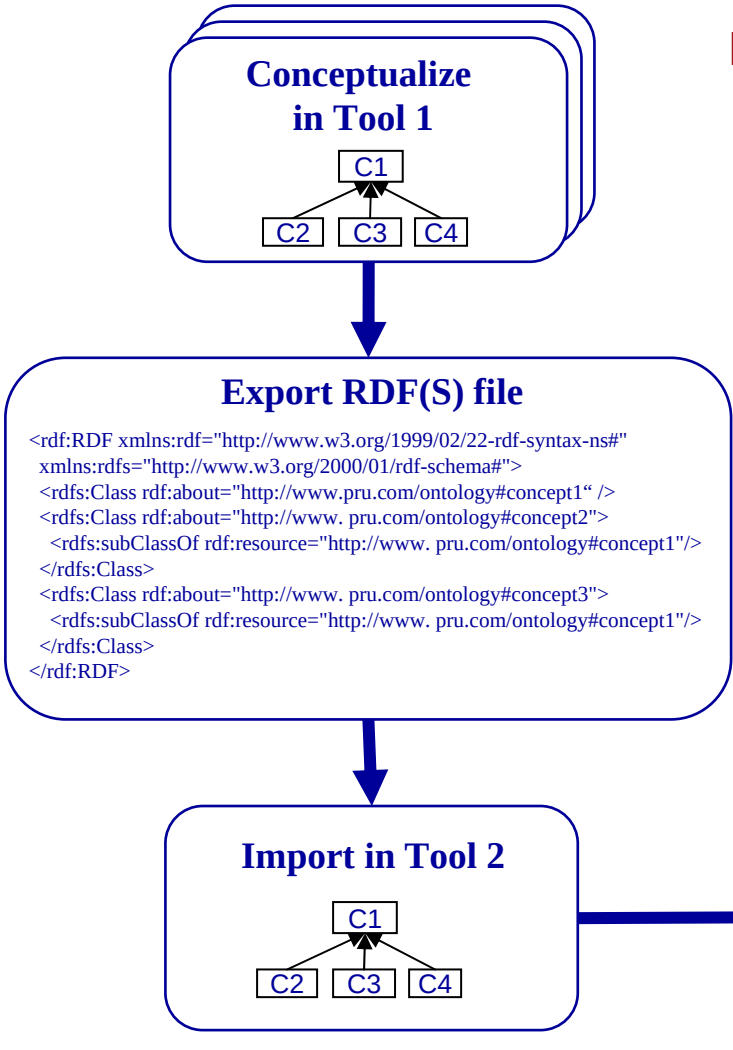
RDF/XML file:

```
<rdf:RDF xmlns="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
   <Class rdf:about="http://www.nothing.org/graph09#class1">
      <subClassOf rdf:resource="http://nothing.org/graph09#class2"/>
      <subClassOf rdf:resource="http://nothing.org/graph09#class3"/>
   </Class>
   <Class rdf:about=http://nothing.org/graph09#class2/>
   <Class rdf:about=http://nothing.org/graph09#class3/>
</rdf:RDF>
```

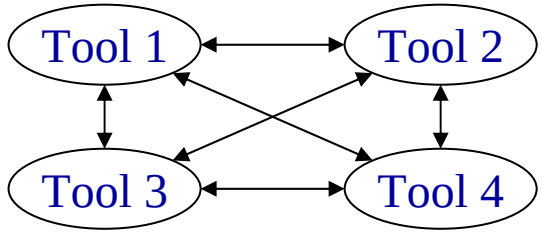
Experiments execution



Experiments execution



INTERCHANGE



Export files are available from the previous experimentation phase.

Benchmark results

- **Modelling** (**YES/NO**)**.** The tool can model the ontology components described in the benchmark.
- **Execution** (**OK/FAIL**)**.** There is no problem while executing the benchmark and the tool produces its expected result.
- **Information added/lost**. The information that is added or lost in the import/export/interchange.

In the export and interoperability benchmark suites:

Execution is **N.E.** if the ontology defined in the benchmark cannot be modelled in a tool.

Index

- Motivation
- RDF(S) interoperability benchmarking
- Definition of the benchmark suites
- **Protégé's interoperability results**
- Recommendations
- Future work

Interoperability

García-Castro et al. Knowledge Web **Deliverable 1.2.2.1.1** Benchmarking ontology development tools using RDF(S) as interchange language. June 2006.

- Import and export results depend mainly on the knowledge model of the tool:
  - RDF(S)-native tools don't need to perform any translation
  - Information addition/loss is common
- Interoperability between the tools depends on:
  - The correct working of their importers and exporters
  - The way of serializing the exported ontologies in the RDF/XML syntax
- Sometimes tools are able to interchange components only in one way

Components modelled by the tools

Combinations	RDF Repositories	KAON	Protégé	WebODE
Classes	X	X	X	X
Classes instance of a single or multiple metaclasses	X	X	X	
Class hierarchies without cycles	X	X	X	X
Class hierarchies with cycles	X			
Classes related through object or datatype properties	X			
Datatype properties without domain or range	X	X	X	
Datatype properties with multiple domains	X	X		
Datatype properties whose range is String	X	X	X	X
Datatype properties whose range is a XML Schema datatype	X	X		X
Object properties without domain or range	X	X	X	
Object properties with a domain and a range	X	X	X	X
Object properties with multiple domains or ranges	X	X		
Property hierarchies without cycles	X		X	
Property hierarchies with cycles	X			
Undefined resources	X			
Instances of a single class	X	X	X	X
Instances of a multiple classes	X	X	X	
Instances related through object or datatype properties	X	X	X	X

Protégé importing and exporting RDF(S)

Protégé deals correctly with most of the components

Main problems:

- Class that is instance of multiple metaclasses
  - Imports/exports the class as instance of only one metaclass
  - Also when a class appears as instance of a metaclass and *rdfs:Class*
- Instance of multiple classes
  - Imports/exports the instance as instance of only one class
- Property with multiple domains
  - Imports/exports the multiple domains
  - In Protégé multiple domains in slots are the union of the domains and in RDF(S) multiple domains in properties are the intersection of the domains

Other issues:

- Represents partially components not present in the destination knowledge model using other components
- Imports XML Schema datatypes as classes
- Crashes when importing some RDF(S) components not modelled by Protégé

Interoperability of Protégé with itself

Protégé interchanges correctly with itself the components that it can model.

Exceptions:

- Classes that are instances of multiple metaclasses
- Instances of multiple classes

Interoperability of Protégé with WebODE

Protégé interchanges correctly with WebODE all the common components that both tools can model.

Interoperability of Protégé with KAON

Protégé interchanges correctly with KAON the common components that both tools can model.

Exceptions:

	Protégé→KAON	KAON→Protégé
Classes that are instances of a single metaclass	OK	FAIL
Classes that are instances of multiple metaclasses	FAIL	FAIL
Instances of multiple classes	FAIL	FAIL
Datatype properties without domain and with range	FAIL	OK
Datatype properties without range	OK	FAIL
Datatype properties whose range is *String*	FAIL	OK
Instances related through datatype properties	FAIL	OK

Interoperability between KAON, Protégé and WebODE

They interchange correctly most of the common components that they can model

Exceptions:

- Datatype properties with domain and whose range is *String*
- Instances related through datatype properties

Regarding URI character restrictions:

- Protégé cannot interchange resources with these restrictions in their names.
- Tools encode some or all the characters that do not comply with these restrictions, changing class and property names

Interchange of components NOT modelled by Protégé:

- The behavior is the same as when importing RDF(S) components not modelled by Protégé

Index

- Motivation
- RDF(S) interoperability benchmarking
- Definition of the benchmark suites
- Protégé's interoperability results
- **Recommendations**
- Future work

Recommendations for ontology developers

When collaboratively developing an ontology using different tools (editors, repositories, etc.):

- Be aware of the knowledge models of the tools and the interchange language:
  - Their components
  - Their semantic equivalences and differences
- Use the common components of these tools to avoid already known knowledge losses
- Don't name resources using spaces or any character that is restricted in the RDF(S), URI or XML specifications
- If the tools have participated in the benchmarking, check the components that they CAN intechange:

http://knowledgeweb.semanticweb.org/benchmarking_interoperability/

Recommendations for software developers

General recommendations:

- Interoperability depends on development decisions:
  - Can produce an interoperability improvement with some tools but a loss with others
  - Their collateral consequences should be analysed
- Semantic equivalences and differences between the knowledge models:
  - Be aware of them
  - Notify the user when the semantics is changed
- Importers and exporters must be robust, dealing with:
  - Unexpected inputs
  - Components of the interchange language
- Serializing:
  - Common components should be completely defined in the file
  - Complete definitions of other components can cause problems to others
  - Resources should have a namespace if the document does not define a default one

Recommendations for improving Protégé's interoperability

Not compulsory to follow, but would solve detected problems:

- Resources that are instances of multiple classes:
  - To export and import correctly classes that are instances of multiple metaclasses (including *rdfs:Class*)
  - To export classes that are instances of a metaclass and metaclasses defining them as a class in the file
  - To export and import correctly instances of multiple classes.
- Properties with multiple domains:
  - To export and import object and datatype properties with multiple domains as properties with no domain, as it occurs in the case of multiple ranges.
- XML Schema datatypes:
  - When importing instances with datatype properties whose range is a XML Schema datatype, to import the literal values as instances of the XML Schema datatype class
- Not to crash when importing:
  - XML Schema datatypes defined as a *rdfs:Datatype*
  - *rdf:datatype* attributes in properties
  - Class and property hierarchies with cycles
  - Resources with empty or blank nodes in the unshortened syntax

Index

- Motivation
- RDF(S) interoperability benchmarking
- Definition of the benchmark suites
- Protégé's interoperability results
- Recommendations
- **Future work**

Future work

Continue the benchmarking

- New tools can join anytime
- Developers are continuously improving their tools
- Develop a web application to exploit the benchmarking results

OWL Interoperability benchmarking

- Follows the same benchmarking methodology and approach
- Right now defining the benchmark suites
- Will automate the execution and analysis of results
- Participation is open to any organization/tool

Join the benchmarking!

- In the definition of the benchmark suites
- In the experimentation with a tool

- RDF(S) benchmarking web page:

  http://knowledgeweb.semanticweb.org/benchmarking_interoperability/

- OWL benchmarking web page:

  http://knowledgeweb.semanticweb.org/benchmarking_interoperability/owl/

Interoperability of Protégé using RDF(S) as Interchange Language

Protégé Conference 2006
24th July 2006

Raúl García Castro
Asunción Gómez Pérez
{rgarcia, asun}@fi.upm.es