INTRODUCTION TO THE NATURAL LANGUAGE ADDRESSING

Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko

***Abstract**: Natural Language Addressing (NLA) is a possibility to access information using natural language words or phrases as direct addresses of the information in the computer memory. For this purpose the internal encoding of the letters is used to generate corresponded address co-ordinates. This paper outlines the main idea of NLA.*

***Keywords**: addressing; natural language addressing.*

***ACM Classification Keywords**: A.1 Introductory and Survey; D.4.3 File Systems Management, Access methods.*

Introduction

The world around us can be described in one word as "Variety". It is difficult to agree that the world hasn't so needed orderliness, created over millennia, developed and maintained constantly as oasises of order in the core of the chaos... It is strange for our perception of the world as a four-dimensional existence. It is strange, because our mind builds a completely different picture of ordered spatiality and extensity.

The concept "order" has many meanings but here it is used in the sense of a condition of logical or comprehensible arrangement among the separate elements of a group [AHD, 2009]; a state in which all components or elements are arranged logically, comprehensibly, or naturally; sequence (alphabetical order) [Collins, 2003]; arrangement of thoughts, ideas, temporal events [WordNet, 2012].

One very important aspect of the order is that every entity of the ordered set has its own location in it. The names of these locations are called **addresses**.

The common sense meaning of the concept "**address**" is such as a description of the location of a person or organization, as written or printed on mail as directions for delivery [AHD, 2009]; the conventional form by which the location of a building is described [Collins, 2003]; a sign in front of a house or business carrying the conventional form by which its location is described [WordNet, 2012].

We will use the concept "address" in the sense accepted in the Computer Science: *the code that identifies where a piece of information is stored* [WordNet, 2012]; a name or number used in information storage or retrieval that is assigned to a specific memory location; the ***memory location identified by this name or number***; a name or a sequence of characters that designates an e mail account or a specific site on the Internet or other network [AHD, 2009].

Usually we make difference between human aspect of the concept "address" and its computer "understanding". This paper is aimed to introduce an approach of using human representation of the address by natural language words as computer memory address. This approach we call "Natural Language Addressing" (NLA).

The paper is structured as follows: in the next section we outline the addressing in the textual information models. After that we discuss the naming the locations and using encoding of the names as addresses. In conclusion we consider possible implementations of NLA.

Addressing in the textual information models

The concept “address” is closely connected with the terms "information interaction” and “information model”. [Markov, 1984; Markov et al, 2003]

We continuously build information models of the world and of ourselves in this world. The need of coordinating our actions with others humans or intelligent devices requires constant information exchange (interaction), the basis of which are the information models.

In the Computer Science, the term “information model” is most popular. The “Information Model” is an abstract but formal representation of entities including their properties, relationships and the operations that can be performed on them [DN, 2013]. “An information model is a representation of concepts, relationships, constraints, rules, and operations to specify data semantics for a chosen domain of discourse. The advantage of using an information model is that it can provide sharable, stable, and organized structure of information requirements for the domain context. An information modeling language is a formal syntax that allows users to capture data semantics and constraints” [Lee, 1999].

It is wrong to believe that the information models are a phenomenon only of humans. But only for humans there exist letters and accordingly textual (formal or not formal) information models. The simplest textual information model is a linear structure of text elements – letters, words, sentences or more complicated structures like tags in the Extensible Markup Language [XML, 2013].

The important elements of textual models may be defined by corresponded definitions located in different places of the text. If the concepts together with theirs definitions are ordered alphabetically, like in a dictionary (Figure 1), going through text one may find needed concept and its definition.

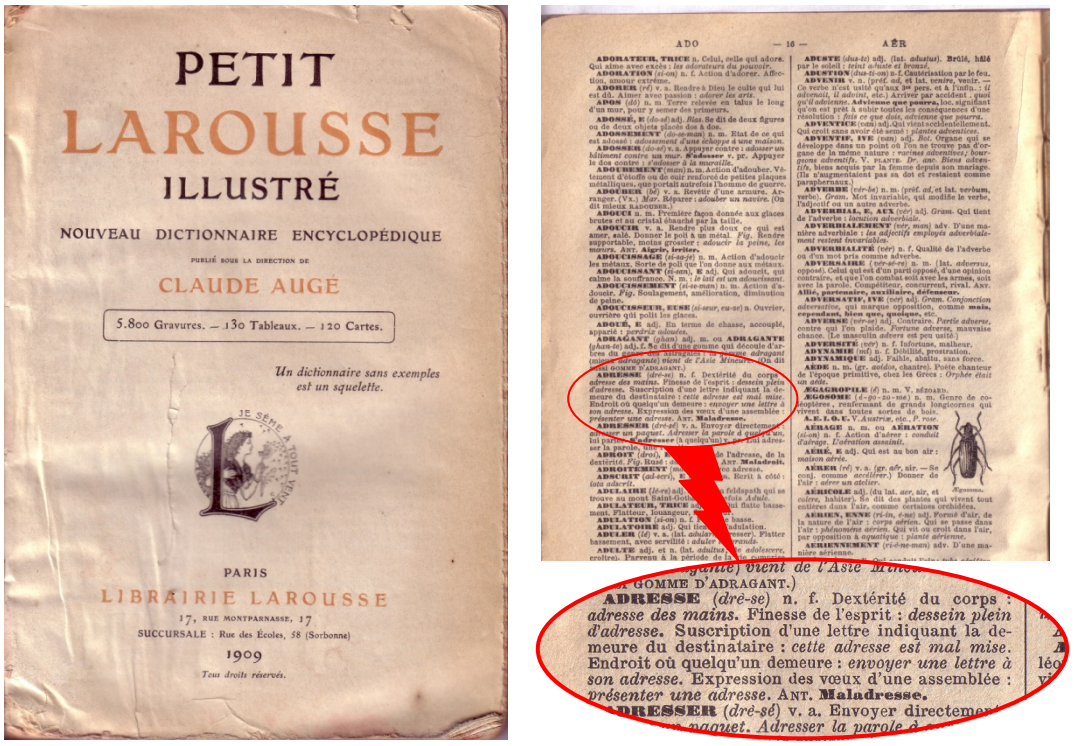

Figure 1. Addressing by natural language order [Auge, 1909]

Some models have internal structure which may be divided on substructures, etc. For instance, the Brookshear’s textbook “Overview of the Computer Science” is such model. It is represented by a book with chapters (Figure 2)

[Brookshear, 2012]. It is a non-formal information model. It is a *complex* one because it contains textual elements, graphics, and pictures. When definitions are placed randomly in a book, for the sake of convenience, at the end of book, an index is located. It contains main concepts and pointers to pages where the concepts are defined. One needs to follow simple algorithm to find a definition. This is illustrated at Figure 2 for the concept "address, of memory cell".

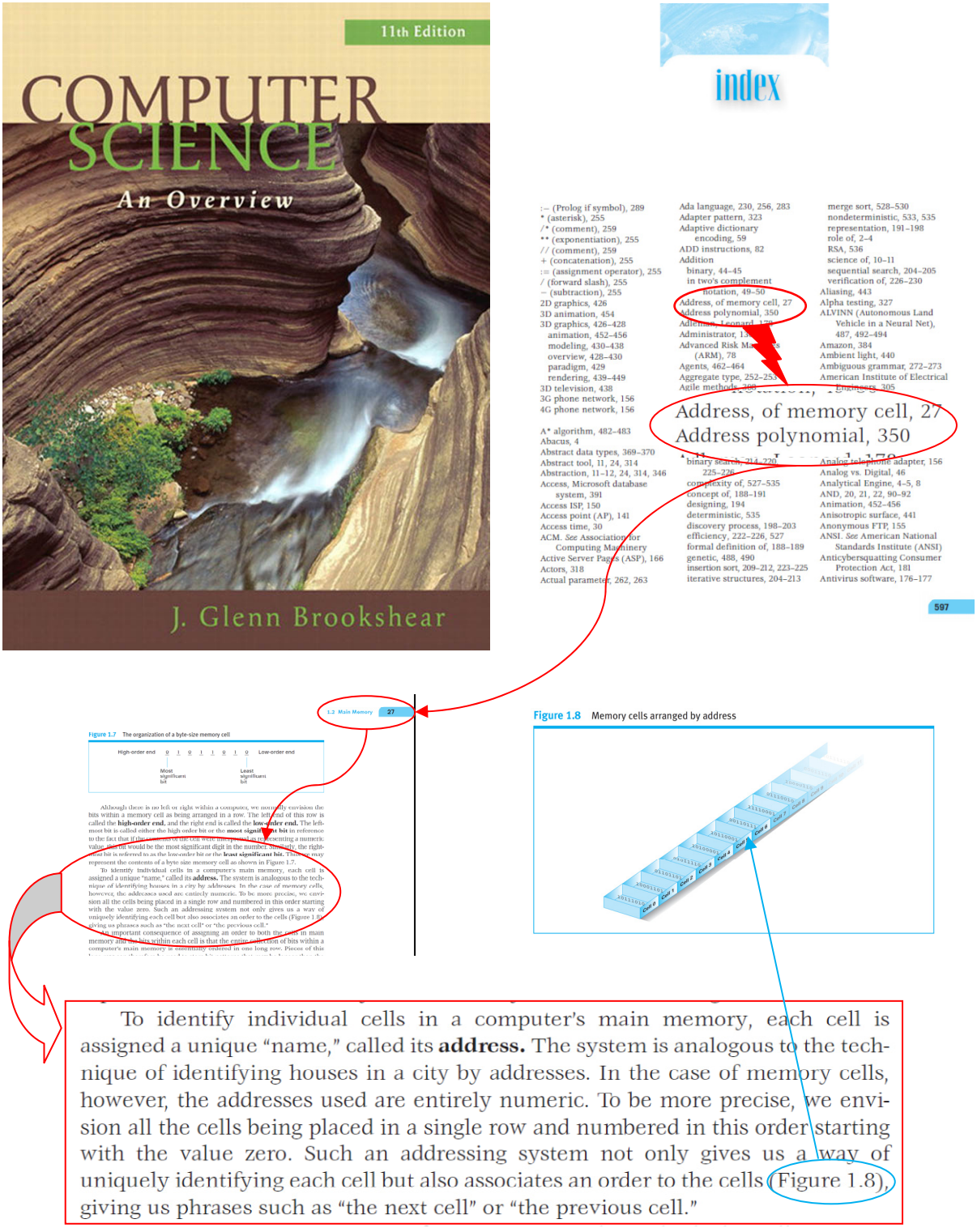

Figure 2. Addressing by indexing [Brookshear, 2012]

Examples of nonlinear textual information models are graphs of interconnected text elements. An example of such model is graphical representation of ontology. Other examples are relational structures usually represented by sets of tables.

In other words, irrespective of type of the textual information model, every its element has its own location and, respectively, its own address in text. Some of the elements may be so important to be pointed by their addresses in an index. It is a sequential arrangement of material, especially in alphabetical or numerical order, which serves to guide, point out or otherwise facilitate reference, especially: a more or less detailed alphabetized list of names, places, subjects, etc, treated in the text of a printed work. It usually appears at the end of book and identifies page numbers on which information about each subject appears [AHD, 2009; Collins, 2003].

Sets of concepts and their definitions, organized in dictionaries, are ordered alphabetically and this way location of every concept and its definition may be found easily.

Computer indexes

The textual information models may be stored in the (internal or external) computer memory. Locating the concepts and definitions may be done by:

- Direct scanning the files;
- Indexing and based on it search of the pointer to address of text element.

Scanning the files is convenient only for small volumes of concepts and definitions. Some rationalization is possible using some algorithms like binary search.

Indexing is creating tables (indexes) that point to the location of folders, files and records. Depending on the purpose, indexing identifies location of resources based on file names, key data fields in a database record, text within a file or unique attributes in a graphics or video file [PC mag, 2013].

In database design, an index is a list or a reference table of keys (or keywords), each of which identifies a unique record or document and is used to locate a particular element within a data array or table. Indexes make it faster to find specific records and to sort records by the index field that is, the field used to identify each record [Webopedia, 2013; AHD, 2009; Collins, 2003].

For large volumes of concepts, the indexes became too large and additional, secondary indexing is needed. Such multi-level index structures are well-known B-trees of Rudolf Bayer [Bayer, 1971] as well as B+trees [Knuth, 1997] (Figure 3).

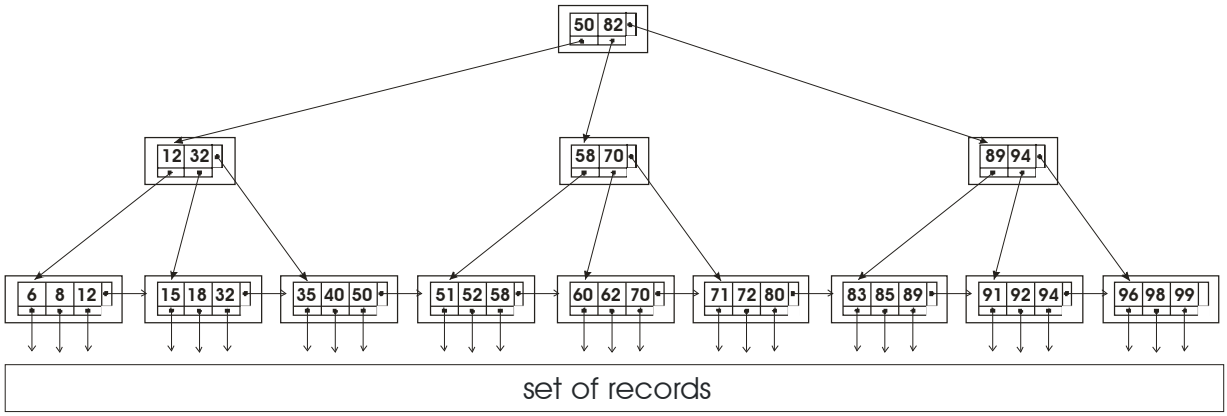

Figure 3. B-tree

Let repeat, the main idea of indexing is to facilitate the search by search in (multi-level) index and after that to ensure the direct access to address given by pointer. In other words, the goal of data indexing is to ease the search of and access to data at any given time. This is done by creating a data structure called index and providing faster access to the data. Accessing data is determined by the physical storage device being used. Indexing could potentially provide large increases in performance for large-scale analysis of unstructured data. Additionally the implementation of the chosen index must be suitable in terms of index construction time and storage utilization [Faye et al, 2012].

Indexing needs resources: memory for storing additional information and time for processing, which may be quite a long, especially for updating of the indexes when new elements are added or some old ones are removed.

Naming the addresses

Basic element of an index is couple:

(name, address).

For instance such couples on Figure 2 are:

("Address, of memory cell", 27)

("Address polynomial", 350)

In different sources the "name" is called "key", "concept", etc. The address usually is given by any "number", "pointer", etc.

There are two interpretations of the couple (concept, address):

1) The address is a connection of the concept with its definition, i.e. practically we have triple:

(name, address, definition)

2) The concept is a name of the address and may be used for user friendly style of programming and the third part (definition) may be variable.

In the very beginning, replacing the address by name was used in the programming languages for pointing the addresses by names (identificators), like in Algol 60 [Naur, 1963].

Later, the same idea was used in the Web navigation systems. Web navigation is mostly based on Uniform Resource Locators (URLs). URLs can be hard to remember and change constantly. For instance, in the "International Human-Friendly Web Navigation System", the Company "RealNames" offered an alternative Web addressing scheme using natural language, replacing unfriendly URLs like http://www.fordvehicles.com/vehiclehome.asp?vid=12 with common names such as "Ford Mustang". Building a fully international system that provides a human-friendly naming infrastructure for the whole Web is a challenging task. By leveraging Unicode to represent names it is possible to build a global naming engine that, coupled with knowledge of local customs simplifies Web navigation through the use of natural language keywords [Arrouse, 1999].

Some of the electronic spreadsheets have possibility to point a group of cells and/or rows with a name and further to use this name in functions and other operations assuming all cells and/or rows named by this name. For instance, "Zoho Sheet" [Zoho sheet, 2012; Zoho sheet_blog, 2012] can recognize and correlate names used in formulas with cells/cell ranges automatically. You have to just give the row/column header of a table as arguments to functions and Zoho Sheet will auto-recognize the cell range associated with the name. It is very

convenient to quickly type in the formulas with these names instead of worrying about keying in the proper cell range.

Consider the following sheet (Figure 4), available at http://zohosheet.com/public.do?fid=25835.

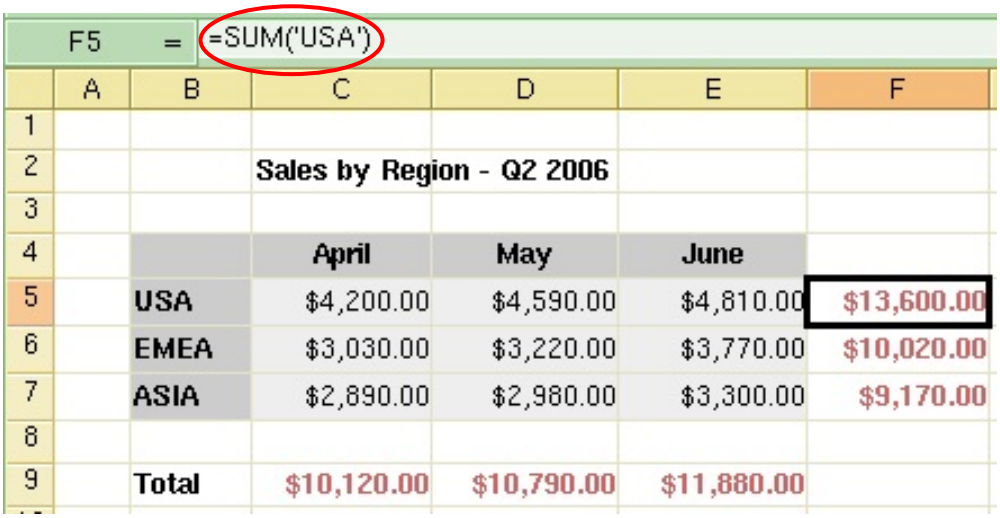

Figure 4. Natural Language Addressing in a spreadsheet

Look at the formulas in the cells F5:F7 and C9:E9. The formula =SUM (USA) will automatically add the cell values in the row with the header 'USA'. Earlier you had to use =SUM (C5:E5). Now the row header can directly be used. You do not even need to name/label the cell ranges. You can even copy and paste these formulas to adjacent rows or columns and they will automatically be adjusted relatively. In this case, copying the F5 cell and pasting it to F6, will result in the formula =SUM (EMEA) in F6.

The approach of replacing cell addresses with names in Zoho Sheet was called "Natural Language Addressing".

Using the encoding of name as address

In this paper we propose to use the computer encoding of name (concept) letters as address of connected to it information. 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 shown at Figure 1 where no explicit index is used but the concept by itself locates the definition. Our approach is similar to one in the Zoho Sheet, too.

Because of this we will use the same term: "***Natural Language Addressing***".

For instance, let have the next definition:

"London: The capital city of England and the United Kingdom, and the largest city, urban zone and metropolitan area in the United Kingdom, and the European Union by most measures."

In the computer memory, for example, it may be stored at address "FF084920" and the index couple is:

("London", "FF084920")

At the memory address "FF084920" the main text, "*The capital … measures.*" will be stored.

To read/write the main text, firstly we need to find name "London" in the index and after that to access memory address "FF084920" to read/write the definition.

If we assume that name “London” in the computer memory is encoded by six numbers (letter codes), for instance by using ASCII encoding system London is encoded as (76, 111, 110, 100, 111, 110), than we may use these codes as direct address to memory, i.e.

(“London”, “76, 111, 110, 100, 111, 110”)

One may remark that above we have written two times the same name and this is truth. Because of this we may omit this couple and index, and read/write directly to the address “76, 111, 110, 100, 111, 110”.

For human this address will be shown as “London”, but for the computer it will be “76, 111, 110, 100, 111, 110”.

From other point of view, the array “76, 111, 110, 100, 111, 110” may be assumed as co-ordinates of a point in a multidimensional (in this case – six dimensional) information space and the definition can be stored in this point.

The natural language does not contain words only of six letters long. The length of the words is variable and in addition there exits names as phrases like “Address polynomial” above. This means that we really have multidimensional address space defined by set of all natural words and phrases.

What we need is a program function which converts such multidimensional addresses in concrete linear machine locations (on the hard disk, for example).

Conclusion

Natural Language Addressing (NLA) is a possibility to access information using natural language words or phrases as direct addresses of the information in the computer memory. For this purpose the internal encoding of the letters is used to generate corresponded address co-ordinates. In this paper we outlined the main idea of NLA and illustrated its genesis.

NLA is applicable for storing dictionaries, thesauruses, and ontologies. The further work is to realize experimental software programs to implement the idea of NLA and to provide practical tests.

Bibliography

[AHD, 2009] The American Heritage® “Dictionary of the English Language” Fourth Edition copyright© 2000 by Houghton Mifflin Company, Updated in 2009; Published by Houghton Mifflin Company. All rights reserved.

[Arrouse, 1999] Arrouye Y. The RealNames System - an International Human-Friendly Web Navigation System http://www.unicode.org/iuc/iuc16/a333.html (accessed: 16.11.2012).

[Auge, 1909] Claude Auge (ed.) „Petit Larouse Illustré”, Librarie Larouse, Paris, 1909.

[Bayer, 1971] Rudolf Bayer. „Binary B-Trees for Virtual Memory”, ACM-SIGFIDET Workshop 1971, San Diego, California, Session 5B, pp. 219 - 235.

[Brookshear, 2012] J. Glenn Brookshear “Computer science – an overview (11-th edition)”, Copyright© 2012, 2009, 2007, 2005, 2003, Pearson Education, Inc., publishing as Addison-Wesley, 2012 ISBN 10: 0-13-256903-5; ISBN 13: 978-0-13-256903-3. pp. 19-72

[Collins, 2003] “Collins English Dictionary – Complete and Unabridged”, HarperCollins Publishers, 1991, 1994, 1998, 2000, 2003

[DN, 2013] "Information Model." Definitions.net. STANDS4 LLC, 2013. Web. 13 Apr. 2013. <http://www.definitions.net/definition/Information Model>. (accessed: 12.03.2013)

[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.

[Knuth, 1997] Donald Knuth “The art of computer programming”, vol 1., Fundamental Algorithms, Third Edition, Addison-Wesley, 1997, ISBN 0-201-89683-4. Section 2.3, especially subsections 2.3.1–2.3.2 (pp. 318–348).

[Lee, 1999] Y. Tina Lee. Information Modeling: From Design to Implementation. Proceedings of the Second World Manufacturing Congress: Manifacturing Systems, Technology, Management. ICSC 1999, ISBN: 9783906454191, pp 315—321.

[Markov et al, 2003] Kr. Markov, Kr. Ivanova, I. Mitov "General Information Theory", Basic Formulations. FOI ITHEA, ISBN 954-16-0024-1, Sofia, 2003.

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

[Naur, 1963] Peter Naur (ed.) Revised Report on the Algorithmic Language Algol 60. Communications of the ACM, Vol. 6, Number 1, Jan. 1963.

[PC mag, 2013] PC Magazine Enciclopedia http://www.pcmag.com/encyclopedia_term/0,1237,t=indexing&i=44896,00.asp (accessed: 23.01.2013)

[Webopedia, 2013] Webopedia QuinStreet, Inc. http://www.webopedia.com/TERM/I/index.html (accessed: 23.01.2013)

[WordNet, 2012] Princeton University "About WordNet", WordNet, Princeton University, 2010 http://WordNet.princeton.edu (accessed: 23.07.2012)

[XML, 2013] W3C. Extensible Markup Language (XML) http://www.w3.org/XML/ (accessed: 12.03.2013)

[Zoho sheet, 2012] https://public.sheet.zoho.com/public.do?docurl=Natural+Language+Formulas&name=m7faALWlvQLgtPUoKu5%2FAA%3D%3D (accessed: 26.11.2012)

[Zoho sheet_blog, 2012] http://www.zoho.com/sheet/blog/natural-language-addressing-in-formulas.html (accessed: 16.11.2012).

Authors' Information

Ivanova Krassimira *– University of National and World Economy, Sofia, Bulgaria. Institute of Mathematics and Informatics, BAS, 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

Vanhoof Koen - *Professor Dr.,*

Universiteit Hasselt; Campus Diepenbeek; Department of Applied Economic Sciences

Wetenschapspark 5; bus 6; BE-3590 Diepenbeek; Belgium

Main research areas: data mining, knowledge retrieval.

e-mail: koen.vanhoof@uhasselt.be

Markov Krassimir *– ITHEA ISS IJ, IBS and IRJ Editor in chief, P.O. Box: 775, Sofia-1090, Bulgaria; e-mail: markov@foibg.com*

Major Fields of Scientific Research: General theoretical information research, Multi-dimensional information systems

Velychko Vitalii – *Institute of Cybernetics, NASU, Kiev, Ukraine*

e-mail: Velychko@rambler.ru

Major Fields of Scientific Research: Data Mining, Natural Language Processing