Bull. Nov. Comp. Center, Comp. Science, 34 (2012), 43–54

Experiments on ontology based semantic systems integration*

Z.V. Apanovich, A.G. Marchuk

Abstract. This paper describes starting experiments on integration of semantic systems based on ontologies. The experiments are carried out with the help of a toolkit intended to simplify visual analysis and integration of data from different datasets. The toolkit comprises several tools for application-specific visualization as well. The Bone ontology and the AKT Reference Ontology along with their datasets are used as test examples.

Introduction

Due to the fast progress of Semantic Web and its new branch, Linked Open Data, large amounts of structured information from various fields are getting available on the Web. The Web of Data forms a single global data space and consists currently of over 28 billion RDF triples. There arise new applications trying to integrate and use information from different data sources.

A four-step strategy for integration of Linked Data into an application is proposed in [20]. The problems of access to linked data (1), vocabularies (schema, ontology) normalization (2), identity resolution (3), and data filtering (4) should be solved manually or semi-automatically in addition to the application specific problems. Specialized tools for solving separate problems started to appear [7, 8, 10, 15, 21]. However, according to [21] the large scale processing, schema mapping and data fusion are still in their infancy. On the other hand, problem (1) can be solved by creating a SPARQL endpoint for the local data set and by downloading the RDF dump of external datasets. Problem (2) can be solved by means of specialized SPARQL queries. These queries can be generated on the base of the ontology visualization. Semi-automatic tools such as SILK[10] and LIMES[15] exist for solution of problem (3). However, for the datasets of moderate size it can be solved manually. Problem (4) can be solved by SPARQL-queries as well. We just have to import different data sets into distinct Named Graphs and to query them separately using the SPARQL GRAPH clause. It means that an ontology visualization tool and a tool for SPARQL-queries processing can be used as a starting point in our work. Section 1 demonstrates specific features

*Supported by RFBR under Grant 11-07-00388-a and SBRAS, Project 15/10.

of our ontology visualization tool in the context of Linked data integration. Two ontologies, the BONE ontology of the Open Archive of the Siberian Branch of the Russian Academy of Sciences (SB RAS Open Archive) and AKT Reference Ontology are used as test examples. Their structures are compared and a strategy of links creation between the data sets based on these two ontologies is discussed. An example explaining why existing tools of ontology alignment are of little help in our case is also demonstrated. A simplified version of a SPARQL-query establishing correspondence between groups of classes and relations of the two ontologies is presented. The problem of identity resolution is discussed on the example of the DBLP and the Open Archive datasets. Section 2 demonstrates a tool for SPARQL-queries creation and visualization.

1. Comparison of the BONE and AKT Reference Ontologies by means of visualization

One of the projects conducted in the A.P. Ershov Institute of Informatics Systems is aimed at investigation of the Linked Data technology and enrichment of the SB RAS Open Archive [3] with the data of the Open Linked Data cloud [5]. The main content of the SB RAS Open Archive constitute various documents (photo documents mainly) reflecting information about people, scientific organizations and major events of the SB RAS since 1957. There can be found information about jobs, scientific achievements, state awards, titles, participation in academic and social events for each person mentioned in the Open archive. It contains 20 505 photo documents, facts about 10 917 persons and 1519 organizations and events. The data sets of the Open Archive are available as RDF triple store as well as Virtuoso endpoint [7] for Archive of SB RAS [http://duh.iis.nsk.su/VirtuosoEndpoint/Home/Index]. Its RDF triple store comprises about 600 000 RDF triples. The structure of the Open Archive knowledge base is organized with the so-called Basic Ontology for Non-specific Entities (BONE), described in OWL and comprising 44 classes. Classes and relations of the BONE ontology are shown in Figure 1. This figure demonstrates some specific features of the BONE ontology as well as specific features of our way of ontology visualization. The ontology visualization is constructed by the hierarchical edge bundles method [9]. Nodes correspond to ontology classes and edges correspond to ontology relations. Tree edges represent the *rdfs:subClassOf* links. They can be drawn either by the radial or circular tree drawing algorithm or by the layered drawing algorithm for directed graphs. Curvilinear edges represent the *owl:ObjectProperty* relationships and are drawn above the taxonomy drawing.

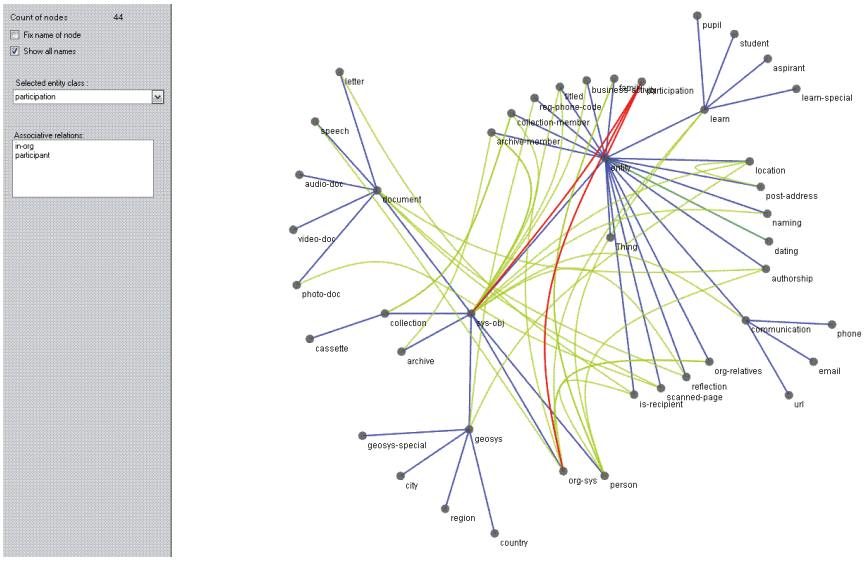

Figure 1. Classes and relations of the BONE ontology

This way of edge drawing addresses the scalability problem mentioned in [11]: "The visualization of relation links is also problematic and the display becomes cluttered very quickly... TGViz[2] and OntoViz[18] became impossible to use when relation links were visible... The most of visualizers prefer not to display them at all". However, the relation links visualization is getting even more important in the Linked Open Data context since the *owl:objectProperty* links correspond to the RDF predicates used in the LOD data sets descriptions. As the edge bundles method reduces the clutter it is very well suited for the relation links representation. Moreover, different shape of edges depending on their type improves ontology comprehensibility. One more scalability issue of [11] is met because we have significantly reduced the number of nodes in the ontology view by displaying only the class nodes. The visualization of instances of classes is delegated completely to the SPARQL visualization component that is much more flexible.

There exists a "Select class" listbox and an "Associative relations" drop down list in the left part of the ontology visualization panel intended for investigation of edges corresponding to the *owl:objectProperty*. When a user selects an item in the "Select class" list box, all the edges connecting the chosen class with other classes are displayed in the visualization panel. Simultaneously, the names of links incident to the selected class node are displayed in the "Associative relations" dropdown list. For example, the "participation" class node selected in Figure 1 has links connecting this node to the "person" and "org-sys" classes.

This kind of visualization shows a special feature of the BONE ontology,

which consists in the fact that many entities, usually described by means of relationships in many other ontologies, are described as instances of classes in the BONE ontology. This feature compensates the lack of attributes of the RDF predicates. For example, the "BONE:participation" class is used in the Open Archive to formulate statements like "someone took part in several events, and each event has its start time and end time". For the same reasons such classes as "BONE:dating", "BONE:naming", "BONE:authorship" are used in the BONE ontology instead of predicates such as "has-author", "has-date", "has-name", etc.

The second ontology of interest is shown in Figure 2. This is the Portal Ontology which is a part of the AKT Reference Ontology comprising 157 classes [22]. This ontology is used for description of bibliographic datasets of the LOD Cloud such as DBLP, SiteSeer, ACM, etc. We intend to use these datasets as a source of additional data for the Open Archive. Therefore, we are interested in comparing the two ontologies.

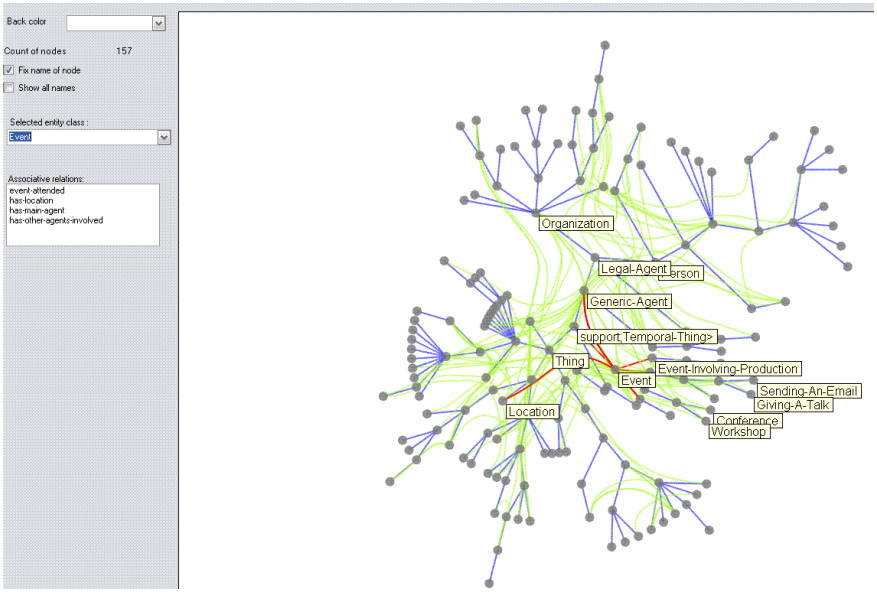

Figure 2. Links of the "AKT-Event" class

It should be noted that our goal is to establish links between them rather than to merge them. Unfortunately, such a known tool of ontology alignment as AgreementMaker [6] appeared to be helpless due to large differences in structure and vocabulary. The only obvious mapping exists between the BONE:person and AKT:Person classes. Other links are much less straightforward. Let us consider the "BONE:participation" class. As it is possible to see in Figure 1, this class is connected by the BONE:participant links to the "BONE:person" class and by the "BONE:in-org" links to the "BONE:org-

sys"class. The "BONE:participation" class is used for description of instances, corresponding either to the facts of person's participation in various events such as scientific symposia or to the facts of person's affiliation to some organizations. But in the AKT Reference ontology the same facts can be represented in many ways. It can be "AKT:works-for" links between the "AKT:Employee" and "AKT:Organization" classes, "AKT:has-affiliation" links between the "AKT:Person" and "AKT:Organization" classes, "AKT:has-main-agent", "AKT:has-other-agents-involved" between the "AKT:Event" class and the "AKT:Generic-Agent" class.

In all these cases we have to systematically establish correspondence between different groups of classes and relations of these two ontologies. More precisely, we have to establish correspondence between one or several groups of the form "Class 1-relation1- Class2" of the AKT Reference Ontology and one or several groups of the form "Class3-relation2-Class4-relation3-Class5" of the BONE ontology. In particular, we have to generate a new instance of the Class4 for every triple <Class1:instance1, relation 1, Class2:instance2>. The problem is complicated by the lack of lexical similarity between the identifiers of the two groups. From our point of view, lexical similarity can be essentially increased by modification of the BONE ontology identifiers. A large group of class and relation identifiers should be changed to make them more mnemonic and comprehensible.

As for the structural ontology difference, the problem of translation between the two ontologies can be solved by an appropriate SPARQL-query. A simplified example of a SPARQL query that generates triples of the BONE ontology corresponding to the triples of the AKT Reference Ontology looks as follows:

```
PREFIX: iis:<http://iis.nsk.su#>
PREFIX:akt:http://www.aktors.org/ontology/portal#
PREFIX:akts:<http://www.aktors.org/ontology/support#>
CONSTRUCT {
?p a iis:Class4.
?p rdfs:label ?Label.
?p iis:relation2 ?instance1.
?p iis:relation3 ?instance2.
}
WHERE {
      ?akt:instance1 akt:relation1 ?instance2
      ?instance1 a akt:Class1.
      ?instance2 a akt:Class2.
      ?instance1 akt:label1 ?instance1_label.
      ?instance2 akts:label2 ?instance2_label.
      BIND(Concat( str(?instance1_label), str (?instance2_label2)) As ?Label))
}
```

A tool allowing for generation of this kind of SPARQL-query on the basis of visualization of the two ontologies is currently under development.

One more example concerns the identity resolution problem. Let us consider an instance of the "BONE:person" class related to a former director of the IIS SB RAS, Vadim Yevgenievich Kotov:

```
<person rdf:about="piu\_200809052136"
    <name xml:lang="ru">  </name
    <name xml:lang="en">Kotov, Vadim Yevgenievich</name>
    <from-date>1938-07-23</from-date
    <sex>m</sex>
  </person
```

Since there is no information about his research papers in the Open Archive, we can look for this information in the DBLP[23] dataset structured with the AKT Reference Ontology. It is possible to find there a record like this:

```
            <akt:Person
rdf:about="http://dblp.rkbexplorer.com/id/people-
d32852eb011dfc13e96887308c2f2ca7-
4762c18c4afe3010e9da3d90f94113fb">
              <akt:full-name>Vadim E. Kotov</akt:full-name
```

Even if the "AKT:Person" class along with its property "AKT:full-name" is matched against the "BONE:person" class along with its property "AKT:full-name", it is not easy to understand, not being a specialist, that the "BONE: Kotov, Vadim Yevgenievich" object and the "DBLP: Vadim E. Kotov" object is the same person. Of course, a tool like SILK[10] using various string similarity metrics can help in generation of the "owl:sameAs" links. (This possibility is now under investigation. Anyway, these tools do not solve the problem of homonims). However, this problem can be essentially simplified by modification and extension of naming agreements in the context of the Open archive data sets.

To include the publications by Vadim E. Kotov into the content of the Open Archive, more sophisticated transformations are needed. First, we should create an instance of the "BONE: document" class for each individual of the "AKT:publication-reference" class; then for each "AKT:has-author" relationship it is necessary to generate an instance of the "BONE: authorship" class along with the "BONE: adoc" and "BONE: author" relationships, linking the instances of the "BONE: authorship" class with relevant instances of the "BONE: person" and "BONE: document" classes. All these transformations can be carried out with a SPARQL-query similar to the described above.

2. Visualization of SPARQL-queries for analysis of data sets

The main tools for investigation of a semantic system's content are creation of application-specific SPARQL-queries and query results visualization. This visualization can be generated by either a standard or specialized visualization algorithm. A window for SPARQL-query input is shown in Figure 4. It consists of three panels. The left panel shows a list of the main classes and relations of the semantic system under investigation, the top right panel is used for SPARQL-query input. In this panel a SPARQL-query is displayed. This query generates a graph whose nodes are persons and edges are "colleague" relations between these persons. The "colleague" relation corresponds to the fact that people are affiliated to the same organization or take part in the same event Note that SPARQL-queries are closely related to the RDF file structure. SPARQL inquires RDF graphs and RDF graph is a set of triples or "statements". Each triple has three parts: a subject, a predicate and an object. Each predicate is described in a corresponding ontology by means of the objectProperty clause. This is the reason why ontology visualization is so helpful at the stage of creation of a SPARQL-query.

The bottom right panel displays the query results in a text form. There is an "Execute" button and an "Execute clustering" button in the top right corner of the visualization panel. The first button starts generation of the query results as a graph, and the second one creates clustering of the resulting graph.

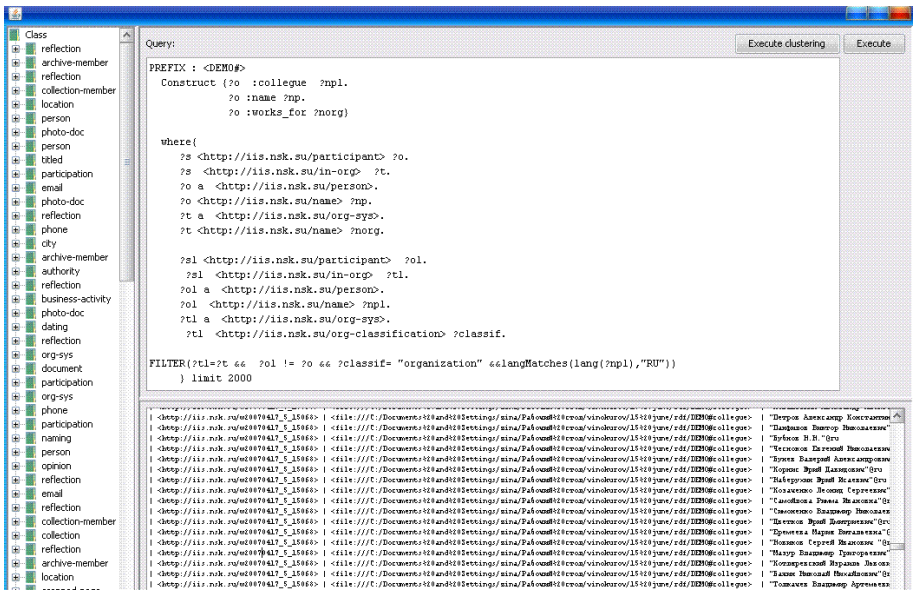

Figure 3. The window for SPARQL-query input

The query result in the graph form is shown in Figure 4. A graph consists of several connected components. People are grouped around the organizations in which they worked previously or are working now. As we do not use filtration by date, some people are assigned to several organizations. There is a high-density component in the center of the drawing. It corresponds to teachers of Novosibirsk State University. Therefore they are colleagues to other people from the institutes of the SB RAS.

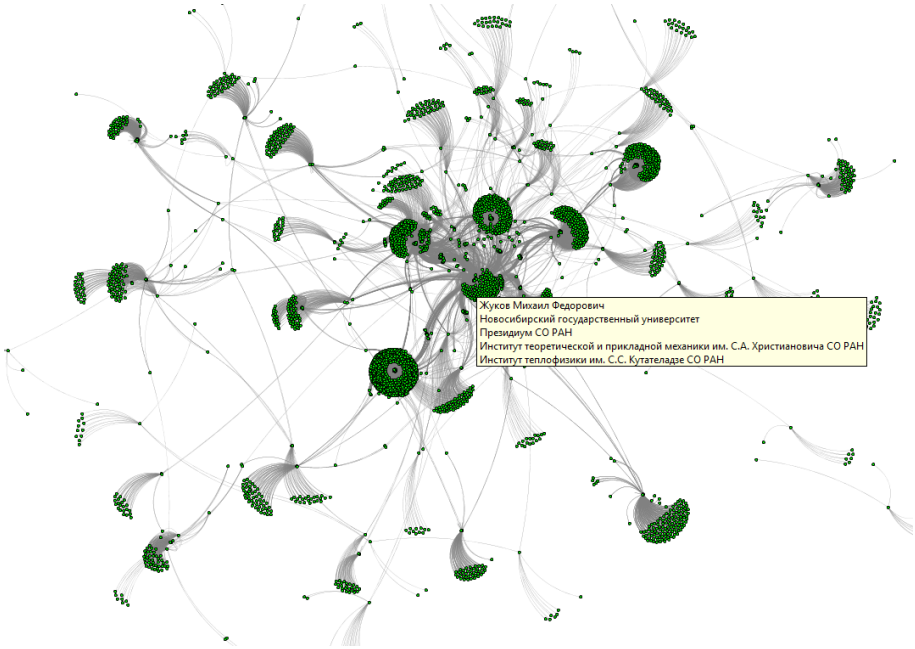

Figure 4. A "colleagues" graph generated by the SPARQL query from Figure 5

After a number of experiments, we have found that the dataset of the Open Archive is rather complete and clustering is not quite important for it, since it is possible to extract any part of the corresponding graph by SPARQL-queries with appropriate attributes. The clustering algorithm is needed in the case where the structure of the resulting graph is not as obvious. This kind of dense graphs arises, for example, during extraction of a co-authorship network from the DBLP dataset [24] or a citation network from the CiteSeer [23] dataset.

A new multilevel version of our clustering algorithm [1, 4] is implemented for these datasets. It uses a modularity measure [13] and a multilevel refinement algorithm [14] for cluster identification. This new algorithm considers each graph node as a separate cluster and merges two clusters if their merging increases the modularity measure. The Kernighan-Lin heuristics [12] is used for iterative refinement of the clustering. An example of the clustering algorithm applied to a co-authorship network extracted from the DBLP

dataset is shown in Figure 5. This co-authorship network is created by the following SPARQL-query:

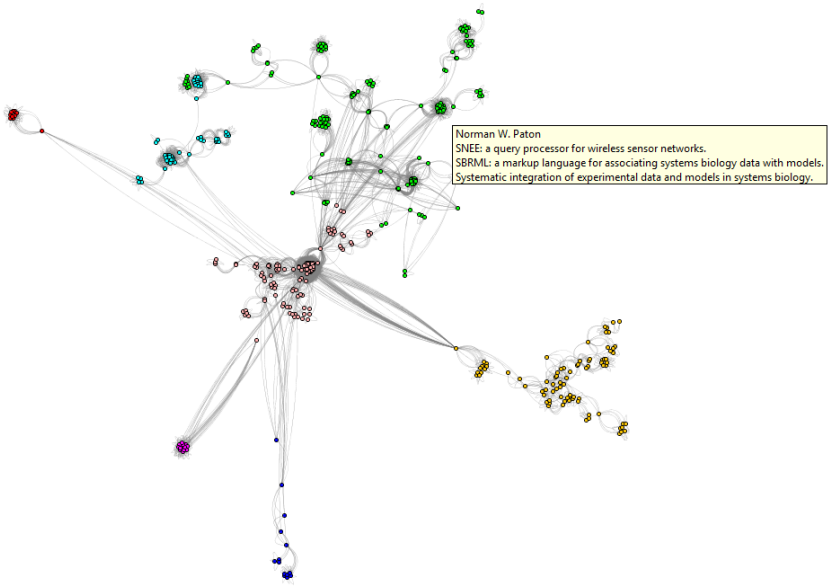

Figure 5. A co-authorship network clustering

```
PREFIX akt <http://www.aktors.org/ontology/portal#>
CONSTRUCT{?author1 :co_author ?author2.
          ?author1 :_label1 ?author_name1.
          ?author2 :_label1 ?author_name2.
          ?author1 :_label2 ?title.
          ?author2 :_label2 ?title.
   }
WHERE{
          ?article a akt:Article-Reference.
          ?article akt:has-author ?author1.
          ?article akt:has-author ?author2.
          ?article akt:has-title ?title.
          ?author1 akt:full-name ?author_name1.
          ?author2 akt:full-name ?author_name2.
FILTER(?author1 != ?author2)
} LIMIT 1000000
```

Figure 5 shows the greatest connected component of the co-authorship network. It is clustered and laid out with the newly implemented clustering

algorithm. As a result of this procedure, 7 clusters were identified. The greatest cluster comprises 200 authors. Typically each cluster is formed around an author with the maximum number of publications in the cluster. To make the found clusters easily recognizable for a user, they are in different colors.

Conclusion

First experiments with a toolkit for visual analysis of different ontology-based semantic systems at the stage of studying the integration possibilities of these systems have been described. The structure of the BONE ontology has been compared to that of the AKT Reference Ontology and one methodological source of their structural difference has been identified. An example of a SPARQL-query establishing correspondence between groups of classes and relations of the two ontologies is presented.

The experiments have also shown that the future research should be conducted in two directions: (i) further development of the toolkit, (ii) modification of the lexical structure of the BONE ontology and the Open Archive datasets.

At last, we would like to outline the difference between our toolkit and the RDF Gravity program [25]. We have to remind that the topic of this paper is not only ontology visualization, but mainly the problem of ontology based semantic systems integration. In this context, the RDF Gravity program is not applicable, because it uses the RDQL query language [http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/] which does not have the needed clauses. In particular, there is no "Construct" clause and many other clauses used by us. In addition, our set of visualization algorithms is significantly richer since the RDF Gravity program "uses the layout algorithms directly supported by the Jung API" [25].

References

[1] Apanovich Z.V., Vinokurov P.S. An extension of a visualization component of ontology based portals with visual analytics facilities // Bull. Novosibirsk Comp. Center. Ser. Computer Science. – Novosibirsk, 2010. – IIS Special Iss. 31. – P. 17–28.

[2] Alani H. TGVizTab: An Ontology Visualization Extension for Protege // Proc. of Knowledge Capture (K-Cap'03),Workshop on Visualization Information in Knowledge Engineering, Sanibel Island, Florida, USA. 2003.

[3] Marchuk A.G., Marchuk P.A. Specific features of digital libraries construction with linked content // Proc. of the RCDL'2010 Conference, 2010. – P. 19–23.

[4] Apanovich Z.V., Vinokurov P.S. Ontology based portals and visual analysis of scientific communities // First Russia and Pacific Conf. on Computer Tech-

nology and Applications, 6–9 September, 2010, Vladivostok, Russia. – 2010. – P. 7–11.

[5] Bizer C., Heath, T. and Berners-Lee, T. Linked Data – The Story So Far // Int. J. Semantic Web Inf. Syst. – 2009. – Vol. 5 (3). – P. 1–22.

[6] Cruz I. F., Stroe C., Caimi F., Fabiani A., Pesquita C., Couto F. M., Palmonari M. Using AgreementMaker to Align Ontologies for OAEI 2011. – http://ceur-ws.org/Vol-814/oaei11_paper1.pdf

[7] Erling O. How Virtuoso uses Relational Technology in its RDF Triple Store and SPARQL implementation
http://virtuoso.openlinksw.com/whitepapers/SPARQL%20RDF%20Store%20using%20SQL-ORDBMS.html

[8] Fruchterman T. M. J., Reingold E. M. Graph Drawing by Force-Directed Placement // Software – Practice and Experience. – 1991. – Vol. 21, N 11. – P. 1129–1164.

[9] Holten D. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data // Transactions on Visualization and Computer Graphics. – 2006. – Vol. 12, N 5. – P. 741–748.

[10] Isele R., Jentzsch A., Bizer Ch. Silk Server – Adding missing Links while consuming Linked Data // 1st Internat. Workshop on Consuming Linked Data (COLD 2010), Shanghai, November 2010.

[11] Katifori A., Halatsis C., Lepouras G., Vassilakis C., Giannopoulou E. Ontology Visualization Methods – a Survey // ACM Computing Surveys. – 2007. – Vol. 39(4).

[12] Kernighan B., Lin S. An efficient heuristic procedure for partitioning graphs // Bell System Technical Journal. – 1970. – Vol. 49. – P. 291–307.

[13] Newman M. E. J., Girvan M. Finding and evaluating community structure in networks // Physical Review E, 69.26113. – 2004.

[14] Noack A. , Rotta R.Multi-Level Algorithms for Modularity Clustering // SEA. – 2009. – P. 257–268.

[15] Ngomo A.-C. N., Auer S. LIMES – A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data // IJCAI 2011: Proc. of the 22nd Internat. Joint Conf. on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011. – P. 2312–2317.

[16] Oren E., Delbru R., Catasta M., Cyganiak R., Stenzhorn H. Tummarello G. Sindice.com: a document-oriented lookup index for open linked data // Int. J. Metadata, Semantics and Ontologies. – 2008. – Vol. 3, N 1. – P. 37–52.

[17] Pietriga E. IsaViz. – http://www.w3.org/2001/11/IsaViz.

[18] Sintek M. Ontoviz tab: Visualizing Protégé ontologies. – 2003. – http://protegewiki.stanford.edu/wiki/OntoViz.

[19] Storey M.-A. D. , Muller H. A. Manipulating and documenting software structures using shrimp views // Proc. of the Intl. Conf. on Software Mainten. – 1995.

[20] Schultz A. et al. How to integrate LINKED DATA into your application // Semantic technology & Business Conference, San Francisco, June 5, 2012. – http://mes-semantics.com/wp-content/uploads/2012/09/Becker-etal-LDIF SemTechSanFrancisco.pdf.

[21] Tramp S., Williams H., Eck K. Creating Knowledge out of Interlinked Data: The LOD2 Tool Stack. – http://lod2.eu/Event/ESWC2012-Tutorial.html.

[22] AKT ontology description. – http://www.aktors.org/ontology.

[23] CiteSeer dataset. – http://citeseer.rkbexplorer.com/.

[24] DBLP dataset. – http://dblp.rkbexplorer.com/.

[25] RDF Gravity. – http://semweb.salzburgresearch.at/apps/rdf-gravity/index.html.