Using Graph Databases to Represent Knowledge Base in the Field of Cultural Heritage

Galina Bogdanova[1], Todor Todorov[1, 2], Nikolay Noev[1]

[1]Institute of mathematics and informatics, BAS
[2]St. Cyril and St. Methodius University of Veliko Tarnovo
galina@math.bas.bg, todor@math.bas.bg, nickey@math.bas.bg

Abstract. This paper presents digital knowledge including context-based annotations of objects in the field of cultural heritage. Some major problems and solutions of digitalization of items, their processing, storing and organizing them in a repository are highlighting. We show how graph databases could be used as data management platform.

Keywords: semantic web, ontology, graph databases, data management platform

1 Introduction

The creation of new digital content of objects in the field of cultural heritage is an important task for the future progress of society.

In this paper are presented methods for organization and management of digital objects in the field of cultural heritage. It is very important to choose the proper data management system for such objects. All digitized objects and their metadata are with specific and nonunified structure. That is why we need a proper database in order to represent such a data model.

At first is shown building of semantic-based knowledge of specific area. At second are presented effective methods and platforms for storage and management of that knowledge.

This research is a result of years of investigations and analyzes of approaches and methods of creating and presenting digital content in the field of cultural heritage, performed by many scientists of different areas and different on various applied-research projects[1].

[1] Partly funded by program BG08 "Cultural heritage and contemporary arts", project "Digital cultural heritage "North+": documentation, preservation and public access to cultural heritage in libraries, museums, archives and galleries in North and Central Bulgaria".

2 Semantic based architecture of knowledge of objects in the field of cultural heritage

The main subject of the idea of Semantic Web, offered by Tim Berners-Lee [1] consist in automation of “intelligent” processing of knowledge of different Internet resources or is explanation and transformation of knowledge at machine-interpretable definitions, through which the computer semantic agents could draw conclusions.

The data described by ontology is interpreted as a set of “objects” and a set of “properties” to interact with one another. The ontology also contains a set of “axioms” which place restrictions on “individuals” and a type of allowed relations between them.

Semantic description of the bell includes concepts, relations, rules, restrictions, individuals and facts applicable for the subject area. The selection of basic concepts is based on real settings, situations and facts [7].

Ontological model of bell objects, along with its main elements (individuals, properties, classes and relations) is presented at [2, 3, 4].

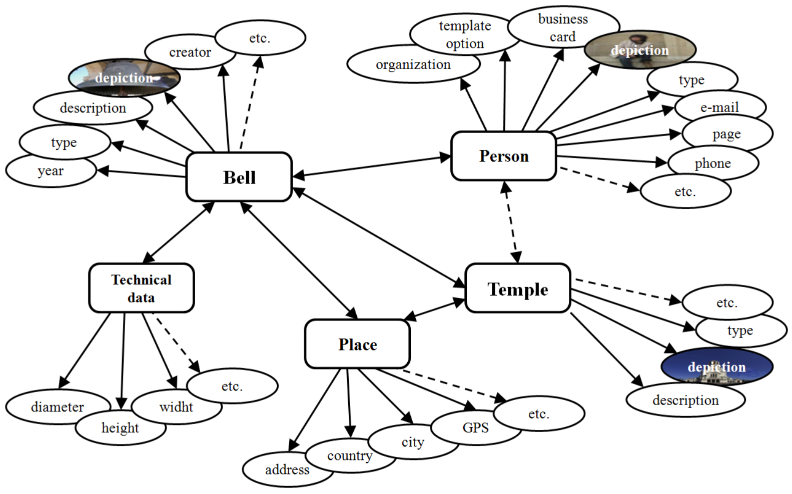

Fig. 1. Scheme of relations of bells objects

3 Usage of Neo4j to represent BELL ontology

3.1 Graph database

NoSQL [5, 6, 9] graph databases is a database that uses graph structures for semantic queries with nodes, edges and properties to represent and store data.

Retrieving data from a graph database requires new concepts and generally a new query language.

The main advantage of graph databases is the versatility, as they can store both relational and documentary and complex semantic data [8].

3.2 Neo4j

Neo4j is a highly scalable, native graph database purpose-built to leverage not only data but also its relationships [10]. Graph databases have three other key advantages:

Performance

For intensive data relationship handling, graph databases improve performance by several orders of magnitude. Graph database performance stays constant even as your data grows year over year.

Flexibility

With graph databases, data architect and information technology (IT) teams move at the speed of business because the structure and schema of a graph model flexes as applications and industries change.

Agility

Developing with graph databases aligns perfectly with today's agile, test-driven development practices, allowing your graph database to evolve in step with the rest of the application and any changing business requirements.

3.3 Cypher

While most relational databases use a form of SQL as their query language, the graph database world is more varied so we'll look specifically at a single graph database query language Cypher.

The system can execute direct queries written in Cypher. Cypher supports queries with parameters which are submitted as JSON[2].

Cypher query:

```
MATCH (x { name: { startName }})-[r]-(friend)
WHERE friend.name = { name }
RETURN TYPE(r)
```

Request:

```
{   "query" : MATCH (x { name: { startName }})-[r]-
(friend) WHERE friend.    name = { name } RETURN TYPE(r),
```

[2] The Neo4j REST API documentation v3.0, http://neo4j.com/docs/rest-docs/current/

```
  “params” : {
    “startName" : "I",
    "name" : "you"
  } }
```

Response:

```
{ columns” : [ "TYPE(r)” ],
  "data" : [ [ "know" ]] }
```

3.4 Import Bell ontology in Neo4j

To import data from OWL file to Neo4j DB we use OWL API[3, 4].

Step 1

Create an OWLOntologyManager object. The manager will load and save ontologies.

```
OWLOntologyManager
m=OWLManager.createOWLOntologyManager();
```

Load ontology from file into OWLOntology object.

```
File f=new File("bell.owl");
OWLOntology o = m.loadOntologyFromOntologyDocument(f);
```

Next we create an instance of HermiT Reasoner and start import operations:

```
OWLReasoner reasoner = new Reasoner(ontology);
```

Step 2

Create a starting node in Neo4j representing the owl:Thing node. This is the root node of the graph we’re going to create.

```
Node thingNode = getOrCreateNodeWithUniqueFacto-
ry("owl:Thing");
```

Step 3

Get all the classes defined in the ontology and add them to the graph.

```
for (OWLClass c :ontology.getClassesInSignature(true))
// Create nodes from classes
```

[3] Neo4j and OWL, https://neo4j.com/blog/using-owl-with-neo4j/
[4] OWL API, http://owlapi.sourceforge.net/

```
Node classNode = getOrCreateNodeWithUniqueFacto-
ry(classString);
```

Step 4

Create relations. If no super class exists then we link back to owl:Thing. The relationship type used to express the rdf:type property is a custom one named “isA”:

```
NodeSet<OWLClass> superclasses = reason-
er.getSuperClasses(c, true);
if (superclasses.isEmpty()) {
 classNode.createRelationshipTo(thingNode, DynamicRela-
tionshipType.withName("isA")); }
else { for
(org.semanticweb.owlapi.reasoner.Node<OWLClass>
  parentOWLNode: superclasses) {
 OWLClassExpression parent
=parentOWLNode.getRepresentativeElement();
     . . . . . . . . . . . . . . . . . . . . . .
 Node parentNode = getOrCreateNodeWithUniqueFacto-
ry(parentString);
 classNode.createRelationshipTo(parentNode, DynamicRela-
tionshipType.withName("isA"));}}
```

Step 5

Now For each class we create nodes and link them back to their parent class.

```
for
(org.semanticweb.owlapi.reasoner.Node<OWLNamedIndividual>
in
 : reasoner.getInstances(c, true)) {
    . . . . . . . . . . . . . . . . . . . . . .
 Node individualNode =
 getOrCreateNodeWithUniqueFactory(indString);
 individualNode.createRelationshipTo(classNode,
 DynamicRelationshipType.withName("isA")); }
```

Step 6

Add object properties to the graph as node properties or relationships.

```
for (OWLObjectPropertyExpression objectProperty:
ontology.getObjectPropertiesInSignature()) {
 for
(org.semanticweb.owlapi.reasoner.Node<OWLNamedIndividual>
  object: reason-
er.getObjectPropertyValues(i,objectProperty)) {
     . . . . . . . . . . . . . . . . . . . . . .
  Node objectNode = getOrCreateNodeWithUniqueFactory(s);
  individual-
Node.createRelationshipTo(objectNode,DynamicRelationshipT
ype.withName(reltype));}}
  for (OWLDataPropertyExpression dataProperty:
ontology.getDataPropertiesInSignature()) {
  for (OWLLiteral object: reasoner.getDataPropertyValues(
i, dataProperty.asOWLDataProperty())) {
    . . . . . . . . . . . . . . . . . . . . . .
    individualNode.setProperty(reltype, s);}}}}
```

In our experiments we also used Neo4jphp (PHP library wrapping the Neo4j graph database) that supports executing both Cypher and Gremlin queries via REST.

4 Conclusion

We present an ontological model of knowledge in specific subject area in a field of cultural and historical heritage and describing of objects in the field of cultural heritage. In detail is described the semantics of the subject area, by defining many features, characteristics, metadata, rules, classes of knowledge and relationships between them. We propose a graph database approach for storing and managing data in the ontological model.

5 References

1. Berners-Lee T., Fischetti M., Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its inventor., 1999;
2. Bogdanova G., Stoffel K., Todorov T., Noev N., Building OWL Ontology of unique Bulgarian bells using Protégé platform, International conference Digital Preservation and Presentation of Cultural and Scientific Heritage - DiPP`12, V. Tarnovo, Bulgaria, 18-21 September 2012, pp. 161-166, ISSN: 1314-4006, 2012
3. Bogdanova G., Todorov T., Noev N., Building OntoWiki Ontology as a Part of Bells Knowledge, In: Proceedings ot the UNESCO International Conference on Digital Preservation and Presentation of Cultural and Scientific Heritage (DiPP`15), Veliko Tarnovo, Bulgaria, 28-30 September, 2015, pp. 29-34, ISSN: 1314-4006, 2015
4. Bogdanova G., Todorov T., Noev N., Semantic Model of Digital Resources of Bulgarian Bells, Mathematica Balkanica, NewSeries Vol. 25, 2011, ISSN 0205-3217, Fasc. 5, pp. 483-490, 2011
5. Grolinger K., W. A . Higashino, A. Tiwari, M. Capretz, "Data management in cloud environments: NoSQL and NewSQL data stores", JoCCASA, Springer, 2014.
6. Mohan C., History Repeats Itself: Sensible and NonsenSQL Aspects of the NoSQL Hoopla (PDF). Proc. 16th Int'l Conf. on Extending Database Technology, 2013.
7. Paneva-Marinova D., R. Pavlov, M. Goynov, L. Pavlova-Draganova, L. Draganov (2010), Search and Administrative Services in Iconographical Digital Library, In the Proceedings

of the International Conference „Information Research and Applications" – i.Tech 2010, July, 2010, Varna, Bulgaria, pp. 177-187

8. Robinson I., Webber J., Eifrem E., Graph Databases, O'Reilly Media, 2013.
9. Shashank T., Professional NoSQL, John Wiley & Sons, 2011.
10. Van Bruggen R., Learning Neo4j, Packt Publishing, 2014.