
ONTOARM - A SYSTEM FOR STORING ONTOLOGIES BY NATURAL LANGUAGE ADDRESSING

Krassimira Ivanova

Abstract: *In this paper we present results from experiments for storing RDF ontologies by means of Natural Language Addressing. For experiments we have realized system OntoArM aimed to store RDF triples in multi-layer hash tables (information spaces with variable size). The main features of system OntoArM are outlined in the paper. Analysis of the experimental results concluded the work.*

Keywords: *Natural Language Addressing; RDF ontologies*

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

Introduction

The word ontology can be used and has been used with very different meanings attached to it. Ironically, the ontology field suffered a lot from ambiguity. The Knowledge Engineering Community borrowed the term "Ontology" from the name of a branch of philosophy and converted into an object: "ontology". In the mid-90s philosophers "took it back" and began to clean the definitions that had been adopted [Gandon, 2002].

Formally, the ontology consists of: (1) Terms organized in taxonomy; (2) Definitions of terms and attributes; (3) Axioms and rules for inference.

The ontology formally can be described by the ordered triple $O = \langle X, R, F \rangle$ [Palagin & Yakovlev, 2005; Gavrilova, 2001; Palagin, 2006; Guarino, 1998], where X , R , F are a finite sets accordingly: X is a set of concepts (terms) from the subject area; R is a set of relationships between the elements of X ; F is a set of functions to interpret the X and/or R .

Classifications of ontologies from different points of view and based on different principles are given in many publications [Bashmakov, 2005; Dobrov et al, 2009].

Traditionally, ontologies are built by highly trained knowledge engineers with the assistance of domain specialists. It's time-consuming and laborious task. Ontology tools also require users to be trained knowledge representation and predicate logic.

There are several approaches for representing ontologies. An example of such approach is using of XML. It is a popular markup language of metadata. With the development of the XML, different definitions of metadata have been proposed such as Dublin Core [Weibel et al, 1998] and ebXML [ebxml, 2012].

However, from the viewpoint of ontology, XML is not suited to describe the interrelationships of resources [Gunther, 1998]. Therefore, W3C has suggested the "Resource Description Framework" (RDF). There are several ontology languages like XML, RDF schema RDF(S), DAML+OIL and OWL. Many ontology tools have been developed for implementing metadata of ontology using these languages [Hertel et al, 2009].

Operations with ontologies are functions of the so called "middleware". What is called middleware is the layer implementing the access to the physical ontology data store.

Besides an inference mechanism, the access layer should provide functions for creating, querying and deleting data in the store.

While adding data requires parsing and ideally a validation of the incoming ontology sentences, querying the ontology store needs the implementation of some kind of query language as well as an interpretation and a translation of this query language into calls to the physical storage.

Another important feature of this layer is the possibility to export ontology data to a file for exchange with other systems [Hertel et al, 2009].

The operations with several ontologies are needed when one application uses multiple ontologies, especially when using modular design of ontologies or when we need to integrate with systems that use other ontologies.

Tools for developing ontologies

The tools for developing ontologies allow users to define new concepts, relationships and instances, i.e. to create and/or expand existing ontologies. The ontology tools may contain some additional features such as graphical representation, information search and additional tuning [Noy & Musen, 2002]. Such tools are, for instance, SWOOP [Kalyanpur et al, 2005], Top Braid composer [TBC, 2012], Internet Business Logic [IBL, 2012], OntoTrack [Liebig & Noppens, 2003] and IHMC Cmap Ontology Editor [Hayes et al, 2005].

“Chimaera” helps with merging ontologies. It provides suggestions for subsumption, disjointness or instance relationship. These suggestions are generated heuristically and are provided for an operator, so that he may choose which one will be actually used [Chimaera, 2012]. “PROMT” (or “SMART”) system is a similar system that provides suggestions based on linguistic similarity, ontology structure and user actions. It points the user to possible effects of these changes [Promt, 2012].

In [OntoTools, 2012] more than 150 tools (ontology editors) are outlined. Many of the tools are closed systems. Therefore, it is not possible to evaluate the full functional capabilities. Thus, the choice of editor of ontologies for practical purposes depends of: Free distribution; Local use of the web interface; Extensibility of functional possibilities of the applications; Ability to include modules designed by the user. The basic features, capabilities, advantages, disadvantages, and comparative analysis of available onto-editors are given in a number of meaningful overviews [Ovdei & Proskudina, 2004; Calvanese et al, 2007; Filatov et al, 2007].

The instrumental systems for ontological engineering can be divided into three main groups [Ovdei & Proskudina, 2004]:

- The first group includes tools for creating ontologies that provide: Maintenance of collaborative development and review; Creation of ontologies according to any methodology; Maintenance of reasoning;
- The second group includes tools for: Unification of ontologies; Discovering semantic relations between different ontologies; Alignment the ontologies by establishing links between them and allowing the aligned ontologies to reuse information from one another [Noy & Musen, 1999];
- The third group includes tools for annotation of Web-based ontology resources.

General disadvantages of the instrumental systems are:

- Lack of automatic (or automated) procedures for forming components of ontologies;
- User interface based only on English, which does not permit using of other languages, such as Bulgarian, Russian, Greek, etc.;
- The structure of concepts may be built by only one type of relationships;

- For most commonly available ontological systems it is impossible to work with ontologies of large volume (e.g. OntoEdit free – up to 50 concepts);
- Many tools store the ontologies in text files, which limits the speed of access to ontologies;
- Some functions are not available in the free versions of the tools;
- User documentation is not good enough.

The above shortcomings of popular English language ontological tools exist in similar instruments from Russian segment, in particular, "Multi-layer ontology editor" [Artemieva & Reshtanenko, 2008], "OntoEditor+" [Nevzorova et al, 2004] and others.

Comparative analysis of the ontology tools shows that all systems use finished products for data storing, which are limited to text files or relational databases. Both approaches for storing do not meet specific structures of the ontologies. This necessitates the development of new tools for storing ontologies.

OntoArM

OntoArM is a system for storing ontologies through Natural Language Addressing. Let remember, that the idea of Natural Language Addressing (NLA) [Ivanova et al, 2012a; 2012b; Ivanova et al, 2013a; 2013b; 2013c; 2013d; 2013e; Ivanova, 2013; Ivanova, 2014] consists in using the computer encoding of name's (concept's) letters 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 to the text elements 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.

OntoArM is built over Natural Language Addressing Access Method and corresponded Archive Manager called **NL-ArM**, realized in [Ivanova, 2014]. OntoArM is aimed to store libraries of RDF ontologies in multi-layer hash tables (information spaces with variable size). Each ontological element can be stored by appropriate path, which is set by a natural language word or phrase. Below we will outline the main features of OntoArM.

The main idea for storing ontologies in OntoArM follows the idea of multi-layer graph representation [Ivanova et al, 2012b]: the ontology relations are assumed as layers and the ontology concepts are assumed as paths valid for all layers. The information about concepts as well information about the links of the concepts with other concepts is stored in the corresponded containers located by the path in the corresponded layers.

OntoArM has two modes of operation: *Automated* and *Manual*.

OntoArM automated mode functions

The OntoArM panel for working in automated mode is shown on Figures 1a and 1b. The main functions are Onto-Write and Onto-Read for which there are corresponded buttons.

By "**Onto-Write**" button the function for storing RDF ontology definitions from a file can be activated.

Ontologies are represented by sets of RDF triples. Each RDF triple (subject, relation and object) occupy one record in the input file. There is no limit to the number of records in the file. After pressing the "Onto-Write" button, the system reads records sequentially from the file and for each of them:

- Transform the subject (concept) into path;
- Store the object (definition and links) of the subject (concept) in the container located by the path in the file which corresponds to the layer given as relation in the triple.

The input file is in CSV file format. Its records have the next format: **<subject>;<relation>;<object><CR>**.

After storing the triples, in the panel near to the “Onto-Write” button, OntoArM displays two informative lines (Figure 1a):

- Total time used for storing all instances from the file;
- Average time used for storing of one instance, in milliseconds.

By “Onto-Read” button the function for reading objects (definitions) from the OntoArM archive can be activated. In the automated mode, Onto-Read uses as input a file with subjects (concepts) and relations (each couple on a separate line) and extract from corresponded layer their objects (definitions). If any object does not exist, the output is empty.

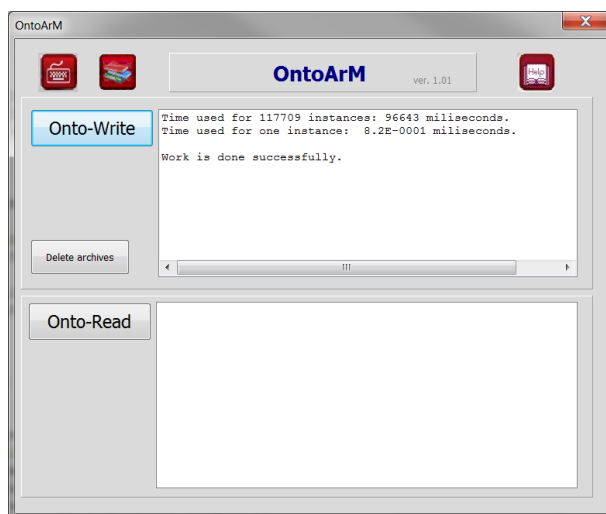
Each subject (concept), its relation and object (definition) occupy one record in the output file. There is no limit to the number of records in the file. After pressing the “Onto-Read” button, the system reads concepts sequentially from the input file and for each of them:

- Transform the subject (concept) into path;
- Extract the object connected to this concept from relation layer using the path to locate it.

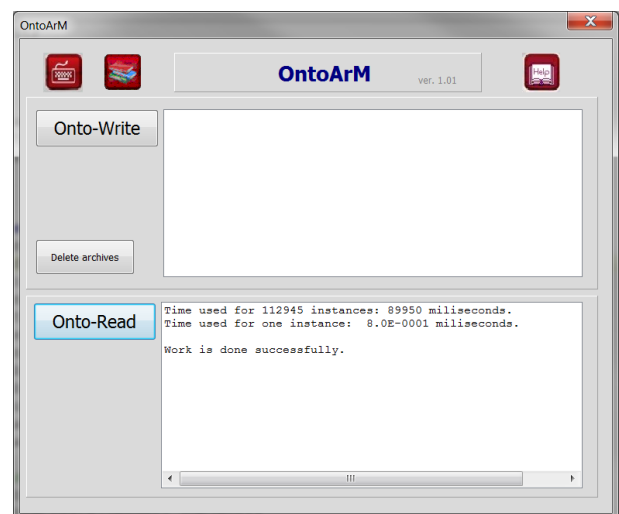
The output file is in CSV file format. Its records have the next format: <subject>;<relation>;<object><CR>.

In the panel next to the Onto-Read button, two informative lines are shown (Figure 1b):

- Total time used for extracting of all instances;
- Average time used for extracting of one instance, in milliseconds.






a. Onto-Write panel with informative lines



b. Onto-Read panel with informative lines

Figure 1. OntoArM panel for working in automated mode

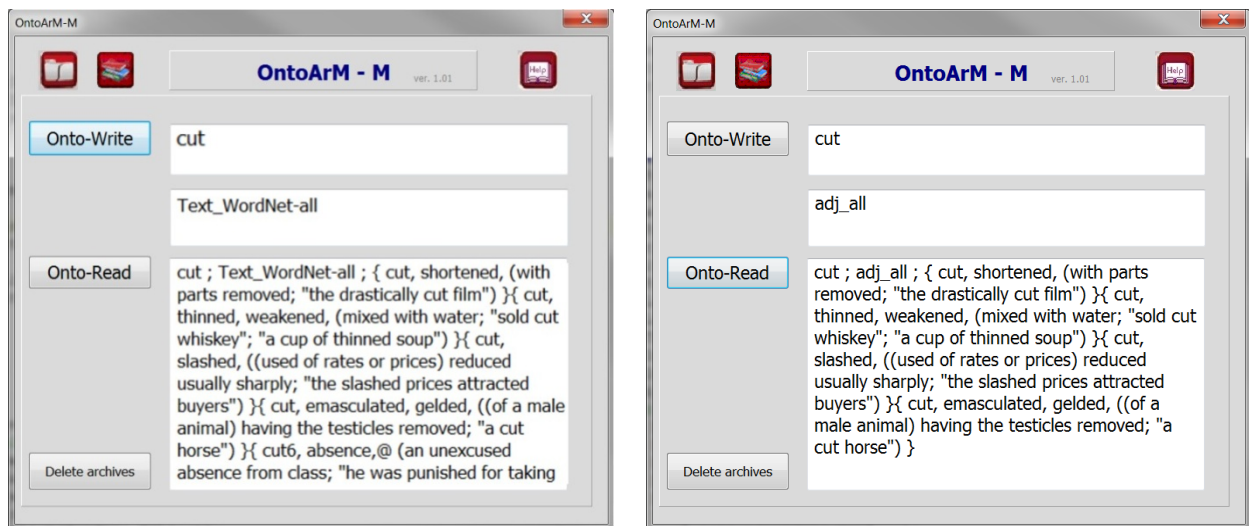
The form has three service buttons:

- The first () serves as a transition to the form for manual input and output of data to/from the system archive;
- The second () is connected to the module for adjusting the environment of the system – archives, input and output information, etc.;
- The third () activates the help text (user guide) of the system.

In the same panel, there is a button which enables deleting the work archives of the OntoArM (for test control in this version, they are stored on the hard disk but not in the computer main memory). OntoArM is completed with compressing program and after storing the information prepares small archive for long time storage.

OntoArM manual mode functions

The OntoArM panel for working in manual mode is shown on Figures 2a and 2b.



a. Manual input of the RDF-triple

b. Manual reading the RDF-triple

Figure 2. OntoArM panel for working in manual mode

By “**Onto-Write**” button (Figure 2a) the function for storing RDF-triples can be activated. Each subject (concept) and its relation and object (definition) can be given in corresponded fields on the screen form. After pressing the “Onto-Write” button, the system reads information from the fields and:

- Transform the subject (concept) into path;
- Store the object (definition) of this subject (concept) in the container located by the path in the layer pointed by the relation.

By “**Onto-Read**” button (Figure 2b) the function for reading RDF-objects (definitions) from the OntoArM archive can be activated. In the manual mode, Onto-Read uses as input the subject (concept) given in the screen field and extract from the archive its definition. If the definition does not exist, the output is empty definition.

There are two possibilities:

- Extract object from concrete layer given by corresponded relation;
- Extract from all layers the objects which correspond to a given subject.

After pressing the “Onto-Read” button, the system reads concept from the screen field and:

- Transform the concept into path;
- Extract the object (definition) of this concept from the container located by the path in the given layer or from all layers if the relation is replaced by an asterisk “*”.

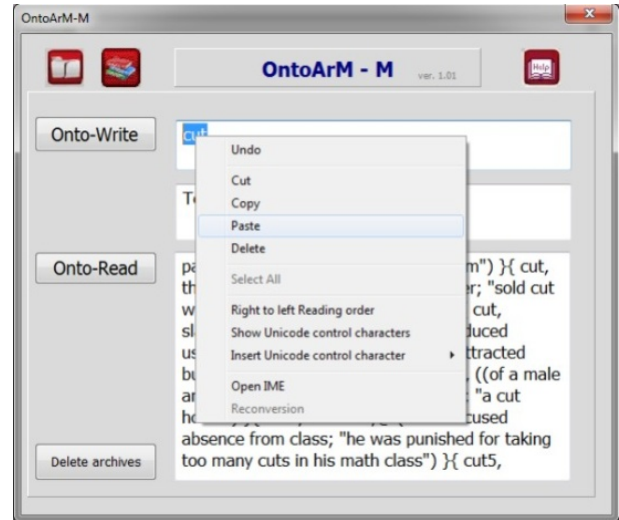
To update the content of a definition one may use form for manual work (Figure 2) and to follow the next steps: (1) Enter the concept and layer; (2) Activate reading current definition pressing the RDF-Read button; (3) Update

content of definition on screen; (4) Press RDF-Write button to store new variant of definition in the correspond archive.

The fields for manual work allow copy/past (Figures 3a and 3b). The service buttons has similar functions as the same in the automated mode. The exit from the system can be done by the conventional way for Windows - by clicking on the cross in the upper right corner of the form.



a. Copy from the input field



b. Past in the field for reading

Figure 3. Copy/past possibilities

The NL-addressing is case sensitive. The words “cut” and “CUT” are absolutely different as NL-addresses. Because of this for words “cut” and “CUT” we have to use separate queries. Of course, it is easy to program the system automatically to use both capital and small letters. In Figure 4 the OntoArM report for the query “cut; *” is shown. The information is shown by program MS WordPad with “no word wrap” option. The definitions are shown “as_is” in the lexicographer files, i.e. the access method does not convert the information to any other style and stores and extracts the information “as_is”.

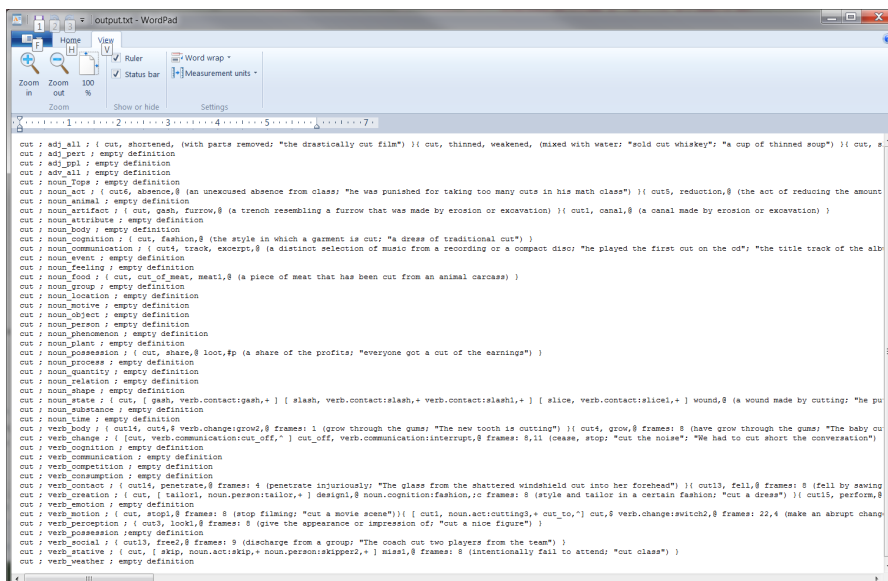


Figure 4. OntoArM report for the query “cut; *”

Experiment for NL-storing ontologies

To make experiment with real data, we use the WordNet [WordNet, 2012] as ontology and its 45 types of relations (given by its files of different types) we stored as 45 layers. The results from experiments for storing WordNet as ontology with 45 layers are given in Table 1.

Time measured during the experiments presented below is highly dependent on the possibilities of operational environment and speed of computer hardware. We provided experiments on the next computer configuration:

- Processor: Intel Core2 Duo T9550 2.66GHz; CPU Launched: 2009;
- Physical Memory: 4.00 GB;
- Hard Disk: 100 GB data partition; 2 GB swap;
- Operating System: 64-bit operating system Windows 7 Ultimate SP1.

Table 1. Experimental data for storing WordNet as ontology

number of layers	operation	number of instances	total time in milliseconds	average time (ms) for one instance
45	writing	117 709	96 643	0.82
45	reading	112 945	91 618	0.81
45	work memory: 538 408 KB; permanent archive: 17 013 KB source text in 45 files - not compressed: 16 338 KB; compressed by WinZip: 4937 KB			

The difference in the numbers of instances in Table 1 is due to removing the equal instances and service information from input files when we use WordNet.

117709 instances were stored for 96643 milliseconds and one instance has been stored for average time of 0.82 milliseconds. 112945 instances were extracted for 89950 milliseconds and one instance has been extracted for average time of 0.80 milliseconds. In both cases, more than one thousand instances were processed for less than one second.

The work memory for storing hash tables and theirs containers was 538 408 KB. To analyze work of system, the work memory was chosen to be in a file in the external memory. In further realizations of OntoArM, to accelerate the speed and to reduce used disk space, work memory may be realized as part of main memory (as dynamically allocated memory or as file mapped in memory).

After finishing the work, occupied disk memory for compressed permanent archives is 17 013 KB, i.e. in this case the NL-indexing takes 12 076 KB additional compressed memory (the 45 sequential files with initial data occupy 16 338 KB, and compressed by WinZip they take 4 937 KB).

After updating, no recompiling of the OntoArM archive is needed. For less than one millisecond after entering new data, the information is ready for using.

Conclusion

In this paper we considered a system called OntoArM aimed for storing complex graph structures such as ontologies by means of Natural Language Addressing.

The presented state of the art in this area has shown that main models for storing ontologies are sequential files and relational databases (i.e. sets of interconnected indexed sequential files with fixed records' structure).

What we gain and loss using NL-Addressing for storing ontologies?

The loss is additional memory for storing internal hash structures. But the same if no great losses we will have if we will build balanced search trees or other kind in external indexing. It is difficult to compare with other systems because such information practically is not published.

The benefit is in two main achievements:

- High speed for storing and accessing the information;
- The possibility to access information immediately after storing without recompilation the database and rebuilding indexes.

For static structured datasets it is more convenient to use standard utilities and compiled indexes. NL-addressing is suitable for dynamic processes of creating and further development of structured datasets due to avoiding recompilation of the database index structures and high speed access to every data element.

Bibliography

- [Artemieva & Reshtanenko, 2008] Artemieva L. I. and N. V. Reshtanenko, "Intellectualized system based on multi-layer chemical ontologies", (Артемяева Л. И. Н. В. Рештаненко, Интеллектуальная система, основанная на многоуровневой онтологии химии/Программные продукты и системы, 2008. № 1, pp. 84-87), http://www.swsys.ru/print/article_print.php?id=113, (accessed: 17.07.2013), (in Russian)
- [Bashmakov, 2005] Bashmakov A.I. Intellectual Information Technologies (Башмаков А. И. Интеллектуальные информационные технологии: Учеб. Пособие. М.: Изд.-во МГТУ им. Н. Э. Баумана, 2005, с.304) (in Russian)
- [Calvanese et al, 2007] Calvanese D., Cuenca B. Grau, Franconi E. Software Tools for Ontology Design and Maintenance FP6-7603 – Thinking ONtologiES (TONES) pp. 1–57. 2007, <http://www.sts.tu-harburg.de/tech-reports/2007/TonesD15.pdf> (accessed: 21.07.2012)
- [Chimaera, 2012] <http://www-ksl.stanford.edu/software/chimaera/> (accessed: 09.08.2012).
- [Dobrov et al, 2009] Dobrov B.V., Ivanov V.V., Lukashevich N.V., Soloviev V.D. Ontologies and Tesaures: models, instruments, applications. (Добров Б. В., Иванов В. В., Лукашевич Н. В., Соловьев В. Д. Онтологии и тезаурусы: модели, инструменты, приложения. Интернет-университет информационных технологий – ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2009, с. 176) (in Russian).
- [ebxml, 2012] <http://www.ebxml.org> (accessed: 25.05.2012).
- [Filatov et al, 2007] Filatov V.A., Shcherbak S.S, Hairova A.A., Development effective tools for creating and processing of ontological knowledge (Филатов В. А., Щербак С. С., Хайрова А. А. Разработка высокоэффективных средств создания и обработки онтологических баз знаний/ Системы обробки інформації, випуск 8 (66), 2007, pp. 120–124), www.nbu.gov.ua/portal/natural/soi/2007_8/Filatov.pdf (accessed:21.07.2012), (in Russian).
- [Gandon, 2002] Gandon F. Ontology Engineering: a survey and a return on experience. ACACIA Team. Thème 3: Interaction homme-machine, images données, connaissances. INRIA: Rapport de recherche n° 4396 - March 2002, pp.181.
- [Gavrilova, 2001] Gavrilova T.A., Knowledge bases of intellectual systems (Гаврилова Т. А. Базы знаний интеллектуальных систем/Т. А. Гаврилова, В. Ф. Хорошев-ский. СПб.: Питер, 2001. С. 384), (in Russian)
- [Guarino, 1998] Guarino N. Formal Ontology and Information Systems/N. Guarino In N. Guarino (ed.) Formal Ontology and Information Systems/FOIS'98, 6–8 June 1998, Trento, Italy: – IOS Press, Amsterdam, 1998, pp. 3–15.
- [Gunther, 1998] Gunther O. Environment Information Systems, Springer, Berlin, New Work, 1998, pp. 244.

-
- [Hayes et al, 2005] Hayes, P., Eskridge, C. T., Reichherzer, T., Saavedra, R., Mehrotra, M., Bobrovnikoff, D. COE: Tools for Collaborative Ontology Development and Reuse. In: Knowledge Capture Conference (K-CAP), 2005.
- [Hertel et al, 2009] Hertel A., J. Broekstra, and H. Stuckenschmidt "RDF Storage and Retrieval Systems" In: S. Staab and R. Studer (eds.), Handbook on Ontologies, International Handbooks on Information Systems, DOI 10.1007/978-3-540-92673-3, Springer-Verlag Berlin Heidelberg 2009. pp 489-508.
- [IBL, 2012] Internet Business Logic http://www.semanticweb.org/wiki/Internet_Business_Logic (accessed: 21.07.2012)
- [Ivanova et al, 2012a] Krassimira Ivanova, Vitalii Velychko, Krassimir Markov. "About NL-addressing" (К вопросу о естественно-языковой 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, 2012b] 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] Krasimira Ivanova, "Storing Data using Natural Language Addressing", PhD Thesis, Hasselt University, Belgium, 2014
- [Kalyanpur et al, 2005] Kalyanpur, A., Parsia, B., Hendler, J. A Tool for Working with Web Ontologies in: Proceedings of the International Journal on Semantic Web and Information Systems, Vol.1, No.1, Jan-Mar (2005)
- [Liebig & Noppens, 2003] Liebig, T., Noppens, O. OntoTrack: Fast Browsing and Easy Editing of Large Ontologies: In: Proceedings of the 2nd International Workshop on Evaluation of Ontologybased Tools (EON-2003) Sanibel Island, Florida, USA (2003), pp. 47-56
- [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).
- [Nevzorova et al, 2004] Nevzorova O.A., Nevzorov V.N. The Analysis of the Structural Features of the Ontology by the Development support system "OntoEditor", (Система визуального проектирования онтологий "OntoEditor": функциональные возможности и применение //IX национальная конференция по искусственному интеллекту с международным участием) KII-2004. Т. 3. pp. 176-183, (In Russian)
- [Noy & Musen, 1999] Noy N, M. Musen. SMART: Automated Support for Ontology Merging and Alignment/Stanford Medical Informatics, Stanford Univ, 1999, pp. 24
- [Noy & Musen, 2002] Noy, N. F., Musen, M. A.: Evaluating ontology-mapping tools: Requirements and experience In: Proceeding of OntoWeb-SIG3 Workshop, 2002, pp. 1-14
- [OntoTools, 2012] A List of Ontology Engineering Tools (Ontology Editors) - <http://www.hozo.jp/OntoTools/>; Mizoguchi Lab., The Institute of Scientific and Industrial Research, Osaka University: <http://www.ei.sanken.osaka-u.ac.jp/> (accessed: 22.07.2012)
- [Ovdei & Proskudina, 2004] Ovdei M .O., Proskudina G.U., "Survey of ontology engineering tools", (Овдей М. О, Г. Ю. Проскудина "Обзор инструментов инженерии онтологий", Российский научный электронный журнал, „Электронные библиотеки“, 2004, т. 7, Вып. 4, ISSN 1562-5419), <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part4/op> (accessed: 21.07.2012), (in Russian).
- [Palagin & Yakovlev, 2005] Palagin A.V., Yakovlev U.S., "System integration of computer technique", (Палагин А. В, Ю. С. Яковлев. Системная интеграция средств компьютерной техники/ А. В. Палагин. Винница: УНІВЕРСУМ, 2005, pp. 680, pp. 677-678, ISBN 966-641-140-7), (in Russian)
- [Palagin, 2006] Palagin A.V. "Architecture of ontologically controled computer systems", (Палагин А. В. Архитектура онтолого-управляемых компьютерных систем /Кибернетика и системный анализ, 2006, №2, pp. 111 – 124), (in Russian)
- [Promt, 2012] <http://protege.stanford.edu/plugins/prompt/prompt.html> (accessed: 09.08.2012)
- [TBC, 2012] Top Braid Composer <http://www.topbraidcomposer.com> (accessed: 21.07.2012)
- [Weibel et al, 1998] Weibel S., J. Kunze, C. Lagoze and M. Wolf "Dublin Core Metadata for Resource Discovery", IETF #2413, The Internet Society, September 1998, <http://dublincore.org/documents/1998/09/dces/> (accessed: 02.09.2012)
- [WordNet, 2012] Princeton University "About WordNet", WordNet, Princeton University, 2010 <http://WordNet.princeton.edu> (accessed: 23.07.2012)

Authors' Information



Ivanova Krassimira – 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