JOINT

BUILDING REAL-WORLD ONTOLOGY-BASED APPLICATIONS WITH JOINT

Ig Ibert Bittencourt
JOINT
BUILDING REAL-WORLD ONTOLOGY-BASED APPLICATIONS WITH JOINT
Ig Ibert Bittencourt
SUMMARY

- Introduction;
- Motivation;
- JOINT;
- Empirical Studies;
- Further Steps.
In the past few years, the use of ontologies for creating more intelligent and effective application has increased considerably;

Ontologies are not only applied as bases for the Semantic Web, but also in other areas of computing research and industry:

- E-Commerce;
- Media systems;
- Mobile devices;
- Adaptive Education;
- …
Introduction
Motivation

- This dissemination is also a consequence of the growing number of tools and software libraries;

- Not all of them aim to support the development of applications;

- The tools that support it do not provide common functionalities
  - Querying ontologies;
  - Reasoning over rules;
  - Manipulate instances via OO;
  - Manage Repositories...
Motivation

- Learning Curve
  - The tools that support it do not provide common functionalities
    - Querying ontologies;
    - Reasoning over rules;
    - Manipulate instances via OO;
    - Manage Repositories...
  - Communication between stakeholders (Managers, Ontology Engineers, Software Engineers);
Motivation

Development using RDF triples API

```java
... ValueFactory f = myRepository.getValueFactory();

// create some resources and literals to make statements out of
URI alice = f.createURI("http://example.org/people/alice");
URI name = f.createURI("http://example.org/ontology/name");
URI person = f.createURI("http://example.org/ontology/Person");
Literal alicesName = f.createLiteral("Alice");

RepositoryConnection con = myRepository.getConnection();
// alice is a person
con.add(alice, RDF.TYPE, person);
// alice's name is "Alice"
con.add(alice, name, alicesName);
...`
```
Motivation

Development using Object-Oriented paradigm

```java
... ObjectConnection con = repository.getConnection();

// create a Person
Person alice = new Person();
alice.setName("Alice");

// add a Person to the repository
con.addObject(alice);
...```
Motivation

- Learning Curve
  - The tools that support it do not provide common functionalities
    - Querying ontologies;
    - Reasoning over rules;
    - Manipulate instances via OO;
    - Manage Repositories...

- Communication between stakeholders (Managers, Ontology Engineers, Software Engineers);
- Transparency (Software development x Ontology);
- Development Productivity;
- Performance and Memory.
Further Steps

- Conceptualization
- Conceptual Modelling
- Formalization
- Computational Artifact (OWL, RDF-S)
- Transformation
- OO Code (Java)
- Software Implementation
- Ontology-based Applications (SW, LD)
JOINT - Team

Ig Ibert

Seiji Isotani

Armando Barbosa
Olavo Holanda
Endhe Elias
Williams Alcantara
Judson Bandeira
JOINT:

- **JOINT:** Java Ontology Integrated Toolkit;

- A toolkit that supports the development of ontology-based applications;

- It provides an integration of existing technologies and techniques to create a unified environment;

- Main goal: Easy and efficient development of ontology-based application.
Services provided:

- Instances manipulation – CRUD (Object-oriented paradigm):
  - Core of the system;
  - Provide detached objects;
  - Sesame as the underlying main tool.

- Ontology operations (add, remove, retrieve, consistency checking);
- Java code generator;
- Queries with SPARQL;
- Reasoning over SWRL;
JOINT - Architecture

- Knowledge Access Object Pattern
- Object-Ontology Mapping
- Ontology Operations
- Reasoner Module
- Repository Operations
- Sesame API

Ontologies

Triple Store Tools
JOINT – KAO Pattern

- The abstract class “AbstractKAO” provides the CRUD and SPARQL operations;

- The JAVA SPARQL methods are protected;

- For each ontology, a concrete class OntologyAKAO must be created:
  - All queries related to this ontology, stay in this KAO.
Interface OnlineAccount

```java
// An online account. *
public interface OnlineAccount extends Thing {

  /** Indicates the name (identifier) associated with this online account. */
  @id(@Iri("http://xmlns.com/foaf/0.1/accountName"))
  @name("http://xmlns.com/foaf/0.1/accountName")
  Set<Object> getFoafAccountName();

  /** Indicates the name (identifier) associated with this online account. */
  void setFoafAccountName(Set<?> foafAccountName);

  /** Indicates a homepage of the service provide for this online account. */
  @id(@Iri("http://xmlns.com/foaf/0.1/foafAccountServiceHomepage"))
  @name("http://xmlns.com/foaf/0.1/foafAccountServiceHomepage")
  Set<Document> getFoafAccountServiceHomepage();

  /** Indicates a homepage of the service provide for this online account. */
  void setFoafAccountServiceHomepage(Set<?> extends Document>
                     foafAccountServiceHomepage);
}
```

Class OnlineAccountImpl

```java
/** An online account. */
public class OnlineAccountImpl extends JOINTResource implements OnlineAccount, Serializable {

  public OnlineAccountImpl(){
    this.inerModifiedFields = new ArrayList<>();
  }

  private Set<Object> foafAccountName;

  /** Indicates the name (identifier) associated with this online account. */
  @id("http://xmlns.com/foaf/0.1/accountName")
  @name("http://xmlns.com/foaf/0.1/accountName")
  public Set<Object> getFoafAccountName(){
    if(!this.isLazyLoaded())
      LazyLoader.loadObject(this, this.getClass().getName());
    return this.foafAccountName;
  }

  /** Indicates the name (identifier) associated with this online account. */
  @id("http://xmlns.com/foaf/0.1/accountName")
  @name("http://xmlns.com/foaf/0.1/accountName")
  public void setFoafAccountName(Set<?> foafAccountName){
    this.inerModifiedFields.add("FoafAccountName");
    this.foafAccountName = (Set<Object>) foafAccountName;
  }
}
```
Lazy Load

Load at First level

Load at First level

Lazy Load
Empirical Studies
Empirical Studies

- Development Productivity
- Performance and use of memory
- Development of Real-World Apps
### Empirical Study Development Productivity

<table>
<thead>
<tr>
<th>Group</th>
<th>Machines</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>F5</td>
<td>Intel Core i5, 4GB de RAM</td>
<td>JOINT</td>
</tr>
<tr>
<td>F3</td>
<td>Intel Core i3, 4GB de RAM</td>
<td>JOINT</td>
</tr>
<tr>
<td>J5</td>
<td>Intel Core i5, 4GB de RAM</td>
<td>Jastor and Jena</td>
</tr>
<tr>
<td>J3</td>
<td>Intel Core i3, 4GB de RAM</td>
<td>Jastor and Jena</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group</th>
<th>Development Time</th>
<th>Codes Lines</th>
<th>Running Time</th>
<th>Memory Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>F5</td>
<td>7 hours</td>
<td>72 lines</td>
<td>2584 ms</td>
<td>15.4 MB</td>
</tr>
<tr>
<td>F3</td>
<td>6 hours</td>
<td>81 lines</td>
<td>3757 ms</td>
<td>16.2 MB</td>
</tr>
<tr>
<td>J5</td>
<td>15 hours</td>
<td>89 lines</td>
<td>4070 ms</td>
<td>61.5 MB</td>
</tr>
<tr>
<td>J3</td>
<td>18 hours</td>
<td>84 lines</td>
<td>4144 ms</td>
<td>52.2 MB</td>
</tr>
</tbody>
</table>
Experiment

Research Hypotheses

- Q1: Are there differences in performance between JOINT and Alibaba regarding the create, retrieve and update operations? If yes, which tool is the best?
- Q2: Are there differences in memory usage between JOINT and Alibaba regarding the create, retrieve and update operations? If yes, what tool uses less memory?

<table>
<thead>
<tr>
<th>Hipótese</th>
<th>Hipótese Nula</th>
<th>Hipótese Alternativa</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>H1-0: T(F1, O1) = T(F2, O1)</td>
<td>H1-1: T(F1, O1) ≠ T(F2, O1)</td>
</tr>
<tr>
<td>H2</td>
<td>H2-0: T(F1, O2) = T(F2, O2)</td>
<td>H2-1: T(F1, O2) ≠ T(F2, O2)</td>
</tr>
<tr>
<td>H3</td>
<td>H3-0: T(F1, O3) = T(F2, O3)</td>
<td>H3-1: T(F1, O3) ≠ T(F2, O3)</td>
</tr>
<tr>
<td>H4</td>
<td>H4-0: M(F1, O1) = M(F2, O1)</td>
<td>H4-1: M(F1, O1) ≠ M(F2, O1)</td>
</tr>
<tr>
<td>H5</td>
<td>H5-0: M(F1, O2) = M(F2, O2)</td>
<td>H5-1: M(F1, O2) ≠ M(F2, O2)</td>
</tr>
<tr>
<td>H6</td>
<td>H6-0: M(F1, O3) = M(F2, O3)</td>
<td>H6-1: M(F1, O3) ≠ M(F2, O3)</td>
</tr>
</tbody>
</table>
Experiment
Interaction Time

Create

Retrieve

Update

Chart: Comparison of Interaction Time for Create, Retrieve, and Update operations.
Experiment

Memory Usage

Create

Retrieve

Update
<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Null Hypothesis</th>
<th>Alternative Hypothesis</th>
<th>Description</th>
<th>Results (SIG. 0.05)</th>
<th>Better Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>H1-0: T(F1, O1) = T(F2, O1)</td>
<td>H1-1: T(F1, O1) ≠ T(F2, O1)</td>
<td>Interaction time at create</td>
<td>With statistical difference</td>
<td>JOINT-DE</td>
</tr>
<tr>
<td>H2</td>
<td>H2-0: T(F1, O2) = T(F2, O2)</td>
<td>H2-1: T(F1, O2) ≠ T(F2, O2)</td>
<td>Interaction time at retrieve</td>
<td>With statistical difference</td>
<td>JOINT-DE</td>
</tr>
<tr>
<td>H3</td>
<td>H3-0: T(F1, O3) = T(F2, O3)</td>
<td>H3-1: T(F1, O3) ≠ T(F2, O3)</td>
<td>Interaction time at update</td>
<td>With statistical difference</td>
<td>JOINT-DE</td>
</tr>
<tr>
<td>H4</td>
<td>H4-0: M(F1, O1) = M(F2, O1)</td>
<td>H4-1: M(F1, O1) ≠ M(F2, O1)</td>
<td>Average of memory at create</td>
<td><strong>Without statistical difference</strong></td>
<td>-</td>
</tr>
<tr>
<td>H5</td>
<td>H5-0: M(F1, O2) = M(F2, O2)</td>
<td>H5-1: M(F1, O2) ≠ M(F2, O2)</td>
<td>Average of memory at retrieve</td>
<td>With statistical difference</td>
<td>JOINT-DE</td>
</tr>
<tr>
<td>H6</td>
<td>H6-0: M(F1, O3) = M(F2, O3)</td>
<td>H6-1: M(F1, O3) ≠ M(F2, O3)</td>
<td>Average of memory at update</td>
<td>With statistical difference</td>
<td>Alibaba</td>
</tr>
</tbody>
</table>
Real-World Application
MeuTutor

MeuTutor - ENEM: Ambiente educacional Web inteligente para preparação de alunos para o Exame Nacional do Ensino Médio (ENEM);

- Focado no acompanhamento individualizado da aprendizagem;
- Possui três aspectos principais:
  - Gamificação;
  - Aprendizagem personalizada;
  - Experiência social.

Olavo Holanda

BIOLOGIA
Clique no assunto para iniciar seus estudos

A Origem da Vida

Histologia

Reprodução e Desenvolvimento Celular

Centro De Estudos Psicopedagogicos
Maceió - AL

220 Pontos ganhos
10 Troféus ganhos
89%
Results

Answer Time - Login

Memory Usage - Login

W3C | Ig Ibert Bittencourt | ig.ibert@ic.ufal.br
Other cases

- A semantic tool to assist authors in the instantiation of software product lines for intelligent tutoring systems context.

- An agent-based semantic web blog crawler;

- Educational resources recommendation system based on agents and semantic web for helping students in a virtual learning environment;

- Linked Knowledge.
Limitations

- Lazy Load x Eager Load;

- Inference is in Triple Stores and Editors;

- More empirical tests about performance and memory usage;

- Linked Closed Data x Linked Open Data;

- OWL Profiles: OWL 2 EL, OWL 2 RL and OWL 2 QL.
Further Steps

Methodology (S.E. + O.E.) + Good Practices

1. Conceptualization
2. Conceptual Modelling
3. Formalization
4. Computational Artifact (OWL, RDF-S)
5. Transformation
6. OO Code (Java)
7. Software Implementation
8. Ontology-based Applications (SW, LD)
Thank You!

www.nees.com.br

www.jointnees.sourceforge.net