ALGORITHM FOR QUICK NUMBERING OF LARGE VOLUMES OF DATA

Krassimira Ivanova

Abstract*: An original algorithm for numbering large datasets by means of Natural Language Addressing (NLA) is presented in the paper. We use a counter to number different instances and store its current value in the container NL-addressed by the instance. If the instance is repeated, from this NL-address we receive its already assigned number. The algorithm is implemented in an experimental program RDFArM for storing large RDF-datasets. The provided experiments have shown that NL-access time for one instance (triple or quadruple) does not depend on number of already stored instances from the dataset. This is very important for storing Big Data.*

Keywords*: Natural Language Addressing, Big Data, Numbering Large Datasets*

ACM Classification Keywords*: H.2 Database Management; H.2.8 Database Applications*

Introduction

There are three main problems of so called "Big Data" connected to its volume, variety, and velocity [Zikopoulos et al, 2012]:

- The sheer volume of data being stored today is exploding. It is no longer unheard of for individual enterprises to have storage clusters holding petabytes of data;
- Enterprises must be able to analyze variety of all types of data, both relational and non-relational: text, sensor data, audio, video, transactional, etc.;
- The velocity at which data is generated has changed. In traditional processing, one can think of running queries against relatively static data. With streams computing, one can execute a process similar to a continuous query to get continuously updated results in real time.

Large unstructured or semi-structured datasets require a high level of computational sophistication because operations that are easy at a small scale - such as moving data between machines or in and out of storage, visualizing the data, or displaying results - can all require substantial algorithmic ingenuity [NRC, 2013].

Consider a real time stream of incoming strings of symbols with possible many repetitions of the same strings. Strings may have different size (number of symbols) which may be very long, for instance, several hundred symbols. To work with such strings, it is convenient to create a unique numeration of strings, i.e. the equal strings have to have equal numbers. It is important for the next generation of

information systems. The problem we have to solve is to propose an algorithm which may create unique numeration of a stream of strings in real time, i.e. without spending time for rebuilding the indexes and recompilation of information base. This paper presents such algorithm with constant complexity for quick numbering Big Data sets.

In the next point we will outline the main approach for solving the problem. Further we will present the algorithm as well as an example of its implementation. Analysis of the results and possible future research conclude the paper.

Natural Language Addressing (NLA)

"Natural Language Addressing" (NLA) is an approach for storing large semi-structured or unstructured datasets. The idea of NLA [Ivanova et al, 2012; Ivanova et al, 2012a; Ivanova et al, 2013a; Ivanova et al, 2013b; Ivanova et al, 2013c; Ivanova et al, 2013d; Ivanova et al, 2013e; Ivanova, 2013; Ivanova, 2014; Ivanova, 2014a] consists in using the computer encoding of symbols as logical address of connected to it information stored in a multi-dimensional numbered information spaces [Markov, 1984; Markov, 2004; Markov, 2004a]. This way no indexes are needed and high speed direct access is available. It is similar to the natural order addressing in a dictionary where no explicit index is used but the concept by itself locates the definition.

To illustrate NL-addressing in the multi-dimensional information spaces let see an example (Figure 1).

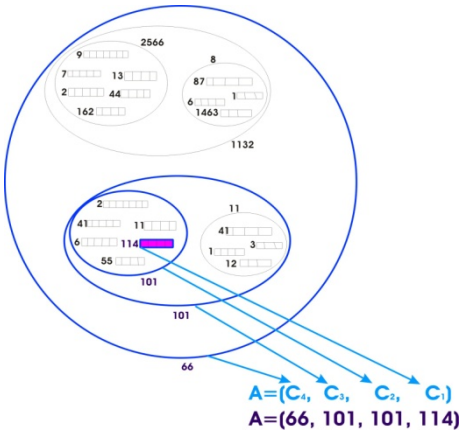

Figure 1. Example of a space address

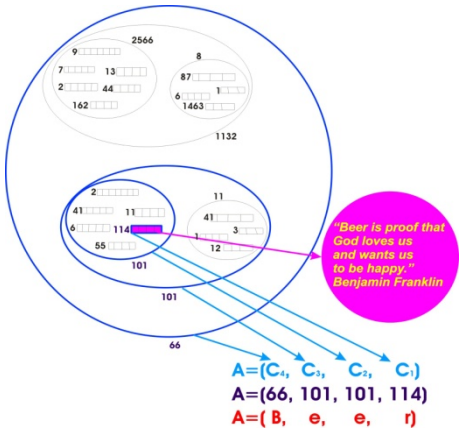

Figure 2. Example of Natural Language Space Address and content pointed by it

Our storing model is a hierarchy of hash tables. The leaves of the hierarchy are containers which may store basic information elements (BIE) which may be a letter, a word, a phrase of words, or, simply, string of symbols. Such container is colored in black on Figure 1 and its number is 114.

Let a set of BIE is numbered and stored in any archive (file). If we have several sets of BIE, we may number them again and store in a common archive, and so on. This way we receive a specific hierarchy

of numbered sets. If one will write the sequence of numbers of the sets starting from the one which contain all others, he will create a space address. The space address on Figure 1 means that the space with number 66 contains the space with number 101. The numbering is unique for every set. Because of this, there is no problem to have the same numbers in the included sets what is illustrated at the Figure 1 – set 101 contains element with number 101 which is a set. Finally, the last set contains element with number 114 which is not a set but container with a string of symbols.

In other words, the space address of this string is A = (66, 101, 101, 114) and its content may be written or read directly using this address.

Now we may illustrate the idea of NL-addressing.

Consider the space address we just have seen – (66, 101, 101, 114).

If we assume these numbers as ASCII codes, i.e. 66 = B, 101 = e, 114 = r, we may "understand" the space address as the word "Beer" (Figure 2).

At the end, we have to illustrate the BIE (content) which may be stored at such NL-space address. It may be arbitrary long string of words. In our example we choose the BIE to be the remarkable aphorism of Benjamin Franklin: "Beer is proof that God loves us and wants us to be happy" (Figure 2).

In this case (Figure 2) the couple {(space address A), (BIE)} is:

{(B, e, e, r), ("Beer is proof that God loves us and wants us to be happy." Benjamin Franklin)}

To access the text, we have to convert index (B, e, e, r) to index (66, 101, 101, 114) and to use corresponded access operations, i.e. we have the consequence:

Beer =>

=> (B, e, e, r) =>

=> (66, 101, 101, 114) =>

=> ("Beer is proof that God loves us and wants us to be happy." Benjamin Franklin).

Algorithm for Quick Numbering by means of NLA

Consider the problem of numerating a sequence of arbitrary words including multiple repeating of words but the same words has to be numbered with the same numbers independently of quantity of its repeating. If the length of the sequence is not so great, we may use a binary tree to store every word and its number. To find if it is already numbered we need to provide binary search and if the word is already indexed than to assign the same number.

The challenge of "Big Data" is that the sequence may be with unlimited length (several trillions) and in the same time words may come permanently during the time. In such case the static search trees could

not be used because after every new word we have to reconfigure the tree. We need a new approach to solve this problem. For this purpose we may use Natural Language Addressing (NLA).

The algorithm for numbering by NLA is very simple – we need to have a counter which will count every new different word and to store its current value in the container NL-addressed by the new word. When the same word is repeated from this NL-address we will receive its already assigned number.

The ***algorithm for assigning unique numbers*** is as follow:

BEGIN

counter := 1; // a counter is used, it starts from 1

input (string); // string to be numbered

STRNumber := NLA-read (string); // from NL-archive, using the incoming string as path, obtain its number

IF STRNumber = 0 // if no number exists in the container

THAN begin

STRNumber := counter; // assign counter value as number of string

NLA-write (string) := STRNumber; // store number in the container located by the string as path

INC (counter); // increment the counter by 1

end

END.

The algorithm is illustrated on Figure 3.

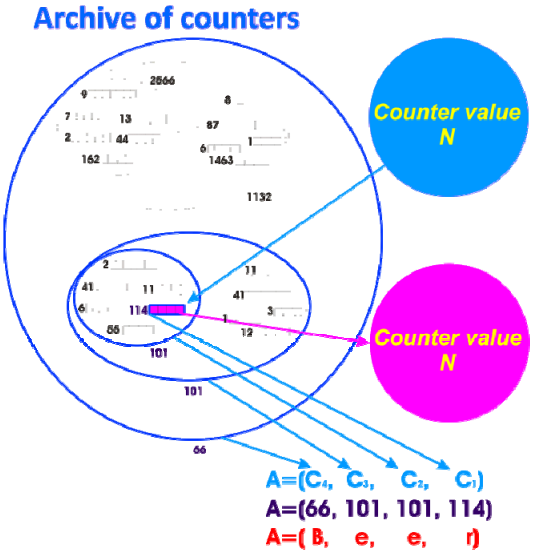

Figure 3. Illustration of the algorithm for assigning unique numbers

Experimental implementation of algorithm

Now, our next step is to implement our algorithm for solving real practical problem. Such very actual problem is storing Resource Description Framework (RDF) datasets. RDF is approach for representing Big Data. The primary goal of RDF is to handle large non regular or semi-structured data [Muys, 2007]. RDF provides a general method to decompose any information into pieces called triples [Briggs, 2012]:

- Each triple is of the form <"Subject", "Predicate", "Object">;
- Subject and Object are the names for two things in the world. Predicate is the relationship between them;
- Subject, Predicate, Object may be given as URI's (stand-ins for things in the real world);
- Object can additionally be raw text.

The power of RDF relies on the flexibility in representing arbitrary structure without a priori schemas. Each edge in the graph is a single fact, a single statement, similar to the relationship between a single cell in a relational table and its row's primary key. RDF offers the ability to specify concepts and link them together into a graph of data [Faye et al, 2012].

The experimental implementation of proposed algorithm is illustrated on Figure 4. The main idea is to use NL-addressing for quick unique numbering of elements of triples/quadruples and after that to use these numbers as co-ordinates for storing information in the archives. In this case we have two kinds of archives: (1) archives of counters and (2) archives of values. The storing model we used is multi-layer [Ivanova et al, 2013b]. Subjects, Predicates, Objects, and Contexts are numbered separately and these numbers are used to construct storing co-ordinates. We assume that the triple datasets contain empty context which has to be omitted.

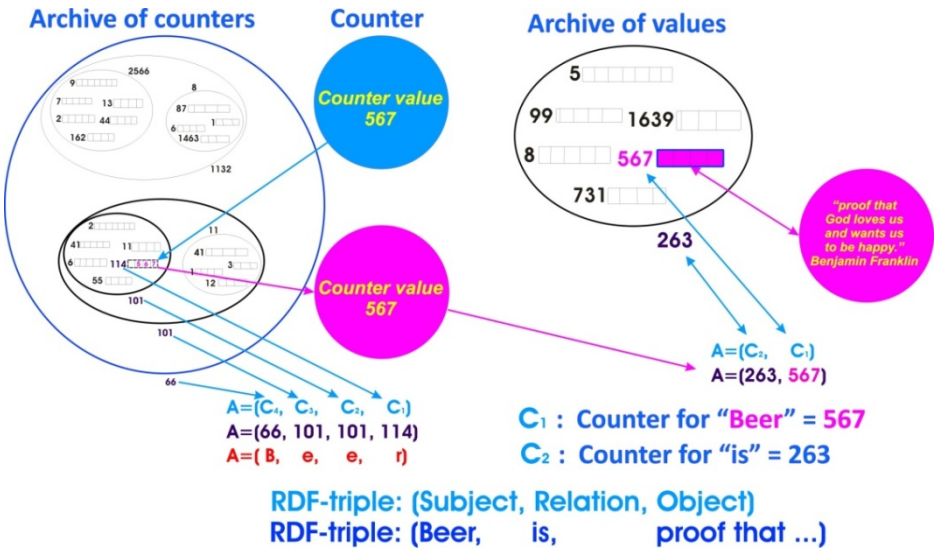

Figure 4. Illustration of the experimental algorithm for storing RDF triples

In Figure 4 we illustrate storing of RDF-triple (beer, is, proof that...).

First we assign a number to subject: "beer".

The same we do for the relation: "is".

And after that we used these numbers as coordinates of the container where the object "proof that ..." has to be stored.

Experimental program RDFArM

We have realized experimental program RDFArM for storing middle-size and large RDF-datasets. Experiments were provided with both real and artificial datasets. Experimental results were systematized in [Ivanova, 2014]. We have compared RDFArM with well-known RDF-stores: OpenLink Virtuoso Open-Source Edition 5.0.2 [Virtuoso, 2013], Jena SDB Beta 1 on Postgre (SQL 8.2.5 and MySQL 5.0.45) [Jena, 2013], and Sesame 2.0 [Sesame, 2012], all tested by Berlin SPARQL Bench Mark (BSBM) team and connected to it research groups [Becker, 2008; BSBM, 2012; BSBMv2, 2008; BSBMv3, 2009].

We have provided experiments with:

Middle-size RDF-datasets based on selected real datasets from DBpedia [DBpedia, 2007a; DBpedia, 2007b] and artificial datasets created by BSBM Data Generator [Bizer & Schultz, 2008; Bizer & Schultz, 2009; BSBM DG, 2013].

The large RDF-datasets, based on selected real datasets from DBpedia's homepages [DBpedia, 2007c], geocoordinates datasets [Becker, 2008], and Billion Triple Challenge (BTC) 2012 [BTC, 2012; datahub_data0, 2012]. The artificial large RDF-datasets were generated by BSBM Data Generator [BSBM DG, 2013] and published in Turtle format [BSBMv1, 2008; BSBMv2, 2008; BSBMv3, 2009; BSBMv5, 2009; BSBMv6, 2011], i.e. in N-quads [N-Quads, 2013]. We converted it to N-triple format using "rdf2rdf" program developed by Enrico Minack [Minack, 2010].

We have used the Friedman and Nemenyi tests to detect statistically significant differences between the systems [Friedman, 1940]. The Friedman test is a non-parametric test, based on the ranking of the systems on each dataset. It is equivalent of the repeated-measures ANOVA [Fisher, 1973]. We have used Average Ranks ranking method, which is a simple ranking method, inspired by Friedman's statistic [Neave & Worthington, 1992]. The null-hypothesis states that if all the systems are equivalent than their ranks R_j should be equal. In our case, the null-hypothesis was rejected; we could proceed with the Nemenyi test [Nemenyi, 1963] which is used when all systems are compared to each other. The performance of two systems is significantly different if the corresponding average ranks differ by at least the critical difference. The Nemenyi test results for tested systems (Figure 5) had shown that RDFArM is

at critical distances to Jena and Sesame. RDFArM is nearer to Jena than to Sesame. RDFArM, Jena, and Sesame are significantly different from Virtuoso [Ivanova, 2014].

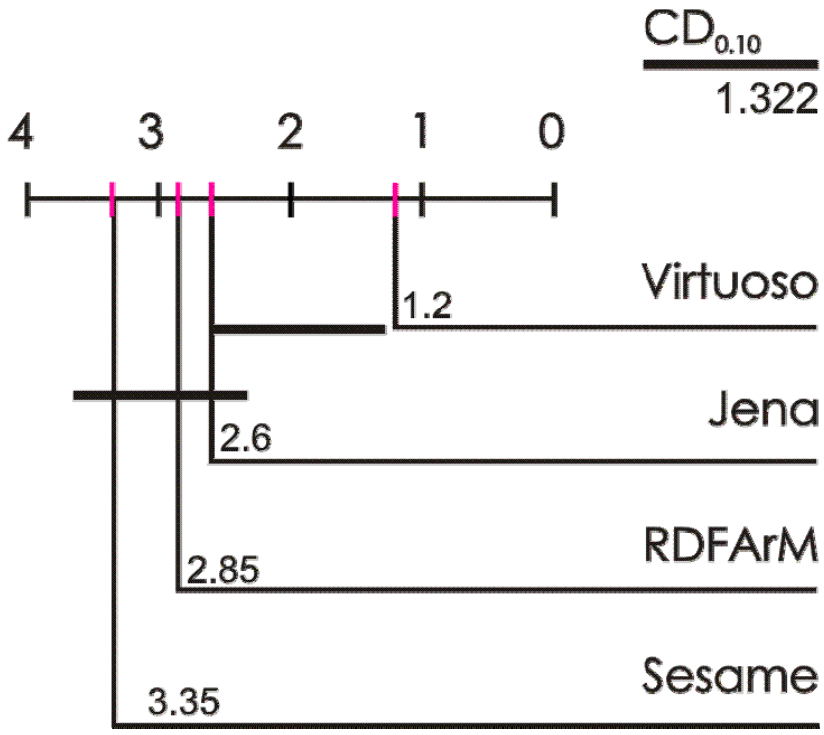

Figure 5. Visualisation of Nemenyi test results

All experiments had shown constant time for storing of one triple independently of the number of already stored ones. It is illustrated on Figure 6 by comparison of graphics of log(n) (black line) and average time in ms for storing one triple from BSBM 100M (gray line).

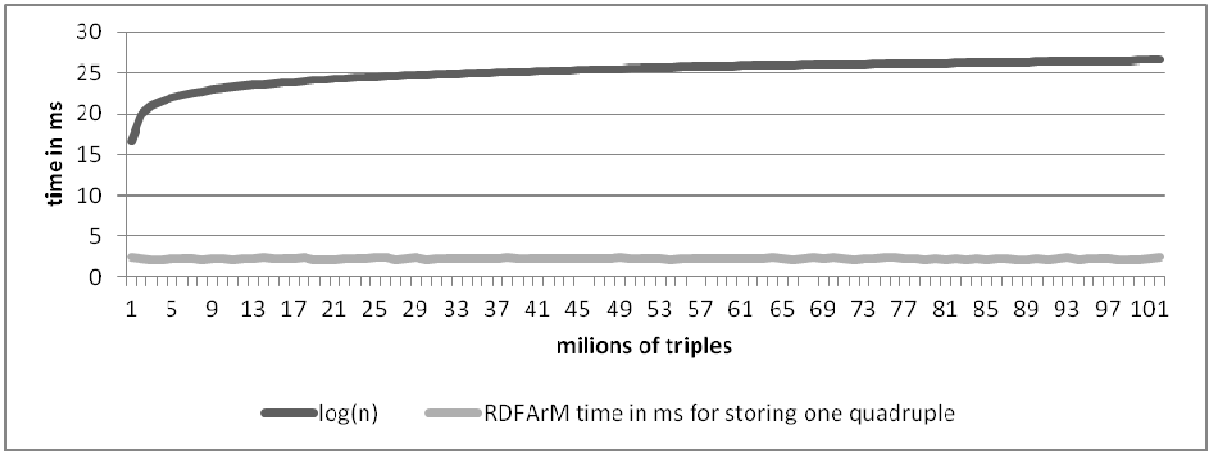

Figure 6. Comparison of log(n) and average time for storing one triple from BSBM 100M

Conclusion

An original algorithm for numbering by means of Natural Language Addressing has been presented in the paper. It is very simple: we have a counter, which count every new different word, and store its current value in the container NL-addressed by the new word. When the same word is repeated, from this NL-address we receive its already assigned number if such exists. In other case we assign the current value of counter to the new word.

We have realized experimental program RDFArM for storing middle-size and large RDF-datasets. We provided series of experiments to estimate the storing time. Experiments were provided with both real and artificial datasets. The main conclusion is optimistic because RDFArM is at critical distances to Jena and Sesame.

NL-access time for one instance (triple or quadruple) does not depend on number of already stored instances from the dataset. This is very important for storing Big Data.

Bibliography

[Becker, 2008] Christian Becker, "RDF Store Benchmarks with Dbpedia", Freie Universität Berlin, 2008, http://wifo5-03.informatik.uni-mannheim.de/benchmarks-200801/ (accessed: 05.04.2013)

[Bizer & Schultz, 2008] Christian Bizer, Andreas Schultz: Benchmarking the Performance of Storage Systems that expose SPARQL Endpoints; In: Proc. of the 4th International Workshop on Scalable Semantic Web knowledge Base Systems (SSWS2008), http://www4.wiwiss.fu-berlin.de/bizer/pub/BizerSchulz-BerlinSPARQLBenchmark.pdf (accessed: 31.07.2013)

[Bizer & Schultz, 2009] Christian Bizer, Andreas Schultz, "The Berlin SPARQL Benchmark", In: International Journal on Semantic Web & Information Systems, Vol. 5, Issue 2, Pages 1-24, 2009, http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/Bizer-Schultz-Berlin-SPARQL-Benchmark-IJSWIS.pdf; see also http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/ (accessed: 31.07.2013)

[Briggs, 2012] Mario Briggs "DB2 NoSQL Graph Store", What, Why & Overview, A presentation, Information Management software IBM, 2012, https://www.ibm.com/developerworks/mydeveloperworks/blogs/nlp/resource/DB2_NoSQLGraphStore.pdf?lang=en (accessed: 01.12.2012)

[BSBM DG, 2013] Data Generator and Test Driver, In: Berlin SPARQL Benchmark (BSBM) - Benchmark Rules, http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/spec/BenchmarkRules/index.html#datagenerator (accessed: 31.07.2013)

[BSBM, 2012] Berlin SPARQL Benchmark, http://www4.wiwiss.fu-berlin.de/bizer/BerlinSPARQLBenchmark/ (accessed 09.04.13).

[BSBMv1, 2008] Berlin SPARQL Benchmark Results, V1, 2008, http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/V1/results/index.html (accessed: 31.07.2013)

[BSBMv2, 2008] Berlin SPARQL Benchmark Results, V2 2008, http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V2/index.html (accessed: 31.07.2013)

[BSBMv3, 2009] Berlin SPARQL Benchmark Results, V3, 2009, http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V3/index.html (accessed: 31.07.2013)

[BSBMv5, 2009] BSBM Results (V5) for Virtuoso, Jena TDB, BigOWLIM, 2009, http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V5/index.html (accessed: 31.07.2013)

[BSBMv6, 2011] Berlin SPARQL Benchmark Results, V6, 2011, http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V6/index.html (accessed: 31.07.2013)

[BTC, 2012] Billion Triple Challenge 2012 Dataset http://km.aifb.kit.edu/projects/btc-2012/ (accessed: 16.03.2013)

[datahub_data0, 2012] BTC data set from Datahub, http://km.aifb.kit.edu/projects/btc-2012/datahub/data-0.nq.gz (accessed: 16.03.2013).

[DBpedia, 2007a] DBpedia dataset "homepages.nt" dated 2007-08-30, http://wifo5-03.informatik.uni-mannheim.de/benchmarks-200801/homepages-fixed.nt.gz (accessed: 31.07.2013)

[DBpedia, 2007b] DBpedia dataset "geocoordinates.nt" dated 2007-08-30, http://wifo5-03.informatik.uni-mannheim.de/benchmarks-200801/geocoordinates-fixed.nt.gz (accessed: 31.07.2013)

[DBpedia, 2007c] DBpedia dataset "infoboxes.nt" dated 2007-08-30, http://wifo5-03.informatik.uni-mannheim.de/benchmarks-200801/infoboxes-fixed.nt.gz (accessed: 31.07.2013)

[Faye et al, 2012] David C. Faye, Olivier Cure, Guillaume Blin. A survey of RDF storage approaches. Received, December 12, 2011, Accepted, February 7, 2012, ARIMA Journal, vol. 15 (2012), pp. 11-35.

[Fisher, 1973] R. A. Fisher, "Statistical methods and scientific inference (3rd edition)", Hafner Press, New York, 1973, ISBN 978-002-844740-7

[Friedman, 1940] Friedman, M.: "A comparison of alternative tests of significance for the problem of m rankings", Annals of Mathematical Statistics, Vol. 11, 1940, pp.86-92.

[Ivanova et al, 2012] Krassimira Ivanova, Vitalii Velychko, Krassimir Markov. "About NL-addressing" (К вопросу о естествено-языконой адрессации) In: V. Velychko et al (ed.), Problems of Computer in Intellectualization. ITHEA® 2012, Kiev, Ukraine - Sofia, Bulgaria, ISBN: 978-954-16-0061 0 (printed), ISBN: 978-954-16-0062-7 (online), pp. 77-83 (in Russian).

[Ivanova et al, 2012a] Krassimira Ivanova, Vitalii Velychko, Krassimir Markov. "Storing RDF Graphs using NL-addressing", In: G. Setlak, M. Alexandrov, K. Markov (ed.), Artificial Intelligence Methods and Techniques for Business and Engineering Applications. ITHEA® 2012, Rzeszow, Poland; Sofia, Bulgaria, ISBN: 978-954-16-0057-3 (printed), ISBN: 978-954-16-0058 0 (online), pp. 84 – 98.

[Ivanova et al, 2013a] Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko, "Introduction to the Natural Language Addressing", International Journal "Information Technologies & Knowledge" Vol.7, Number 2, 2013, ISSN 1313-0455 (printed), 1313-048X (online), pp. 139–146.

[Ivanova et al, 2013b] Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko, "Introduction to Storing Graphs by NL-Addressing", International Journal "Information Theories and Applications", Vol. 20, Number 3, 2013, ISSN 1310-0513 (printed), 1313-0463 (online), pp. 263 – 284.

[Ivanova et al, 2013c] Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko, "Storing Dictionaries and Thesauruses Using NL-Addressing", International Journal "Information Models and Analyses" Vol.2, Number 3, 2013, ISSN 1314-6416 (printed), 1314-6432(online), pp. 239 - 251.

[Ivanova et al, 2013d] Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko, "The Natural Language Addressing Approach", International Scientific Conference "Modern Informatics: Problems, Achievements, and Prospects of Development", devoted to the 90th anniversary of academician V. M. Glushkov. Kiev, Ukraine, 2013, ISBN 978-966-02-6928-6, pp. 214 - 215.

[Ivanova et al, 2013e] Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko, "Storing Ontologies by NL-Addressing", IVth All–Russian Conference "Knowledge-Ontology-Theory" (KONT-13), Novosibirsk, Russia, 2013, ISSN 0568 661X, pp. 175 - 184.

[Ivanova, 2013] Krassimira Ivanova, "Informational and Information models", In Proceedings of 3rd International conference "Knowledge Management and Competitive Intelligence" in the frame of 17th International Forum of Young Scientists "Radio Electronics and Youth in the XXI Century", Kharkov National University of Radio Electronics (KNURE), Kharkov, Ukraine, Vol.9, 2013, pp. 6-7.

[Ivanova, 2014] Krassimira Ivanova, "RDFArM - A System for Storing Large Sets of RDF Triples and Quadruples by means of Natural Language Addressing", International Journal "Information Models and Analyses" Vol.3, Number 4, 2014, ISSN 1314-6416 (printed), 1314-6432(online), pp. 303 - 322.

[Ivanova, 2014a] Krasimira Ivanova, "Storing Data using Natural Language Addressing", PhD Thesis, Hasselt University, Belgium, 2014

[Jena, 2013] Apache Jena, http://jena.apache.org/about_jena/about.html (accessed: 23.03.2013)

[Markov, 1984] Krassimir Markov, "A Multi-domain Access Method", Proceedings of the International Conference on Computer Based Scientific Research, PLOVDIV, 1984, pp. 558 - 563.

[Markov, 2004] Krassimir Markov, "Multi-domain information model", Int. J. Information Theories and Applications, 11/4, 2004, pp. 303 - 308

[Markov, 2004a] Krassimir Markov, "Co-ordinate based physical organization for computer representation of information spaces", (Координатно базирана физическа организация за компютърно представяне на информационни пространства) Proceedings of the Second International Conference "Information Research, Applications and Education" i.TECH 2004, Varna, Bulgaria, Sofia, FOI-COMMERCE – 2004, стр. 163 - 172 (in Bulgarian).

[Minack, 2010] Enrico Minack, "RDF2RDF converter", http://www.l3s.de/~minack/rdf2rdf/ 2010, (accessed: 31.07.2013).

[Muys, 2007] Andrae Muys, "Building an Enterprise Scale Database for RDF Data", Seminar, Netymon, 2007.

[Neave & Worthington, 1992] Neave, H., Worthington, P., "Distribution Free Tests", Routledge, 1992.

[Nemenyi, 1963] Peter Nemenyi, "Distribution-free multiple comparisons Unpublished", PhD thesis; Princeton University Princeton, NJ, 1963

[N-Quads, 2013] N-Quads: Extending N-Triples with Context http://sw.deri.org/2008/07/n-quads/ (accessed: 16.03.2013).

[NRC, 2013] National Research Council, "The Mathematical Sciences in 2025", Washington, DC: The National Academies Press, USA, 2013. ISBN-13: 978-0-309-28457-8. http://www.nap.edu/catalog.php?record_id=15269.

[Sesame, 2012] Sesame, OpenRDF, http://www.openrdf.org/index.jsp http://www.openrdf.org/doc/sesame2/2.3.2/users/userguide.html#chapter-sesame2-whats-new (accessed: 01.12.2012)

[Virtuoso, 2013] OpenLink Virtuoso Universal Server: Documentation http://docs.openlinksw.com/pdf/virtdocs.pdf, http://virtuoso.openlinksw.com/ (accessed: 23.03.2013)

[Zikopoulos et al, 2012] P.C. Zikopoulos, C. Eaton, D. deRoos, T. Deutsch, G. Lapis, "Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data", McGraw-Hill Companies, USA, 2012, ISBN 978-0-07-179053-6, 166 p.

Authors' Information

Krassimira Ivanova *– University of National and World Economy, Sofia, Bulgaria; e-mail: krasy78@mail.bg*

Major Fields of Scientific Research: Software Engineering, Business Informatics, Data Mining, Multidimensional multi-layer data structures in self-structured systems