Data transition chain*

Ladislav Hluchy, Martin Seleng, Viktor Oravec, Ivana Budinska, Michal Laclavik, Emil Gatial, Zoltan Balogh, and Marek Ciglan

Institute of Informatics Slovak Academy of Sciences, Dubravska cesta 9, 845 07 Bratislava, Slovakia,
ladislav.hluchy@ui.sav.sk,
WWW home page: http://ui.sav.sk

Abstract. *The paper presents a chain of tools for transition of data from internet to the data for presentation according user's circumstance conditions and requirements; that is developed in the scope of the NAZOU[1] project. The tools are mainly responsible for acquisition and maintenance of data but there is also a tool responsible for annotation of retrieved relevant documents from the web. The chain is applied within a job offers domain; however it can be extendible also for other domains. The following tools are presented: RIDAR, ERID, WebCrawler, DocConverter, NALIT[2], ExPoS/OSID, RFTS, RDB2Onto, Tvaroslovník[3], and OnTeA. Also main methods of individual tools and data transition chain from the internet to the users are presented here.*

1 Motivation

There is an enormous amount of information from various domains and in various formats in Internet and WWW. Many approaches exist to simplify retrieving and acquiring knowledge and information from heterogeneous and distributed resources. Among them, the information retrieval techniques and methods are the most developed and exploited. Methods of informational retrieval are used also within big web search engines, like Google[4], Yahoo[5], etc. Google and most other web engines utilize PageRank[6] and more than 150 criteria to determine relevancy. PageRank[1] is based on citation analysis that was developed in the 1950s by Eugene Garfield[2] at the University of Pennsylvania. In this way virtual communities of webpages are found. Teoma's[7] search technology uses a communities approach in its ranking algorithm. Nippon Electric Corporation—NEC Research Institute[8] has worked on similar technology. Web link analysis[9] was first developed by Jon Kleinberg and his team while working on the CLEVER[10] project at IBM's Almaden Research Center. In 2004, Yahoo! launched its own search engine based on the combined technologies of its acquisitions and providing a service that gave preeminence to the Web search engine over the directory. However, there are a lot of chalenges in developing new and more effective methods and techniques for information retrieval.

Besides some automatic and semi-automatic techniques and methods, experts knowledge can be used to increase effectivness and to simplify information retrieval within one specific domain considering user's requirements and circumstance conditions. A problem that is adressed in this paper is a transition of information in various types data and formats to prepare data for presentation to users in a consistent presentation frame.

2 Introduction

Tools in acquisition, maintenance and annotation chain are indirectly integrated utilizing the corporate memory[3] sharing asynchronously updated information space. Each tool has to solve adaptation to the changes in all parts of the corporate memory. Data transition chain obtains tools for data acquisition, data organization and maintenance.

Tools in the data acquisition chain solve this issue independently. Tools are using pulling data method to cooperate. Every tool of the chain only adds the stamp (in document's metadata) of the data's actualization (mainly the timestamp). Each document's metadata consists of the set of timestamps created by particular tool. Other tools check their own stamps and perform particular action (SQL query based on this stamp or SQL query based on empty fields in the database tables). The integration of the following tools is presented: RIDAR, ERID, WebCrawler, DocConverter, NALIT, ExPoS/OSID, RFTS, RDB2Onto, Tvaroslovník and Ontea (Fig. 1). These tools are integrated in one sub chain and the output of the last

* This work is supported by projects NAZOU SPVV 1025/2004

[1] *http://nazou.fiit.stuba.sk/home/index.php*
[2] *http://nazou.fiit.stuba.sk/home/?page=nalit-tool*
[3] *http://nazou.fiit.stuba.sk/home/?page=tvaroslovnik-tool*
[4] *http://www.google.com/*
[5] *http://www.yahoo.com/*
[6] *http://www.google.com/technology/*
[7] *http://www.teoma.com/*
[8] *http://www.neci.nj.nec.com/*
[9] *http://www.research.rutgers.edu/~davison/discoweb/*
[10] *http://www.almaden.ibm.com/projects/clever.shtml*

tool in the chain (OnTeA) is used to fulfill the domain ontology (job offers). These tools are not visible in the presentation layer and that is not necessary to wait for all tools in the chain to complete their work. That is why the indirect data oriented integration via the corporate memory was applied.
The tools transform data and the data are stored in the corporate memory as it is depicted in the Fig. 2)
Individual tools work with dose of data and are running in the background. Their running is managed by UNIX system service CRON. The system service CRON runs all the tools in the specified time and date (in the current implementation once a day in different times).

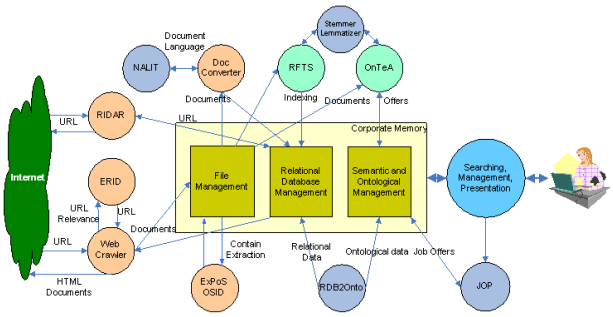

Fig. 1. Integration of tools responsible for acquisition, organization, maintenance and annotation

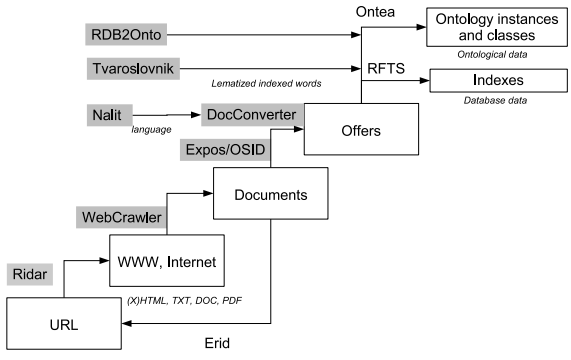

Fig. 2. Data transition chain

3 Description individual tools in the chain

3.1 RIDAR tool

As we can see in the Fig. 1, the first tool in the chain is a RIDAR tool (Relevant Internet Data Resource). RIDAR is using a Google API[11] and Yahoo! API[12] for searching the web using these two search engines. The goal is to identify the relevant web sources on the Internet and the retrieved results in the form of URL are written to the relational part of the corporate memory (step 1 in the Fig. 3).

3.1.1 Methods used by RIDAR tool

RIDAR is used for primary source identification of job offers on the Internet. This method exploits the potential of existing search services for acquisition of links to potential resources of job offers on the Internet. Thus extensive space of keyword-indexed data sources, covered by existing search engines such as Google , AllTheWeb[13] or Yahoo!, is utilized.
Method utilizes search based on keywords or collocations describing particular domain in existing search engines. The tool integrates search engines providing APIs (Application Programming Interface). Integration of search engines includes following three steps:

- registration,
- downloading of developing libraries,
- libraries integration.

Search engines Google and Yahoo! have been registered for purposes of NAZOU project. API has been implemented using web services using SOAP and WSDL standards which are also used for communication with search engines.

3.2 WebCrawler and ERID tools

WebCrawler performs the SQL Query to receive the list of not processed web pages containing job offers fulfilled by RIDAR tool (step 2 in the Fig. 3). ERID tool is used to evaluate relevance of internet documents and enables focused crawling and downloading implemented in WebCrawler (step 3 in the Fig. 2). WebCrawler saves downloaded document in the file part of the corporate memory (step 4 in the Fig. 2). The sequence of actions performed by RIDAR, WebCrawler and ERID tools is described in the following figure.

[11] *http://code.google.com/*
[12] *http://developer.yahoo.com/*
[13] *http://www.alltheweb.com/*

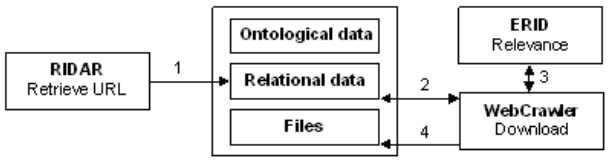

Fig. 3. Integration of RIDAR, ERID and WebCrawler tools

3.2.1 Methods used by ERID tool

A method used in the tool for relevance evaluation analyzes the text content of HTML page and evaluates its relevance according to the desired domain. This method utilizes a theory of neural networks and data analysis based on clustering [4] [5]. The aim of designed method is to decide whether a processed HTML page is relevant to the particular domain. The method of the relevance estimation utilizes ability of neurons to process input signals and define document's relevance using threshold value.
A method for choosing appropriate keywords candidates is designed and applied as well. These keywords define a merit of an input vector and determine the particular domain. A set of training data, which is based on the set of sorted HTML documents, is essential for selection of suitable keywords. During this selection, words located in documents are sorted according to the difference between occurrence frequency in documents, which belong to the particular domain and documents which do not.

3.2.2 Methods used by WebCrawler tool

WebCrawler uses web crawling method (also known as web spidering method) for automatic focused downloading (focused crawling) of pages on the Internet. Implemented crawling method, known as focused crawling, is based on content relevancy evaluation performed by ERID tool using keywords [6]. WebCrawler tool processes defined set of pages and searches for hypertext links leading to other resources which are then processed in next iteration. The stopping condition must be also specified. Pages are downloaded and stored in cache located on a local server for further processing by tools in acquisition chain. The method requires maintaining the list of processed links.
Process of web crawling can be influenced in many ways. The following factors are most significant: selection, re-access, moral rules, parallel downloading.
Selection: There are various possibilities how to choose the next not processed web page. Very popular are breadth-first searching, backlink-count and Page Rank algorithms. WebCrawler tool implements breadth-first search with relevancy evaluation (focused crawling).
Re-access: Due to dynamic characteristics of the web it is efficient to define time limits of re-access, redownload of the webpage, respectively. The following two factors are considered: actuality and age of a page. The downloader's task is to maintain actuality of the set of pages as high as possible. Two reaccessing methods can be considered: uniform and proportional approach. Uniform method downloads every expired page. Proportional approach downloads only pages changed with the highest frequency. The designed algorithm implements uniform approach.
Moral rules: Common downloader is able to download pages faster than human, thus server hosting processed web site can be overloaded during not controlled downloading. Downloaders have to implement various approaches to avoid server overloading, such as delayed download, or list of pages which should not be downloaded (robot exclusion protocol). Server overloading can be controlled with appropriate selection strategy in parallel downloading mechanism.
Parallel downloading: Aim of parallel downloading is to maximize speed of downloading and processing several web sites concurrently. To avoid processing one page by more threads of the downloader, a page assigning strategy to downloader threads has to be designed and implemented in the downloader. Static approach assigns pages to threads at the beginning of downloading process. On the other hand, dynamic approach defines page assignments during downloading process.

3.3 ExPoS/OSID tools

All methods for job offer extraction are bounded into one tool located in acquisition chain - ExPoS. The main tool objective is to extract useful information (job offer) from the content of the page downloaded and stored in the corporate memory by WebCrawler tool (step 5 in the Fig. 4). Extracted information is stored in a webpage with original structure. Afterwards all filtered pages are stored in the corporate memory (step 6 in the Fig. 4) and processed by semantic annotation tool. Fig. 4 shows integration of ExPoS and OSID tools with corporate memory.

3.3.1 Methods used by ExPos/OSID tools

This experimental tool encompasses the following four methods utilizing different approach how to identify useful information: Recursive Substring Analysis (RSA), linear substring analysis (LSA), Common Header and

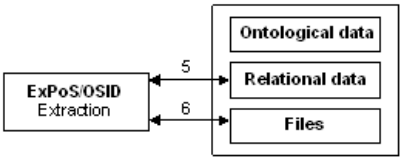

Fig. 4. ExPoS/OSID tools integration with corporate memory

Footer Recognition (CHFR), and Genetic Pattern Recognition (GPR). RSA is a method based on recursive division of a plain text into a substring and counting frequency of its occurrence ina set of pages with the same structure. Patterns with high frequency are removed from the page as an undesirable content. This method is systematic with high complexity:

$$\left\lfloor \int_0^{\log_2 \frac{n}{l}} 2^x dx \right\rfloor = \left\lfloor \frac{1}{\ln 2} \left(\frac{n}{l} - 1 \right) \right\rfloor$$

where n is a number of words in text and l is a minimal length of patterns.

LSA is a modification of RSA method which processes plain text rather in linear than in recursive manner. This modification rapidly reduces method's complexity n-l, where l is a size of a pattern.

CHFR method is based on assumption of document's vertical structure, i.e. each document from same source has common footer and header. The method is based on some statistical method, which is robust to small changes in headers and footers. Advantage of this method is its low complexity; however method cannot identify patterns inside the document. Thus, this method can be used as pre-filtering method for RSA and LSA, which reduces information space.

GPR searches for a pattern utilizing genetic algorithms [7] [8], where each possible solution is evaluated by fitness function. Each possible solution is represented by chromosome, which consists of a string of genes, while each gene is one word in a document. The order of genes is the same as order of words in the document. If the gene is a part of a potential pattern, its value is 1; otherwise 0. A fitness function used in the genetic algorithm minimizes difference between an average frequency of patterns and a number of documents, and maximizes a size of patterns. The genetic algorithm utilizes a library GAJIT, which uses selection commensurable to quality [9]. The result of this method for the case, when the first 100 words of a document are considered, is following: the quality of the method is 72%. The genetic search is oriented on points in the search space, however the algorithm can be transformed into searching for regions by utilizing of special operators [10].

3.4 DocConverter tool

DocConverter tool is the next tool in our chain and is responsible for converting documents preprocessed by ExPoS/OSID tools to the text form. The main purpose of the tool is to perform a batch conversion of documents and generate meta information about the documents in the corporate memory [3] (step 7 in the Fig. 5). The tool then saves the job offers to the file part of corporate memory again (step 8 in the Fig. 5). DocConverter tool is also integrated with NALIT (step 9 in the Fig. 5) tool, which is responsible for language identification. The meta information about the document, e.g. document's language, is stored in the relational part of the corporate memory and is further used by successive tools in the chain (e.g. RFTS, OnTeA). Fig. 5 shows integration of NALIT and DocConverter tools and communication between corporate memory and the tools.

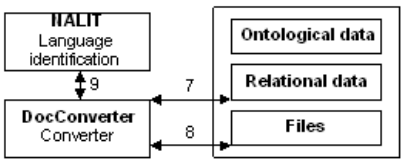

Fig. 5. Integration of NALIT and DocConverter tools

3.4.1 Methods used by DocConverter tool

Main method of DocConverter tool integrates some external tools for conversion from different formats to the text format. It is possible to define supported formats and also external tools and running parameters for conversion of these formats.

3.5 RFTS (Rich Full-Text Search) tool

RFTS tool is responsible for indexing the documents stored in the file part of the corporate memory [3]. The motivation for implementing another search engine was to have an easily expendable and configurable document indexing tool to evaluate novel methods for information retrieval, documents statistical analysis and lemmatization and stemming methods for Slovak language. This tool is linked with Tvaroslovnk tool (step 12 in the Fig. 6), which is responsible for lemmatization of indexing words.

3.5.1 Methods used by RFTS tool

The tool's indexing part consists of two consecutive phases:

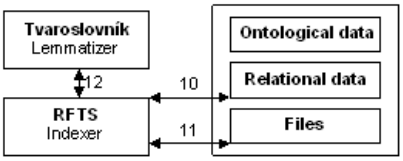

Fig. 6. Integration between RFTS, Tvaroslovnk and corporate memory

1. text preprocessing
2. indexing

In text preprocessing phase, the input document is split to tokens based on defined set of delimiters and the tokens are converted to their word base form. The stemming and lemmatization methods are considered for word base form identification. The tool selects appropriate word base form module based on the language of the document. Adapters for following tools for word base form identification are currently implemented:

- WordNet adapter (WordNet [11] [12] is a large lexical database of English language - it provides also the lemmas of the words stored in the database),
- Porter Stemming Algorithm (popular stemming algorithm for English language),
- Tvaroslovník (lemmatizer for Slovak language developed within NAZOU project),
- SKLemmatizer (lemmatizer for Slovak language developed at Ľ. Štúr Institute of Linguistics).

The indexing phase matches the tokens in base form to the dictionary and stores the information about documents words in the relational database. The indexes are constructed following the boolean model of information retrieval systems. In addition to the term membership within the document, the information about word positions and phrase number are stored in index structures.
In the query processing phase, the query submitted to RFTS is translated to SQL query suitable to retrieve requested information from back-end relational database. The SQL query is generated based on the query type and input parameters for the query type. Boolean queries with AND and OR operators are implemented as well as search for relative distance of given terms. RFTS provides plug-in style mechanism for custom query type definition. This feature proved of high usefulness for integration of RFTS with other tools (e.g. Ontea, which requires special query types providing additional statistical information about the term occurrences in the document collection).

3.6 RDB2Onto tool

The RDB2Onto (Relational Database Data to Ontology Individuals Mapping) tool is next tool in the tool chain. This tool is responsible for replication data from relational database (step 13 in the Fig. 7) (e.g. language of the job offer, relative path to the offer source) to the ontological model (step 14 in the Fig. 7). This tool only creates empty instances of job offers in the domain ontology and the next tool in the chain (OnTeA) fulfills these instances.

3.6.1 Methods used by RDB2Onto tool

The tool works on a domain ontology model and a relational database. The overall idea is to map SQL query to RDF/OWL XML template. Such OWL data are then sent to an ontology model. The tool is implemented in Java using Sesame[14] library for ontology manipulation and MySQL[15] database for testing but it is possible to use any other relational database using JDBC[16] connector. It performs three basic steps:

- SQL query, for example:

```
SELECT id, url, original_doc_path, converted_doc_path,
download_date, IF(lang = 'sk', 'Slovak', 'English')
AS lang FROM document
```

- The SQL query is executed and for each row of the query results it fills in the XML-based OWL template (see below). Each element enclosed with brackets is replaced with adequate value from SQL query for a given row.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
xmlns:jo="http://nazou.fiit.stuba.sk/nazou/ontologies/
--v0.6.17/offer-job#"
   xmlns:inst="http://nazou.fiit.stuba.sk/nazou/ontologies/
--v0.6.17/offer-job-inst#"
   xmlns:ofr="http://nazou.fiit.stuba.sk/nazou/ontologies/
--v0.6.17/offer#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
   <rdf:Description rdf:about="offer-job-inst:jo_{id}">
     <rdf:type rdf:resource="offer-job:JobOffer"/>
     <ofr:hasSource rdf:resource="offer-job-inst:source_{id}
----"/>
     <ofr:hasOfferCreator rdf:resource="offer-job-inst:
----OfferCreator_NAZOU_RDB2Onto"/>
   </rdf:Description>
   <rdf:Description rdf:about="offer-job-inst:source_{id}">
     <rdf:type rdf:resource="offer:OfferSource"/>
     <ofr:acquisitionDate>{download_date}
     </ofr:acquisitionDate>
     <ofr:originalURI>{url}</ofr:originalURI>
     <ofr:localURI>{original_doc_path}</ofr:localURI>
     <ofr:localConvertedURI>{converted_doc_path}
     </ofr:localConvertedURI>
     <ofr:language rdf:resource="region:{lang}"/>
   </rdf:Description>
</rdf:RDF>
```

- Composed OWL data are stored to the ontology part of the Corporate Memory.

[14] *http://www.openrdf.org/*
[15] *http://www.mysql.com/*
[16] *http://java.sun.com/javase/technologies/database/*

3.7 OnTeA tool

The last tool in the chain, OnTeA (Ontology based Text Annotation) tool, analyzes a document or a text using regular expression patterns and detects equivalent semantics elements according to the defined domain ontology. Several cross application patterns are defined but in order to achieve good results, new patterns need to be defined for each application. In addition, OnTeA creates a new ontology individual (step 19 in the Fig. 7) of a defined class and assigns detected ontology elements/individuals as properties of the defined ontology class. It is integrated with RFTS tool (step 18 in the Fig. 7) (used for relevancy) and Tvaroslovnk tool (step 17 in the Fig. 7).

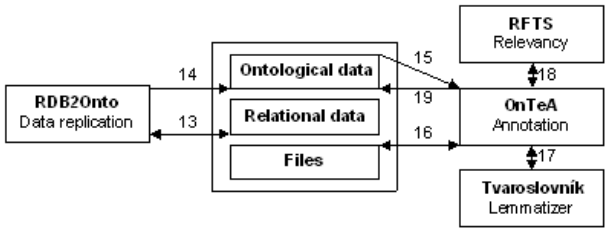

Fig. 7. Integration of RDB2Onto, RFTS, OnTeA and Tvaroslovník tools

3.7.1 Methods used by OnTeA tool

OnTeA's method of automatic annotation based on regular expressions matching showed promising results for domain specific texts. However it suffers from frequent mismatching which leads to creation of imprecise instances of ontological concepts. We propose to overcome this obstacle by evaluating the relevance of candidate instances by the means of statistical analysis of the occurrence of the matched words in the document collection. Based on regular expression, OnTeA identifies part of a text related to semantic context and matches the subsequent sequence of characters to create an instance of the concept. Let us denote the sequence of words related to semantic context as C and word sequence identified as a candidate instance as I. We evaluate the relevance of the new instance by computing the ratio of the close occurrence of C and I and occurrence of I:

$$\frac{close\_occurrence(C, I)}{occurrence(I)}$$

RFST indexing tool provides us with enough functionality to retrieve required statistical values computed from the whole collection of documents stored in RFTS index structures.
OnTeA tools works in these seven steps:

1. The text of a document is loaded.
2. The text is proceed by defined regular expressions and if they are found, corresponding ontology individual according to rest of pattern properties is added to a set of found ontology individuals.
3. If no individual was found for matched pattern and createInstance property is set, a simple individual of the class type contained in the hasClass property is created with only property rdf:label containing matched text.
4. Such process is repeated for all regular expressions and the result is a set of found individuals.
5. An empty individual of the class representing proceed text is created and all possible properties of such ontology class are detected from the class definition.
6. The detected individual is compared with the property type and if the property type is the same as the individual type (class), such individual is assigned as this property.
7. Such comparison is done for all properties of a new individual corresponding to the text/document as well as for all detected individuals.

4 Conclusion

The paper describes the basic tools integrated via the corporate memory that are used for acquisition, organization and partial maintenance of information from heterogenous and distributed resources. The tools process various types of data and information and that is why, the data is transformed from one type to another. The tools process various type of files, documents, idexes, data from database and ontological data. By the end of the data transformation chain are data that can be used for presentation to the users.

References

1. Page L., Brin S., Motwani R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. Manuscript in progress. http://google.stanford.edu/~backrub/pageranksub.ps
2. Garfield E.: Citation Analysis as a tool in Journal Evaluation; Essays of an Information Scientist, Vol1, p **527-544**, 1962 - 73, Reoprinted form Science, (178): **471-479**, 1972, http://www.garfield.library.upenn.edu/essays/V1p527y1962-73.pdf (available on August 2007)
3. M. Ciglan, M. Babik, M. Laclavik, I. Budinska, L. Hluchy: Corporate memory: A framework for supporting tools for acquisition, organization and maintenance of information and knowledge. In J. Zendulka, editor,

Proc. of 9th Int. Conf. on Information Systems Implementation and Modelling (ISIM 2006), MARQ, Ostrava, 2006, p **185-192**

4. Ng H.T., Goh W.B., Low K.L.: Feature selection, perceptron learning, and a usability case study for text categorization. In: Belkin Nicholas, Desai Narasimhalu and Willett Peter Eds., Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia, PA, July 1997, p **67-73**
5. Wiener E., Pedersen J.O., Weigend A.S.: A Neural Network Approach to Topic Spotting. In: Proceedings of SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval 1995, p **317-332**
6. Chakrabarti S., M. van den Berg, Dom B: Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. In: Proceedings of The Eighth International World Wide Web Conference, Toronto, Canada 1999, http://www8.org/w8-papers/5a-search-query/crawling/index.html
7. Lažanský, J.: Evoluční výpočetní techniky. In: Umělá inteligence 3, ACADEMIA, Prague, Czech Republic 2001, ISBN 80-200-0472-6
8. Sekaj, I.: Prednášky k predmetu Fuzzy a neurónové regulátory. Katedra automatizovaných systémov riadenia, Fakulta Elektrotechniky a Informatiky, Slovenská Technická Univerzita 2003
9. Lažanský, J., Kubalík, J.: Genetické programování a vybrané problémy evolučních výpočtú. In: Umělá Inteligence 4, ACADEMIA, Prague, Czech Republic 2003, ISBN 80-200-1044-0
10. Brown, E. C., Sumichrast, R. T.: Evaluating performance advantages of grouping genetic algorithms. In: Engineering Applications of artificial Intelligence, Volume 18, ELSEVIER 2005, p **1-12**
11. WordNet http://wordnet.princeton.edu/
12. Miller, G.: WordNet: An On-line Lexical Database, Special Issue, International Journal of Lexicography, Vol. 3, Num. 4, 1990