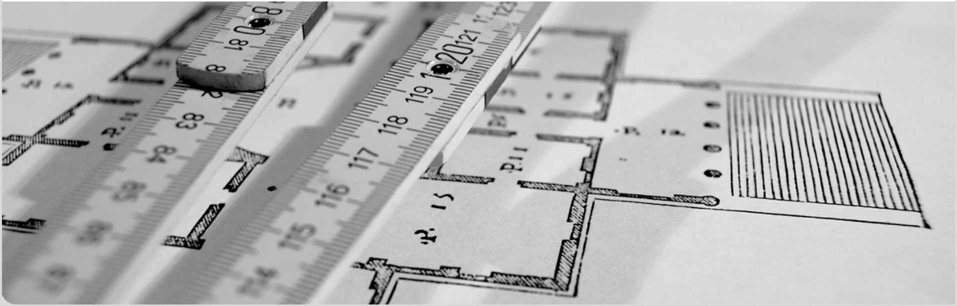
Karlsruhe Institute of Technology

Replica Framework

A Framework for Ontology Sharing and Distributed Ontology Systems

Jan Novacek | August 16, 2011

FZI Department for Information Process Engineering

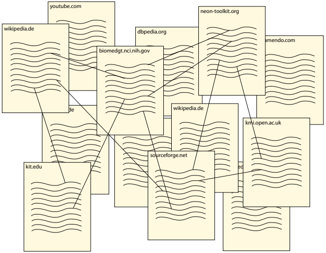


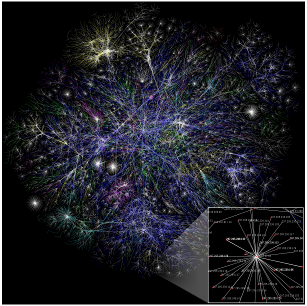
Outline

1. Introduction
   - Semantic Web
   - Ontologies
2. Motivation
   - Purpose and Contribution
   - Usage scenarios
3. Demonstrators
   - Standalone
   - NeOn Toolkit Plugin
4. Conclusions
5. Implementation Details
   - Technologies
   - Shared Ontology Implementation
   - Requirements met
6. References

Semantic Web

- Plenty of information on the internet today, information representations are designed to be used by humans.





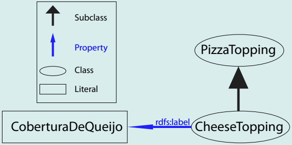
- The approach of the Semantic Web is to augment the existing web with machine processable meta information [Berners-Lee 98].

Ontologies

Ontologies

- Used to represent knowledge in a machine readable form.
- Knowledge modeled as a set of concepts and the relations between these concepts.
- Reasoning systems can be used to infer additional knowledge.

Example



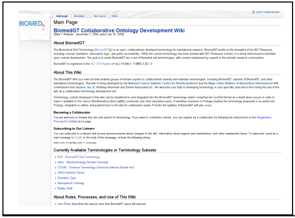
```
<owl:Class rdf:about="#CheeseTopping">
   <rdfs:label xml:lang="pt">
        CoberturaDeQueijo</rdfs:label>
   <rdfs:subClassOf>
      <owl:Class rdf:about="#PizzaTopping"/>
   </rdfs:subClassOf>
</owl:Class>
```

OWL Axioms

Motivation 1/2

An example for a large, complex ontology is BiomedGT

- New terminology aimed at supporting translational research
- Collaborative Ontology Development Wiki
- Ontology experts incorporate and integrate changes
- Builds on the strengths of the NCI Thesaurus[1]



Main Page

BiomedGT Collaborative Ontology Development Wiki

(Beta 1 Release - December 2, 2008, patch Jan 16, 2009)

About BiomedGT

The Biomedical Grid Terminology (BiomedGT) is an open, collaboratively developed terminology for translational research. BiomedGT builds on the strengths of the NCI Thesaurus, including concept orientation, description logic, and public accessibility. While the current terminology has been seeded with NCI Thesaurus content, it is being restructured to facilitate open content development. The goal is to evolve BiomedGT into a set of federated sub-terminologies, with content maintained by experts in the relevant research communities.

BiomedGT is registered in the HL7 OID Registry as 2.16.840.1.113883.3.26.1.3.

About This Wiki

The BiomedGT Wiki is a new tool that enables groups of domain experts to collaboratively develop and maintain terminologies, including BiomedGT, subsets of BiomedGT, and other standalone terminologies. This wiki is being developed by the National Cancer Institute Center for Bioinformatics and the Mayo Clinic Division of Biomedical Informatics with contributions from Apelon, Inc., Northrup Grumman and Dionne-Associates Inc. We welcome your help in developing terminology in your specialty area and in fine tuning the use of the wiki as a collaborative terminology development tool.

Terminology content developed in this wiki can be transferred to and integrated into the BiomedGT terminology and/or converted into LexGrid format as a stand alone source in order to make it available to the Cancer Bioinformatics Grid (caBIG) community and other interested users. A workflow extension to Protege enables the terminology proposals to be pulled into Protege, integrated or edited, and pushed back to the wiki for collaborator review. Periodic full updates of BiomedGT will also occur.

Becoming a Collaborator

You are welcome to browse this site and search for terminology. If you want to contribute content, you can register as a collaborator by following the instructions on the Registration Process for Collaborators page.

Subscribing to Our Listserv

You can subscribe to a listserv and receive announcements about changes to the wiki, information about support and maintenance, and other newsworthy items. To subscribe, send an e-mail message to [1]. In the body of the message, include the following string:

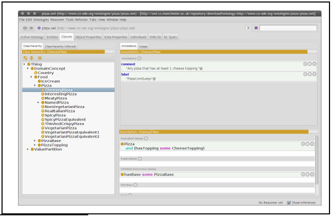
```
subscribe biomedgt_wiki-l yourname
```

Currently Available Terminologies or Terminology Subsets

- BGT - BiomedGT Grid Terminology
- Nano - Nanotechnology Domain Concepts
- CTCAE - Common Terminology Criteria for Adverse Events v4.0
- CRCH Nutrition Terms
- Semantic Type
- Nanoparticle Ontology
- Radlex Draft

About Roles, Processes, and Use of This Wiki

- User Roles describes the various roles that BiomedGT users will assume.



pizza.owl (http://www.co-ode.org/ontologies/pizza/pizza.owl) - [http://owl.cs.manchester.ac.uk/repository/download?ontology=http://www.co-ode.org/ontologies/pizza/pizza.owl]

pizza.owl (http://www.co-ode.org/ontologies/pizza/pizza.owl)

Class hierarchy: CheeseyPizza

```
Thing
  DomainConcept
    Country
    Food
      IceCream
      Pizza
        CheeseyPizza
        InterestingPizza
        MeatyPizza
        NamedPizza
        NonVegetarianPizza
        RealItalianPizza
        SpicyPizza
        SpicyPizzaEquivalent
        ThinAndCrispyPizza
        VegetarianPizza
        VegetarianPizzaEquivalent1
        VegetarianPizzaEquivalent2
      PizzaBase
      PizzaTopping
  ValuePartition
```

Annotations: CheeseyPizza

Annotations

comment
"Any pizza that has at least 1 cheese topping."@

label
"PizzaComQueijo"@

Description: CheeseyPizza

Equivalent classes

Pizza
and (hasTopping some CheeseTopping)

Superclasses

Inherited anonymous classes

hasBase some PizzaBase

Members

[1] The NCI Thesaurus covers vocabulary for clinical care, translational and basic research, and public information and administrative activities, see http://ncit.nci.nih.gov/ncitbrowser/

Motivation 2/2

Collaborative Ontology Development (COD)

- **WHY** Ontology development very hard or impossible for a single person.
- **WHAT** In collaborative work, communication is essential.
- **HOW** Supplying chats, message boards and other tools which assist communication and matters to work collaboratively.

Distributed Ontology System (DOS)

- **WHY** Handle very large Ontologies.
- **WHY** Performance of current reasoners is not sufficient [Chen 09].
- **HOW** Scatter data and query processing across a set of nodes.

Purpose and Contribution

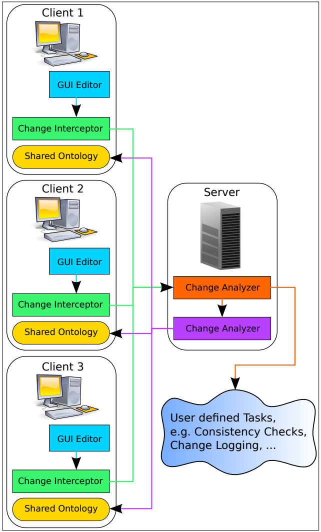
Purpose

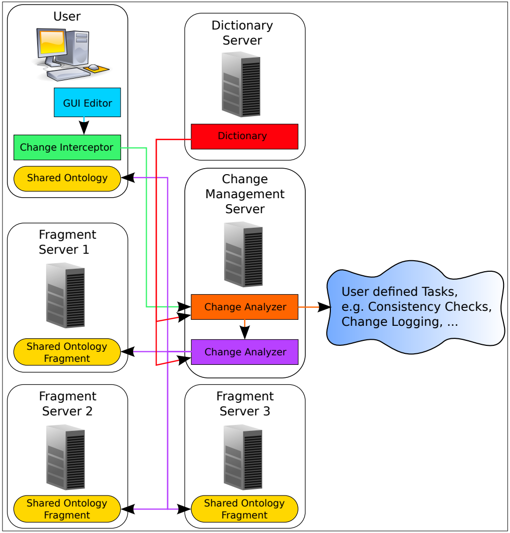
- A novel framework for developing a Distributed Ontology System (DOS) and tools for Collaborative Ontology Development (COD).
- An initial step to met requirements of both fields.

Contribution

- Providing a toolset for developers.
- Supporting the Semantic Web effort.

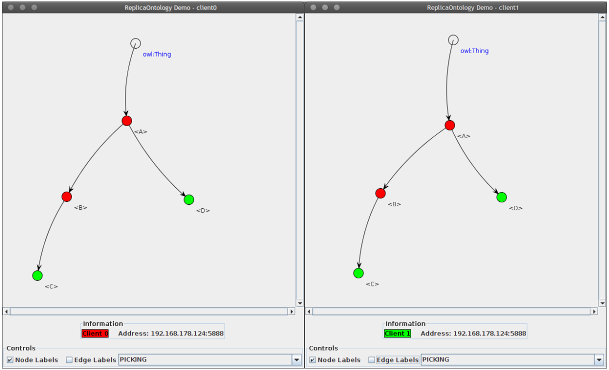
Usage scenarios





Standalone Demonstrator

The Replica Framework demonstrator is a standalone application meant to demonstrate and test the Replica Framework implementation.



ReplicaOntology Demo - client0

Information

Client 0 Address: 192.168.178.124:5888

Controls

☑ Node Labels □ Edge Labels PICKING

ReplicaOntology Demo - client1

Information

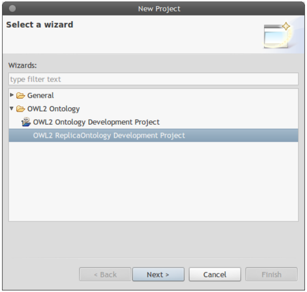
Client 1 Address: 192.168.178.124:5888

Controls

☑ Node Labels □ Edge Labels PICKING

NeOn Toolkit Plugin

For demonstration purposes and evaluating the framework in a real-world application, a plug-in for the NeOn Toolkit[2] Platform has been implemented.



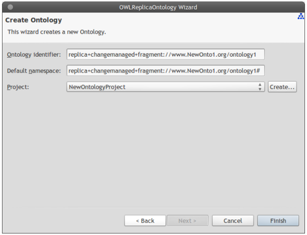
New Project

Select a wizard

Wizards:

type filter text

- General
- OWL2 Ontology
  - OWL2 Ontology Development Project
  - OWL2 ReplicaOntology Development Project



OWLReplicaOntology Wizard

Create Ontology

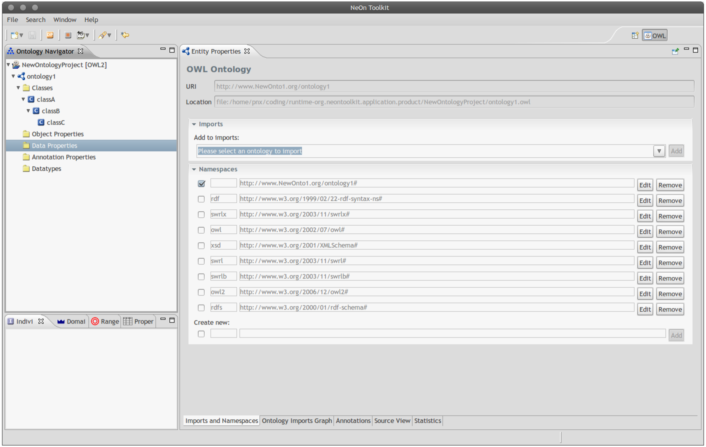
This wizard creates a new Ontology.

Ontology identifier: replica+changemanaged+fragment://www.NewOnto1.org/ontology1

Default namespace: replica+changemanaged+fragment://www.NewOnto1.org/ontology1#

Project: NewOntologyProject

[2] Ontology engeneering environment developed as part of the NeOn Project, `http://neon-toolkit.org/`



- NewOntologyProject [OWL2]
  - ontology1
    - Classes
      - classA
        - classB
          - classC
    - Object Properties
    - Data Properties
    - Annotation Properties
    - Datatypes

OWL Ontology

URI http://www.NewOnto1.org/ontology1

Location file:/home/pnx/coding/runtime-org.neontoolkit.application.product/NewOntologyProject/ontology1.owl

Imports

Add to imports:

Please select an ontology to import ▾ Add

Namespaces

☑		http://www.NewOnto1.org/ontology1#	Edit	Remove
□	rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	Edit	Remove
□	swrlx	http://www.w3.org/2003/11/swrlx#	Edit	Remove
□	owl	http://www.w3.org/2002/07/owl#	Edit	Remove
□	xsd	http://www.w3.org/2001/XMLSchema#	Edit	Remove
□	swrl	http://www.w3.org/2003/11/swrl#	Edit	Remove
□	swrlb	http://www.w3.org/2003/11/swrlb#	Edit	Remove
□	owl2	http://www.w3.org/2006/12/owl2#	Edit	Remove
□	rdfs	http://www.w3.org/2000/01/rdf-schema#	Edit	Remove

Create new:

□ Add

Conclusions

Status

- Framework concept and implementation, demonstrators.
- Some aspects have not been addressed (*Query management*, *policy management* and private/shared workspace support).
- Good performance in unit tests and demonstrators.
  Various improvement possibilities left.

Conclusions

- Unified COD and DOS framework.
- High expandability.
- ECF-based implementation reliable basis for communication.
- Many features remain to be implemented in future work.

Technologies

Karlsruhe Institute of Technology

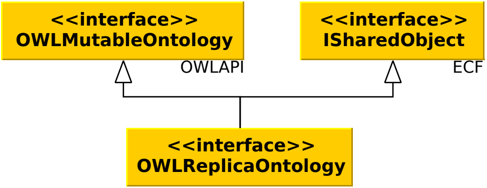
The framework has been implemented in Java.

Core technologies used:

- AspectJ, aspect-oriented programming, aspectj
  → for implementing crosscutting-concerns.
- The Eclipse Communication Framework, eclipse
  → providing a reliable basis for communication.
- Script languages integration Ruby and Groovy, Groovy Ruby
  → for rapid prototyping.
- OSGi service platform, OSGi Alliance
  → modular architecture and service platform.

Shared Ontology Implementation

The shared ontology object is a fundamental component of the framework. To combine the OWLAPI *OWLOntology* interface with the ECF *ISharedObject* interface, the facade *OWLReplicaOntology* was introduced.



For implementing *OWLReplicaOntology* the ECF class *TransactionSharedObject* was extended, a reference implementation of the ECF *ISharedObject* interface, in combination with the proxy pattern for delegating *OWLOntology* method calls.

Requirements met

General [Tudorache 10]

- Scalability, reliability and robustness by implementing Unit tests
- Support for various levels of expressiveness no restrictions in ontology development

COD specific requirements [Tudorache 10]

- Access Control granting access based on user IDs, fine grained control by cross-cutting this concern with AspectJ
- Provenance of information by augmenting change data with meta information
- Communication tooling by leveraging ECF framework components

DOS specific requirements [Hobo 04]

- Heterogeneity by platform abstraction and relying on asynchronous messaging
- Openness by using open standarts and simple APIs for all modules
- Transparency by change filtering, analyzing, logging
- Fault-Tolerance initial step: transactional messaging

References

- Tim Berners-Lee.
  The semantic web road map.
- Xueying Chen, Michel Dumontier.
  A framework for distributed ontology systems.
  2009.
- Remco Hobo.
  Distributed system requirements.
  2011.03.08.
- Tania Tudorache, Natalya Noy, Samson Tu, Mark Musen.
  Supporting Collaborative Ontology Development in Protégé.
  In Amit Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy Finin, Krishnaprasad Thirunarayan, Hrsg., The Semantic Web - ISWC 2008, Band 5318 of Lecture Notes in Computer Science, Kapitel 2, Seiten 17–32. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

License