WORDARM - A SYSTEM FOR STORING DICTIONARIES AND THESAURUSES BY NATURAL LANGUAGE ADDRESSING

Krassimira Ivanova

Abstract: *The main features of WordArM system for storing dictionaries and thesauruses by means of Natural Language Addressing are outlined in the paper. Experiments with WordArM have shown that the NL-addressing is suitable for dynamic processes of creating and further development of datasets due to avoiding recompilation of the database index structures and high speed access to every data element.*

Keywords: *Natural Language Addressing*

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

Introduction

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.

In this paper we present program system WordArM based on NLA Access Method and corresponded NLA Archive Manager called NL-ArM [Ivanova, 2014]. Below we will present main features of WordArM.

WordArM

WordArM is a system for storing dictionaries and thesauruses by means of Natural Language Addressing.

WordArM is upgrade over Natural Language Addressing Access Method and corresponded Archive Manager called **NL-ArM**, realized in [Ivanova, 2014]. WordArM is aimed to store libraries of terms and their definitions. WordArM concepts are organized in multi-layer hash tables (information spaces with variable size). The definition of each term is stored in a container located by appropriate path - mapping of the natural language word or phrase, which presents the concept.

There is no limit on the number of terms in a WordArM archive, but their total length plus internal hash indexes could not exceed the file length (4G, 8G, etc.) which is enough space for several millions of concepts’ definitions. There is no limit on the number of files in the data base, as well as their location, including the Internet. This permits to store unlimited number of concepts’ definitions.

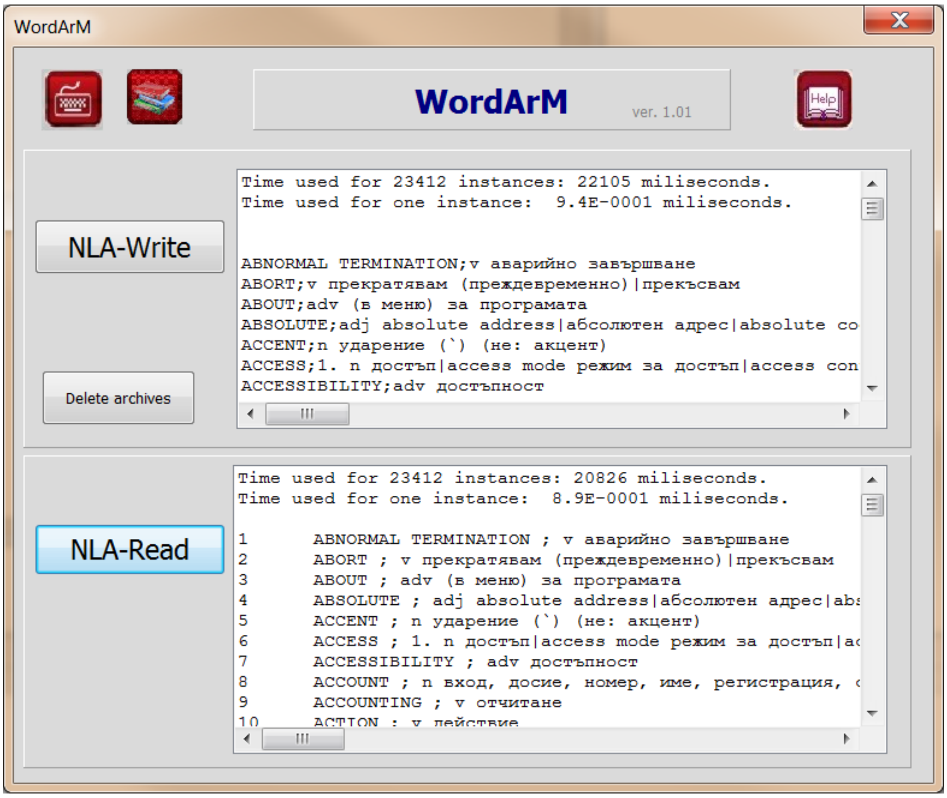
WordArM has two modes of operation: Automated and Manual.

- *The automated mode* supports reading the input information from file (concepts with definitions to be stored in the archive or only list of concepts to receive their definitions from the archive). The result is storing the definitions in the WordArM archive or exporting definitions from the WordArM archive in a file;
- *The manual mode* does the same but only for one concept which is entered manually from the corresponded screen panel.

To support these modes, WordArM has two main operations – information storing (NLA-Write) and information reading (NLA-Read), which have two variants – for automatic input and output of data from and to files, and for manually performing these operations.

WordArM automated mode functions

The WordArM panel for working in automated mode is shown on Figure 1.



```
Time used for 23412 instances: 22105 miliseconds.
Time used for one instance:  9.4E-0001 miliseconds.


ABNORMAL TERMINATION;v аварийно завършване
ABORT;v прекратявам (преждевременно)|прекъсвам
ABOUT;adv (в меню) за програмата
ABSOLUTE;adj absolute address|абсолютен адрес|absolute co
ACCENT;n ударение (`) (не: акцент)
ACCESS;1. n достъп|access mode режим за достъп|access con
ACCESSIBILITY;adv достъпност
```

```
Time used for 23412 instances: 20826 miliseconds.
Time used for one instance:  8.9E-0001 miliseconds.

1       ABNORMAL TERMINATION ; v аварийно завършване
2       ABORT ; v прекратявам (преждевременно)|прекъсвам
3       ABOUT ; adv (в меню) за програмата
4       ABSOLUTE ; adj absolute address|абсолютен адрес|abs
5       ACCENT ; n ударение (`) (не: акцент)
6       ACCESS ; 1. n достъп|access mode режим за достъп|ac
7       ACCESSIBILITY ; adv достъпност
8       ACCOUNT ; n вход, досие, номер, име, регистрация, 
9       ACCOUNTING ; v отчитане
10      ACTION ; v действие
```

Figure 1. Screenshot of WordArM panel for working in automated mode

By “**NLA-Write**” button the function for storing definitions from a file can be activated. Each concept and its definition occupy one record in the file. There is no limit for the number of records in the file. After pressing the “NLA-Write” button, the system reads records sequentially from the file and for each of them:

- Transform the concept into path;
- Store the definition of this concept in the container located by the path.

The input file is in CSV file format. Its records have the next format: **<word/words>;<definition><CR>**.

After storing the concepts’ definitions, WordArM displays the contents of the input file in the field near to the “NLA-Write” button. Before the information from the file, two informative lines are shown (Figure 1):

– Total time used for storing all instances from the file;

– Average time used for storing of one instance

in milliseconds.

In the same panel (Figure 1) corresponded button enables deleting the work archive of the WordArM (ArmDict.dat, which in this version for test control is stored on the hard disk but not in the computer memory). WordArM is completed with compressing program and after storing the information prepares small archive for long time storage.

By “**NLA-Read**” button, the function for reading definitions from the WordArM archive can be activated. In the automated mode, NLA-Read uses as input a file with concepts (each on a separate line) and extract from the archive theirs definitions. If any definition does not exist, the output is empty definition.

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

– Transform the concept into path;

– Extract the definition of this concept from the container located by the path.

The output file is in CSV file format: **<word/words>;<definition><CR>**.

The content of the output file is displayed in the field next to the NLA-Read button. Before the information from the file, two informative lines are shown (Figure 1):

– Total time used for extracting of all instances;

– Average time used for extracting of one instance,

in milliseconds.

Finally, 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 (Help) activates the help text (user guide) of the system.

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.

WordArM manual mode functions

The WordArM panel for working in manual mode is shown on Figure 2. The NL-addressing supports multi-language work. In other words, in the same archive we may have definitions of the concepts from different languages.

By “**NLA-Write**” button the function for storing definitions from the form can be activated. Each concept and its definition can be given in corresponded fields on the screen form. After pressing the “NLA-Write” button, the system reads information from the fields and:

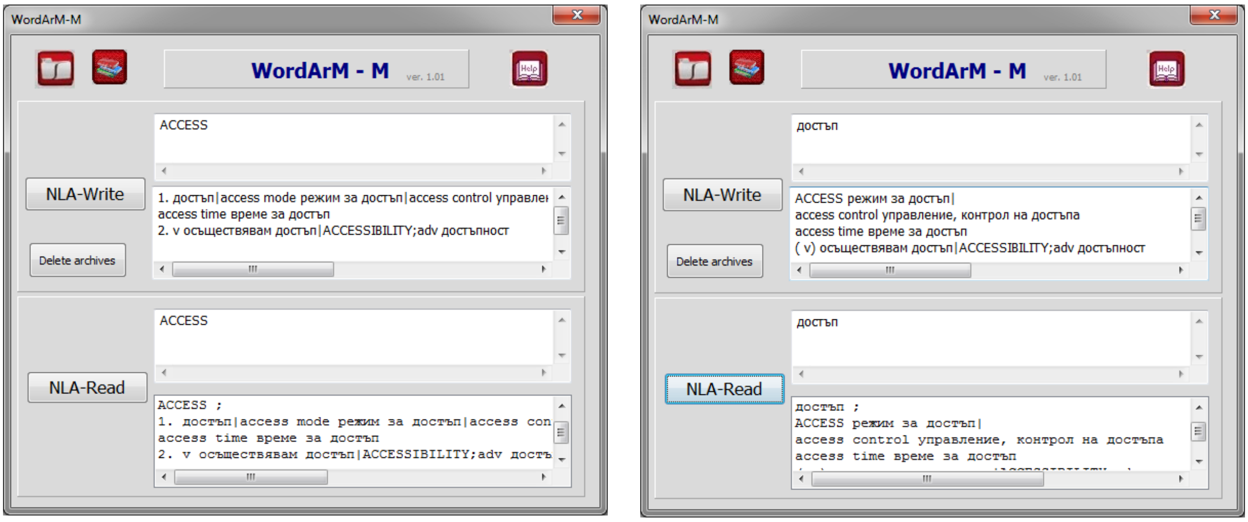
— Transform the concept into path;

— Store the definition of this concept in the container located by the path.

By “**NLA-Read**” button the function for reading a definition from the WordArM archive can be activated. In the manual mode, NLA-Read uses as input the concept given in the screen field and extracts from the archive its definition. If the definition does not exist, the output is empty definition. After pressing the “NLA-Read” button, the system reads concept from the screen field and:

— Transform the concept into path;

— Extract the definition of this concept from the container located by the path.



ACCESS

1. достъп|access mode режим за достъп|access control управлен
access time време за достъп
2. v осъществявам достъп|ACCESSIBILITY;adv достъпност

ACCESS

ACCESS ;
1. достъп|access mode режим за достъп|access con
access time време за достъп
2. v осъществявам достъп|ACCESSIBILITY;adv достъ

достъп

ACCESS режим за достъп|
access control управление, контрол на достъпа
access time време за достъп
(v) осъществявам достъп|ACCESSIBILITY;adv достъпност

достъп

достъп ;
ACCESS режим за достъп|
access control управление, контрол на достъпа
access time време за достъп

Figure 2. Screenshots of WordArM panel for working in manual mode showing simultaneous work with concepts defined in different languages

The fields for manual work allow copy/past options (Figure 3a and Figure 3b).

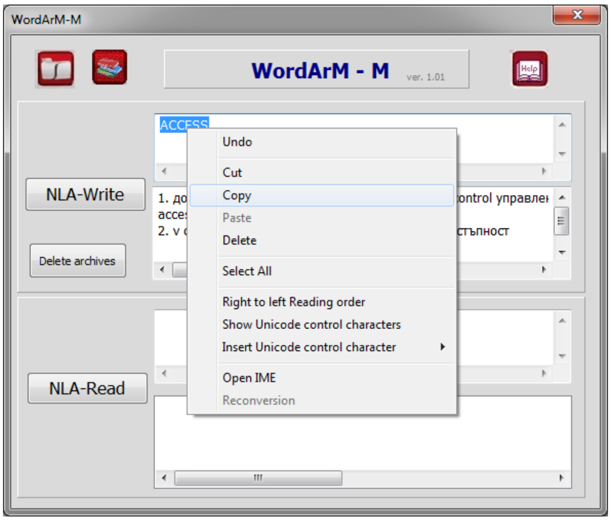


Figure 3a. Copy from the input field

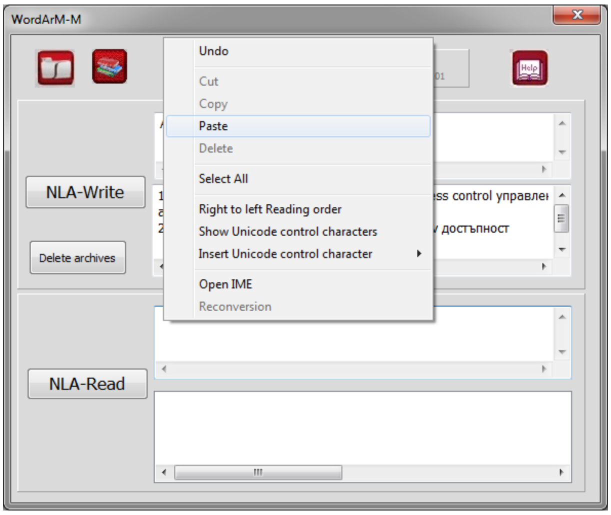


Figure 3b. Past in the field for reading

The service buttons has similar functions as the same in the automated mode.

Experiments with WordArM

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.

Theoretical background of WordArM was presented in [Ivanova et al, 2013c]. Below we will remember main results from it.

NL-storing dictionaries

Our first experiment was to realize a small multi-language dictionary based on NL-addressing. We have taken data from the popular in Bulgaria “SA Dictionary” [Angelov, 2012]. SA Dictionary is a computer dictionary, which translates words from Bulgarian language to English and vice versa.

For experiments we used a list of ***23 412*** words in English and Bulgarian with their definitions in Bulgarian, stored in a sequential file with size of 2 410 KB.

For storing dictionaries we used simple model: the words (concepts) are used as paths to theirs definitions stored in corresponded terminal containers.

The speed for storing, accessing, and size of the work memory and permanent archives are given in Table 1.

Work memory is the memory taken for storing hash tables and service information. Usually it has to be part of main computer memory. To analyze its real size in our experiments, work memory is allocated as file.

Permanent archives are static copies of work memory (zipped files), aimed for long storing the information. They have to be of small size and converting to and from expanded work memory structures has to be quick (usually several seconds or minutes). For compressing of work memory we use a separate archiving program.

Table 1. Experimental data for NL-storing of a dictionary

operation	number of instances	total time in milliseconds	average time for one instance	work memory	permanent archive
NL-writing	23 412	22 105	0.94 ms	80 898 KB	5 938 KB
NL-reading	23 412	20 826	0.89 ms		

The work memory taken during the work was ***80 898*** KB.

After finishing the work, occupied permanent compressed archive is ***5 938*** KB. This means that the NL-indexing takes 5 102 KB additional compressed disk memory (the sequential file with initial data is 2 410 KB and compressed by WinZip it is 836 KB).

To analyze work of the system, work memory was chosen to be in a file but not in the main memory. In further realizations of WordArM, it may be realized as a part of main memory of computer as:

- Dynamically allocated memory;
- File mapped in memory.

In this case, the speed of storing and accessing will be accelerated and used hard disk space will be reduced.

The analysis of the results in Table 1 shows that the NL-addressing in this realization permits access practically equal for writing and reading for all data. The speed is more than a thousand instances per second. *Reading is possible immediately after writing and no search indexes are created.*

NL-storing thesauruses

We have used NL-addressing to realize the WordNet lexical database [WordNet, 2012]. The WordNet database organization has the following important disadvantages: 1) Relative addressing is convenient for the computer processing, but it is difficult to be used by the customer; 2) Manual creating of numerical addresses is impossible, and their use can be done only by the special program; 3) The end user has access only to the static ("compiled") version of the database, which couldn’t be extended and further developed; 4) Building the WordNet database requires the use of the “Grinder” program and the processing of all lexicographers’ source files at the same time; 5) Using the current format is not only cumbersome and error-prone, but also limits what can be expressed in a WordNet resource.

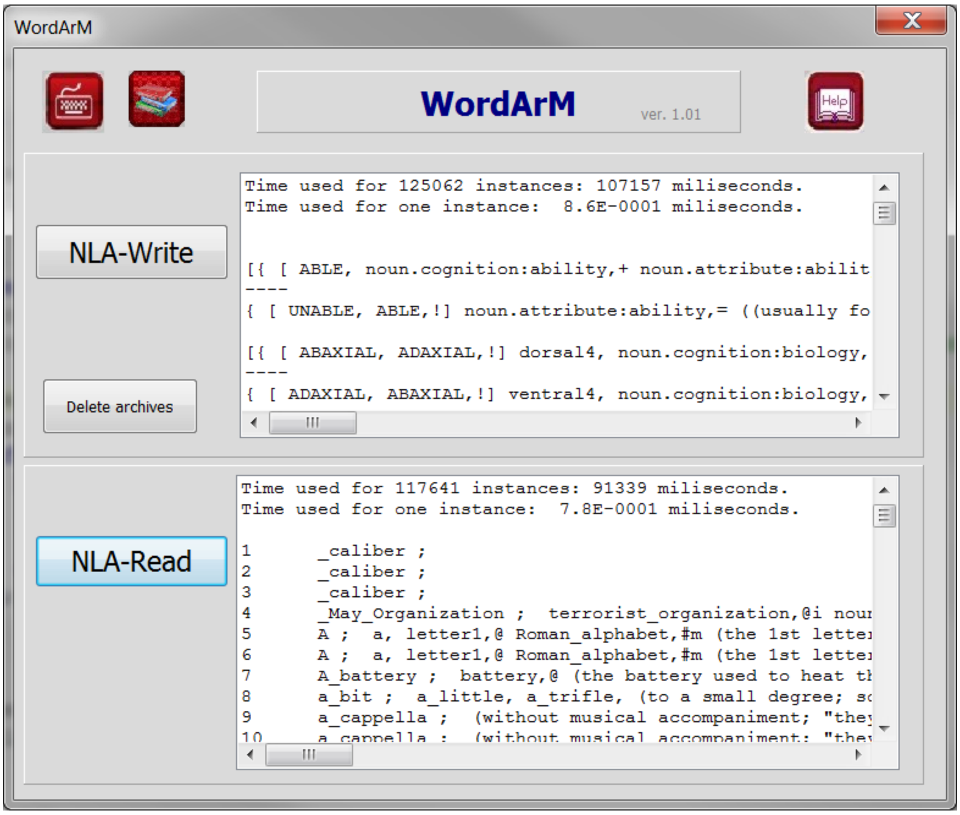
The main source information of WordNet is published as lexicographer files. The total number of instances (file records) is 117 871, but 206 instances contain service information (not concepts’ definitions), so we have 117 665 instances for experiments, distributed in 45 thematically organized lexicographer files. It is important to note that there is equal synsets in several lexicographer files. This has matter when we integrate the 45 files in one source file for representing a thesaurus, i.e. for experiment we have used all 45 files concatenated in one big file as thesaurus with more than one hundred thousands of concepts. The results are given in Table 2. A screenshot from the WordArM for the case of WordNet as thesaurus is shown at Figure 4.

We receive practically the same results as for storing dictionaries.

The analysis of the results in Table 2 shows that the NL-addressing permits access practically equal for writing and reading for all data. The speed is more than a thousand instances per second. Reading is possible immediately after writing and no search indexes are created.

The work memory for hash tables and their containers taken during the work of WordArM was *385 538 KB*. To analyze work of the system, work memory was chosen to be in a file in the external memory. In further realizations, to accelerate the speed and reduce of used disk space, the 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 permanent archive for compressed archive is *15 603 KB*, i.e. in this case the NL-indexing takes 14 270 KB additional compressed memory (the sequential file with initial data is 1 333 KB).



```
Time used for 125062 instances: 107157 miliseconds.
Time used for one instance:  8.6E-0001 miliseconds.


[{ [ ABLE, noun.cognition:ability,+ noun.attribute:abilit
----
{ [ UNABLE, ABLE,!] noun.attribute:ability,= ((usually fo

[{ [ ABAXIAL, ADAXIAL,!] dorsal4, noun.cognition:biology,
----
{ [ ADAXIAL, ABAXIAL,!] ventral4, noun.cognition:biology,
```

```
Time used for 117641 instances: 91339 miliseconds.
Time used for one instance:  7.8E-0001 miliseconds.

1      _caliber ;
2      _caliber ;
3      _caliber ;
4      _May_Organization ;  terrorist_organization,@i nou
5      A ;  a, letter1,@ Roman_alphabet,#m (the 1st lette
6      A ;  a, letter1,@ Roman_alphabet,#m (the 1st lette
7      A_battery ;  battery,@ (the battery used to heat th
8      a_bit ;  a_little, a_trifle, (to a small degree; s
9      a_cappella ;  (without musical accompaniment; "the
10     a_cappella ;  (without musical accompaniment; "the
```

Figure 4. WordArM results for the case of WordNet as thesaurus

Table 2. Experimental data for storing WordNet as thesaurus

operation	number of instances	total time in milliseconds	average time for one instance
writing	125 062	107 157	0.86 ms
reading	117 641	91 339	0.78 ms
work memory: 385 538 KB; permanent archive: 15 603 KB; source text: 1 333 KB			

Conclusion

In this paper we presented program system WordArM based on NLA Access Method and corresponded NLA Archive Manager called NL-ArM [Ivanova, 2014]. Main features of WordArM were outlined.

WordArM is aimed to support experiments. Because of this it was realized with two modes – automatic and manual. In addition, work memory of the system was realized as disk files for analyzing its behavior during the experiments.

Analyzing results from the experiment with a real dictionary data we may conclude that it is possible to use NL-addressing for storing such information. Next experiment was aimed to answer to question: “What we gain and loss using NL-Addressing for storing thesauruses?”

The loss is additional memory for storing hash structures which serve NL-addressing. 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 the information immediately after storing without recompilation the database and rebuilding the indexes.

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

Bibliography

[Angelov, 2012] St. Angelov. SA Dictionary http://www.thediction.com/ (accessed: 11.01.2013)

[Ivanova et al, 2012a] 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, 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

[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).

[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