Intelligent System Structure for Web Resources Processing and Analysis

Vasyl Lytvyn[1], Victoria Vysotska[2], Lyubomyr Chyrun[3], Andrzej Smolarz[4], Oleh Naum[5]

[1,2] Information Systems and Network Department, Lviv Polytechnic National University, Bandery str., 12, Lviv, Ukraine, 79013

vasyl.v.lytvyn@lpnu.ua[1], victoria.a.vysotska@lpnu.ua[2]

[3]Computer-Aided Design Department, Lviv Polytechnic National University, Bandery str., 12, Lviv, Ukraine, 79013

chyrunlv@mail.ru[3]

[4]Institute of Electronics and Information Technology, Lublin University of Technology, Nadbystrzycka str., 38A, Lublin, Poland, 20618

smolan64@gmail.com[4]

[5]Information Systems and Technologies Department, Drohobych Ivan Franko State Pedagogical University, I. Franko str., 24, Drohobych, Ukraine, 82100.

oleh.naum@gmail.com[5]

Abstract. The paper describes the general detailed and formal description of intelligent system of information resources processing (ISIRP) based ontology. The content life cycle phase implementation of ISIRP structure is improved. The general principles of ISIRP designing structures enable automated information resource processing to increase regular user text content realization, reducing the production cycle, saving time and increasing the e-commerce capabilities.

Keywords: content analysis, information resources, rating evaluation, content management system, ontology, knowledge base, machine learning, intelligent agent, stemming, parser.

1 Introduction and review of current research on the topic

Information resource in ISIRP – data set with a property set (Tab. 1), which is the action object of IT content transformation. The result of one IT usage may be

information resource of another [1]. Content in IT is formalized information and knowledge, placed in IS environment and, unlike data without detailed properties specification, formalization methods and regulation. Transformation of heterogeneous data in an organic centralized information resource is one of the most important problems of ISIRP construction and operation. The procedure of information resources formation and usage in ISIRP (Fig. 1) are determined by primary source data select method, data fixation, filtering, conversion to a specified format to create content and placement in the database [2].

Table 1. The main properties of information resources in ISIRP

Name	Property
Heterogeneity	Components of different origin, content and format of presentation.
Consistency	Absence of conflicting or converse content values.
Format accessibility	Accessibility for all users on the basis of standardized methods, tools and interfaces.
Openness	The ability to interact, exchange and share values with external resources.
Dynamism	Quick update under the terms of a system or environment.
Scalability	Ability to change the logical / physical volume of content (values / concepts and their designations).
Manageability	Changes identification /content usage and its implications on the IS processes.

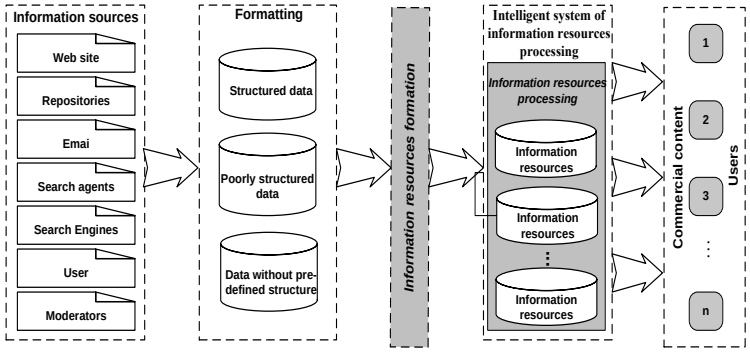

Fig. 1. The procedure for information resources formation and usage in ISIRP

Suppose there is some pre-defined set of content n_X primary sources $Sourse(x_i)$ with fixed or variable composition, where x_i is *i*-content from a source at $i = \overline{1, n_X}$, generates a certain set of values containing information / knowledge / facts from the domain ISIRP (Tab. 2).

Table 2. Objectives and tasks of research

Objective	Scientific innovation	Conclusion compliance
perform ISIRP analysis and evaluation based on compatibility option detalization of these systems to improve their classification;	improved ISIRP classification based on theoretical studies and compatibility option detalization of these systems by analyzing features of the system creation and usage.	studied and improved ISIRP classification based on the analysis and evaluation of such systems, allowing us to determine, detail and justify choice of their compatibility options.
develop a text content forming method through lifecycle improvement for management requirement determining of content flow;	for the first time developed the method of text content formation by its life cycle stages improvement through information resources detailed study.	for the first time developed content formation method by its lifecycle stages improvement for requirement determining of content flow control.
improve the text content control method on the basis of its system formation and analysis for determining text content control parameters;	further developed methods for text content managing on the basis of its system formation and analysis, ensuring control parameter regulation and requirements of content formation.	improved text content control method on the basis of its system formation and analysis for determining text content control parameters.
develop a method of content tracking based on statistical analysis of ISIRP to change the control parameters and the content forming requirements ;	for the first time developed a method of text content tracking based on statistical analysis of ISIRP allowing us to determine the content managing parameters.	developed a method of text content tracking based on statistical analysis of ISIRP to change the control parameters and content forming requirements which makes it possible to increase turnover volumes of content at 9%.
improve the structure of ISIRP by analyzing the information resources processing to develop typical system design recommendations;	improved the structure of ISIRP based on information resources processing specification and through the sharing of the formation processes, management and content tracking, ensuring the implementation its life cycle stages and development of typical system design recommendations;	improved the structure of ISIRP based on information resources processing different than the current with subsystems for formation, management and text content tracking; elaborated recommendations for ISIRP structure design
carry out results evaluation through the implementation of information resources processing technological software in ISIRP to reduce the time and cost of the textual content formation, management, and tracking .	elaborated recommendations for ISIRP structure design different than the current for stage specification and information resources processing subsystems existence that enable content lifecycle support; developed and implemented software for textual content formation, management, and tracking to increase the constant user text content volume at 9%.	developed and implemented software for textual content formation, management, and tracking to increase the potential user active attraction and the target audience expansion by 11%.

2 Information flows in intelligent system of information resources processing

As a conversion result of ISIRP certain technological means to the source $Sourse(x_i)$ is generating a range set $X = \{x_1, x_2, ..., x_{n_X}\}$ through Web-source information parser, perceived and presented figurate. In the process of selection and fixation of generated values according to the technological features of the system, every source of information generated range set is converted into input content set $C = \alpha(u_f, x_i, t_p)$ of defined format c_r, where $r = \overline{1, n_C}$. The main objective of the ISIRP development project is creating information resource information architecture by creating up to date text content that is formed on the reverse reaction of the users according to the type of content distribution (Fig. 2). Each content set is presented in a structured, semi structured data or data without description of the structure and stored in a text content database. Content structuring involves the formation for each set describing its composition, methods of combining elements and their regulation, i.e. condition set $U = \{u_1, u_2, ..., u_{n_U}\}$, where u_j is content formation condition at $f = \overline{1, n_U}$. Source data set is a combination of set values in a given format and condition set $\langle X, U \rangle$ when forming a content input set without structure description $U = \varnothing$.

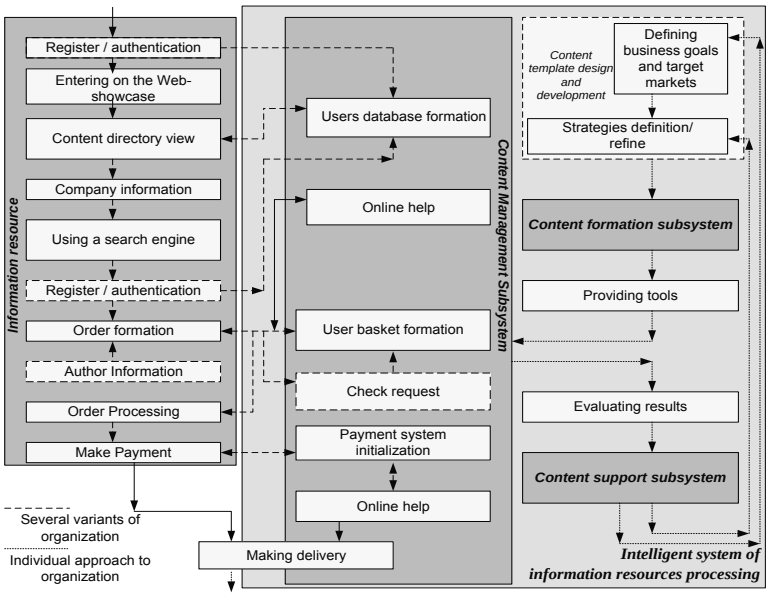

Fig. 2. Data flow diagram in ISIRP

The resulting content before saving is verified / validation for its formal / substantial accuracy / relevancy confirmation upon the system demand. In case of inconsistency to required criteria piece of content is removed from further use. Filtered content is formatted and stored, then the relevant information and knowledge

$\langle C,H\rangle$ become available to users through information resource ISIRP, i.e. $Sourse(x_i) \rightarrow x_i \in X \rightarrow X \rightarrow \langle X,U\rangle \rightarrow \alpha(u_f,x_i,t_p) \rightarrow c_r \rightarrow C \rightarrow DataBase(C) \rightarrow \beta(q_d,c_r,h_k,t_p) \rightarrow \langle C,H\rangle$, where $i=\overline{1,n_X}$, where n_X – content source number, $Sourse(x_i)$ - i-content source, $x_i \in X$ - i-source content, $Sourse(x_i)$; $X=\{x_1,x_2,...,x_{n_X}\}$ - data set as a result of the source selection, $Sourse(x_i)$; $\langle X_i,U_i\rangle$ – data set with condition set, $\alpha(u_f,x_i,t_p)$ – forming content operator, c_r – formed content, C – generated content set, $DataBase(C)$ – content preserving statement in the database, $\beta(q_d,c_r,h_k,t_p)$ – content control statement, $\langle C,H\rangle$ – ISIRP information resources composed of text content sets and content steering conditions [1].

The text content formation process submitted on this coupling scheme: $Sourse(x_i) \rightarrow x_i \in X \rightarrow X \rightarrow \langle X,U\rangle \rightarrow \alpha_1(Downloading(\langle X,U\rangle),T) \rightarrow \alpha_2(Verification(\langle X,U\rangle),T) \rightarrow \alpha_3(Conversion(\langle X,U\rangle),T) \rightarrow \alpha_4(\langle X,U\rangle,T) \rightarrow \alpha_5(Qualification(\langle X,U\rangle),T) \rightarrow \alpha_6(\langle X,U\rangle,T) \rightarrow \alpha_7(\langle X,U\rangle,T) \rightarrow c_r \in C$, where $X=\{x_1,x_2\ldots,x_{n_X}\}$ – income data set $x_i \in X$ from different information resources or moderators under $i=\overline{1,n_X}$; α_1 – content collecting statement from different sources, α_2 – content duplication identification statement, α_3 – content formation statement, α_4 – content key words and concepts identification statement [3], α_5 – content automatic categorization statement [4], α_6 – content digest forming statement, α_7 – content selective distribution statement, $T=\{t_1,t_2\ldots,t_{n_T}\}$ – transaction time $t_p \in T$ of text content formation under $p=\overline{1,n_T}$, $C=\{c_1,c_2\ldots,c_{n_C}\}$ – text content set $c_r \in C$ under $r=\overline{1,n_C}$, $Verification(\langle X,U\rangle)$ – content verification statement, $Qualification(\langle X,U\rangle)$ – content qualification statement, $Conversion(\langle X,U\rangle)$ – content transformation statement, $Downloading(\langle X,U\rangle)$ – content uploading statement. Improved ISIRP structure is through the addition of text content technological formation, management and support software (Fig. 3) [1].

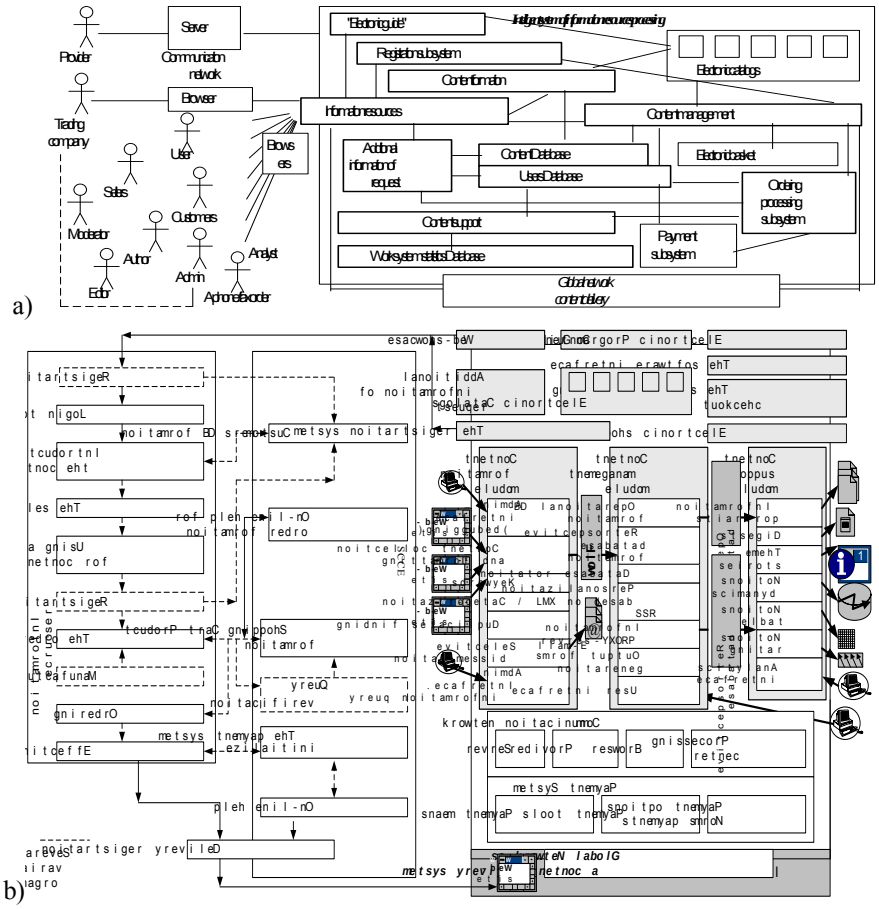

Fig. 3. Improved ISIRP structure

Text content formation - a set of measures to ensure data processing control from various sources (Fig. 4) to create a content with an additional set of values such as relevance, authenticity, uniqueness, completeness, accuracy and so on.

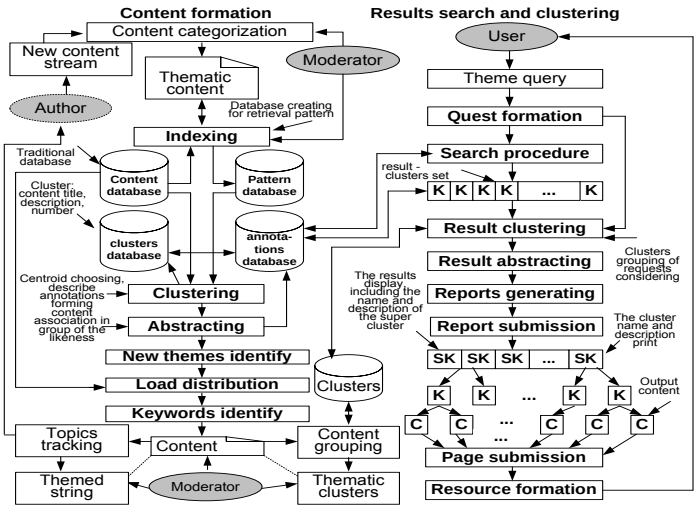

Fig. 4. Usage scheme of annotation database for text content search

Text content managing is a set of measures to provide text content defining parameter value support as topicality, completeness, relevance, authenticity, reliability to determined requirements by a criteria set (Fig. 5). Text content tracking is a set of measures to ensure the ISIRP functioning under certain requirements and any subsequent changes to these requirements (Fig. 6). ISIRP information resources processing allows to get current and objective data about the system operation and for competition level evaluation on the segment of the content financial market; estimate competitor level and measure their competitiveness across the financial landscape for content distribution. The main classes of information resource users/characters (clients, managers and administrators) define the information resource design and decision-making process. ISIRP necessarily includes Web-mart (information resource) with a text content catalog (searchable) and the necessary interface elements to enter registration data, the formation of orders, making payments over the Internet, delivery handling (e-mail / on-line), obtaining data about the company and on-line help. Registration/user authorization happens while making order or entering the system. The interaction is carried out over a secure channel SSL for protection purposes. The whole process is recorded in the content management subsystem for ISIRP functioning statistic formation and offers as a list of popular content topics for content forming subsystem.

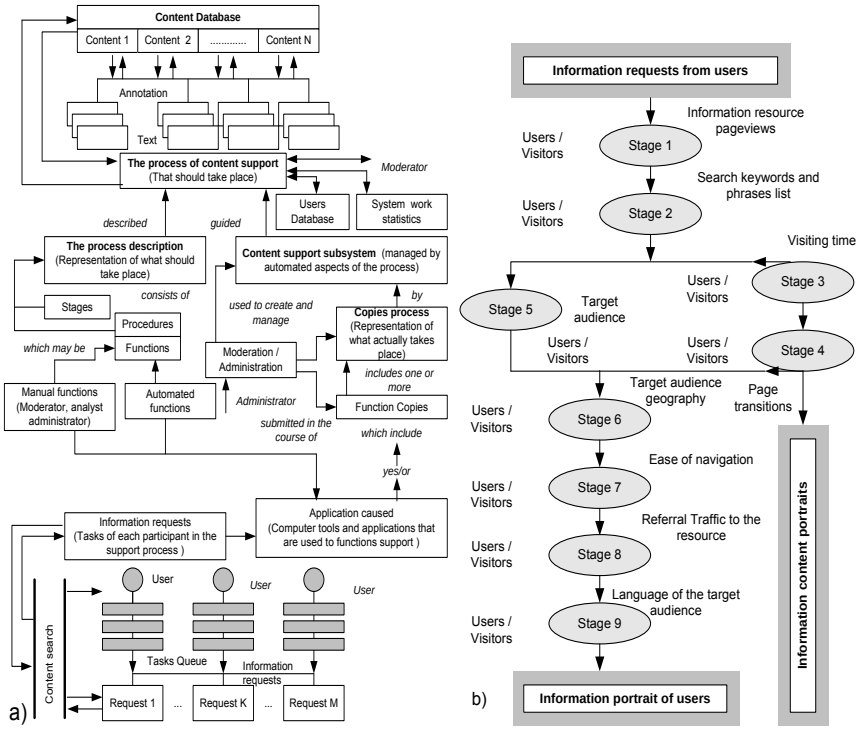

Fig. 5. The dependence scheme of a) component and b) stages of the text content tracking

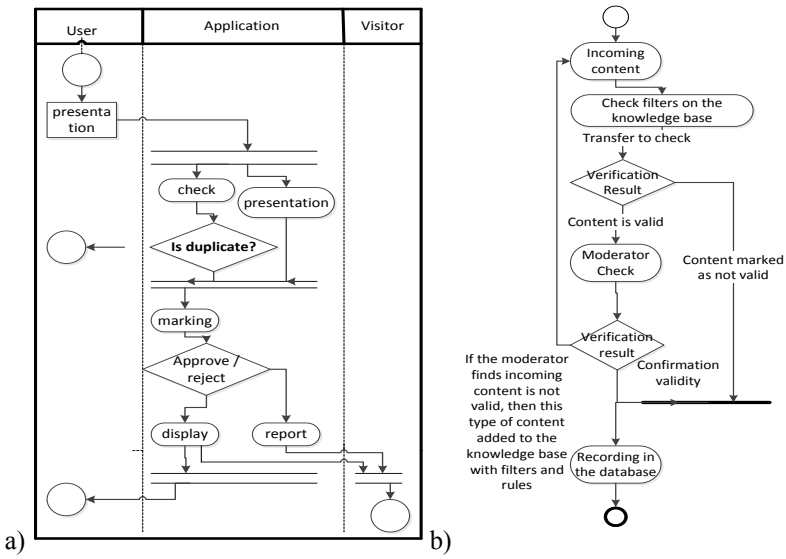

Fig. 6. The detailed scheme of a) tracking and b) a set of measures to ensure data processing control from various sources

3 Method of Web Resources Processing and Analysis

The process of content formation executes a transformation between a set of input data from different sources and a set of formatted and saved content elements: $S(x_i) \rightarrow x_i \rightarrow X \rightarrow \alpha(u_f, x_i, t_p) \rightarrow c_r \rightarrow C \rightarrow D(C)$, where $S(x_i)$ – is a source of data, $D(C)$ – content database. The formation of content $\alpha : X \rightarrow C$ is presented as a superposition of functions

$$\alpha = \alpha_7 \circ \alpha_6 \circ \alpha_5 \circ \alpha_4 \circ \alpha_3 \circ \alpha_2 \circ \alpha_0, \text{ or } \alpha = \alpha_7 \circ \alpha_6 \circ \alpha_5 \circ \alpha_4 \circ \alpha_3 \circ \alpha_2 \circ \alpha_1, \quad (1)$$

where α_0 – is an operator of content creation; α_1 – operator of gathering content from different sources through Web-source information parser; α_2 – operator of content deduplication; α_3 – operator of content formatting; α_4 – operator of keywords and concepts elucidation; α_5 – operator of automatic categorization; α_6 – operator of compilation of content digests; α_7 – operator of discretionary content sharing. The process of content formation is presented as

$$\alpha = \langle X, T, U, C, \alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7 \rangle. \quad (2)$$

1. The operator for content creation is a mapping of input data from various sources into content, which is actual and commercially worthy $\alpha_0 : (X, U_C, T) \rightarrow C_0$.

2. The operator of content gathering through Web-source information parser is a mapping between input data obtained from authors or moderators into content which is actual and authentic $\alpha_1 : (X, U_G, T) \rightarrow C_0$.

3. The operator of content deduplication is a mapping of initial content into content having no duplicate elements $\alpha_2 : (C_0, T, U_B) \rightarrow C_1$.

4. The operator of content formatting is a changing of content's format $\alpha_3 : (C_1, U_{FR}, T) \rightarrow C_2$.

5. The operator of keywords elucidation define an addition to content in the form of the set of keywords, which generally describe it $\alpha_4 : (C_2, U_K, T) \rightarrow C_3$.

6. The operator of content categorization – is a transformation of content via analysis and validation into a new state where content is assigned to some thematic category $\alpha_5 : (C_3, U_{CT}, T) \rightarrow C_4$.

7. The operator of compilation of digests based on content is a transformation of content to new state having a short content digest $\alpha_6 : (C_4, U_D, T) \rightarrow C_5$.

8. The operator of discretionary sharing of content adds to content a target audience definition and sharing to this audience $\alpha_7 : (C_5, U_{Ds}, T) \rightarrow C_6$.

The process of content formation is described by operator $c_{r+1}(t_{p+1}) = \alpha(c_r, t_p, X, u_f)$, where $u_f = \{u_{1f}, u_{2f}, \ldots, u_{n_U f}\}$ – is a set of conditions for content c_r formation:

$$c_r = \left\{ \bigcup_i^{n_X} x_i \middle| \begin{array}{l} \forall x_i \in X_{u_f}, x_i \notin X_{\overline{u_f}}, \exists u_f \in U_{x_i}, u_f \notin U_{\overline{x_i}}, \\ X = X_{u_f} \cup X_{\overline{u_f}}, U = U_{x_i} \cup U_{\overline{x_i}}, f = \overline{1, n_U} \end{array} \right\}.$$

The process is going through data transformation stages into a set of relevant, formatted, categorized and validated content elements: $x_i \in X \rightarrow \alpha_0(X, U_C, T)$ або $\alpha_1(X, U_G, T) \rightarrow \alpha_2(C_0, T, U_B) \rightarrow \alpha_3(C_1, U_{FR}, T) \rightarrow \alpha_4(C_2, U_K, T) \rightarrow \alpha_5(C_3, U_{Ct}, T) \rightarrow \alpha_6(C_4, U_D, T) \rightarrow \alpha_7(C_5, U_{Ds}, T) \rightarrow c_r \in C$.

Keywords discovery in content. Textual content (articles, commentaries, books, etc.) contains a lot of information in natural language, some of which is abstract. The text is presented as a sequence of character units, the basic properties of which are informational, structural and communicative connectivity / integrity that reflects the informational and structural nature of the text. The method of text processing is the linguistic analysis of content. This process splits the text on lexical tokens using finite automata (Fig. 7).

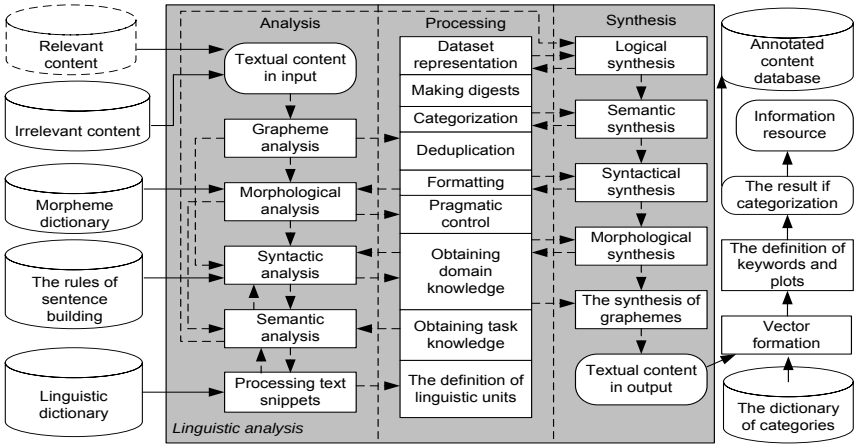

Fig. 7. The structure of linguistic analysis of textual content

The operator of keyword discovery in content is a mapping of content into a new state, which is different from the previous state by the presence of keywords describing this content. During the process of analysis are explored: a multi-layered structure of content; a linear sequence of characters; a linear sequence of morphological structures; a linear sequence of sentences. The discovery of keywords in text snippet is accomplished using such process. On the compositional level the sentences, paragraphs, sections, chapters and pages which are not related to the text snippet are isolated. Therefore, they are not considered. Next, using database of terms

/ morpheme units and text analysis rules, the search of the term is performed. Using the rules of generative grammar, the term is corrected according to the rules of its use in the context. After parsing, the text is drawn into a data structure, such as a tree, which corresponds to the syntactic structure of the input sequence, and is best suited for further processing. After analyzing the text snippet and term, a new term is synthesized as a keyword using a database of terms and their morphemes. Our approach to keyword identification is based on Zipf law and comes to the choice of words with an average frequency of occurrence (most used words, found in stop dictionaries, and rare words are ignored). Keyword detection module is implemented on the website Victana (available at address http://victana.lviv.ua/index.php/kliuchovi-slova).

The process of content categorization. The analysis of the lexical, grammatical and pragmatical structure of a text is used to automatically categorize content and build its digest. The operator of categorization of content is a mapping of the content to a new state via its validation. The new state is different from the previous one by availability of its assignment to some set of contents themes $\alpha_5 : (C_3, U_{CT}, T) \to C_4$. The analysis of content's meaning is performed by the process of pulling grammatical data from the word using grapheme analysis and the correction of morphological analysis results by analyzing the grammatical context of linguistic units. The process of categorization $C_4 = \alpha_5(\alpha_4(C_2, U_K), U_{CT})$ via procedure of automatic content indexing C_3 is divided sequentially into following blocks: morphological analysis, syntactical analysis, semantical and syntactical analysis of linguistic structures, the variation of textual content's written representation.

Creating a digest of content. The digest is a summary of a publication in ISIRP. In order to form it, the weighted content analysis is used and the frequency of words usage from a previously created dictionary of terms is considered. The operator of a digest formation is a mapping of content into a new state. This state is different from a previous one by the availability of additional part (digest) which adds to the content value. The process of a digest formation creates the set of the brief annotations and main points of the content for a specified time. This is convenient for a quick familiarization with the content's basics for a specified subject, and also when doing research on some topic.

Content distribution process. Our implementation of distribution process of content shares a workload between authors and moderators while contributing to increase of the reading audience and the volume of content. In the beginning, the system obtains digests of sources via RSS. Next the digests are distributed among authors according to author's rating. Digests are sent first to authors with higher rating. The rating reflects the performance of each author and is influenced by such criteria as uniqueness of content, the number of views (either direct or referenced), user ratings. The rating assessment system utilizes many criteria, so final result is rather objective and it stimulates authors to improve their performance. After this the system goes into the standby mode until some new content will be added. The workload for moderator is reduced, especially in such tasks as sorting, assessment, rating, evaluation and analysis of content. Therefore, the content is created faster, and has higher quality because of objective process of evaluation of the content.

As a result of analysis of ISIRP functioning S and content support C the set

$Y = \{Y_P, Y_T, Y_C, Y_R\}$ is created according to conditions $V = \{V_P, V_T, V_C, V_R\}$, where $Y_P = Y_{Pc} \vee Y_{Pq}$ - the subset of informational snapshots of content Y_{Pc} and users Y_{Pq}, Y_T - the subset of thematic plots, Y_C - the subset of content dependencies tables, Y_R - the subset of content ratings, $V_P = V_{Pc} \vee V_{Pq}$ - the subset of conditions for information snapshots, V_T - the set of conditions for thematic plots discovery, V_C - the set of conditions for creating content dependencies tables, V_R - the set of parameters used in content rating assessment. The set of informational snapshots of content Y_{Pc} is represented as $Y_{Pc} = BuInfPort(V_{Pc}, C, H, Q, T)$, and the set of user's snapshots Y_{Pq} as $Y_{Pq} = BuInfPort(V_{Pq}, Q, H, Z, T)$, where $V_P = V_{Pc} \vee V_{Pq}$ - the set of conditions for creating snapshots, $BuInfPort$ - the operator of snapshot creation $Y_P = Y_{Pc} \vee Y_{Pq}$.

The set of thematic plots Y_T is presented as $Y_T = IdThemTop(C, H, Q, V_T, T)$, where V_T - is the set of conditions for plot discovery. $IdThemTop$ - the operator of thematic plot discovery Y_T. The set of content dependencies tables Y_C is shown as $Y_C = ConCorrTablConc(C, V_c, T)$, where V_C - the set of conditions for creating content dependencies tables, $ConCorrTablConc$ - the operator for creation of content dependencies tables. The set of ratings Y_{Rc} is represented as $Y_{Rc} = CalRankConc(C, Q, H, Y_C, V_{Rc}, Spam, Tonality, T)$, where $V_R = V_{Rc} \vee V_{Rm}$ - the set of parameters used in rating assessment, $Tonality(Q^+, Q^0, Q^-, T, H)$ - criteria of content tonality, $Spam(Q, T)$ - the operator for comments filtering, $CalRankConc$ - the operator of rating definition for content and moderators $Y_R = Y_{Rc} \vee Y_{Rm}$. The set of output statistical data Y is presented by:

$$Y = \{Y_P, Y_T, Y_C, Y_R\} = Support(V, C, Q, H, Z, T, \Delta T)$$
$$Y = \{Y_P, Y_T, Y_C, Y_R\} = Support(V_P, V_T, V_C, V_R, C, Q, H, Z, T, \Delta T)$$

where $Y_P = Y_{Pc} \vee Y_{Pq}$ is the set of informational snapshots of content and its users, Y_T - the set of thematic plots, Y_C - the set of content dependencies tables, $Y_R = Y_{Rc} \vee Y_{Rm}$ - the set of ratings for content and moderators, $Support$ - the operator for content support.

4 Basic idea and structure of web-source information parser

The use of standard software libraries avoids unjustified time, financial and human resources overrun for their re-development. That is why a wide range of active similar to developed one projects were analyzed, most of which are based on open source code concept and software distribution on terms of free licensing. Leading development team provide their projects by means of API (Application Programming Interface), thanks to which the functionality of these projects can be effectively used

by simple cataloged and well documented procedures and functions with corresponding arguments. Software development community worked out application software package use policies under different license provisions as well as participation in existing projects promotion so that each developer may receive, establish for personal use and develop as possible in one's direction these projects or included software environment libraries. Internet portals such as SourceForge.net contain all the necessary toolkit range o for deployment, documentation and project maintaining of arbitrary degree of complexity, development stage, access level and popularity among users. Developers make heavy use of development version-control special-purpose servers which provide collective (though thousands of participants) software development. The most popular among these is Git server. It can be installed separately as an individual or corporate server, and as well can be used by the global public Git-server GitHub. As studies have shown, among programming languages the vast majority of developments in the field of human language text document processing and almost all developments the field of ontology construction and education are written on Java. In addition, Java retains dominance among project languages, placed on the website SourceForge (Fig. 8) [5].

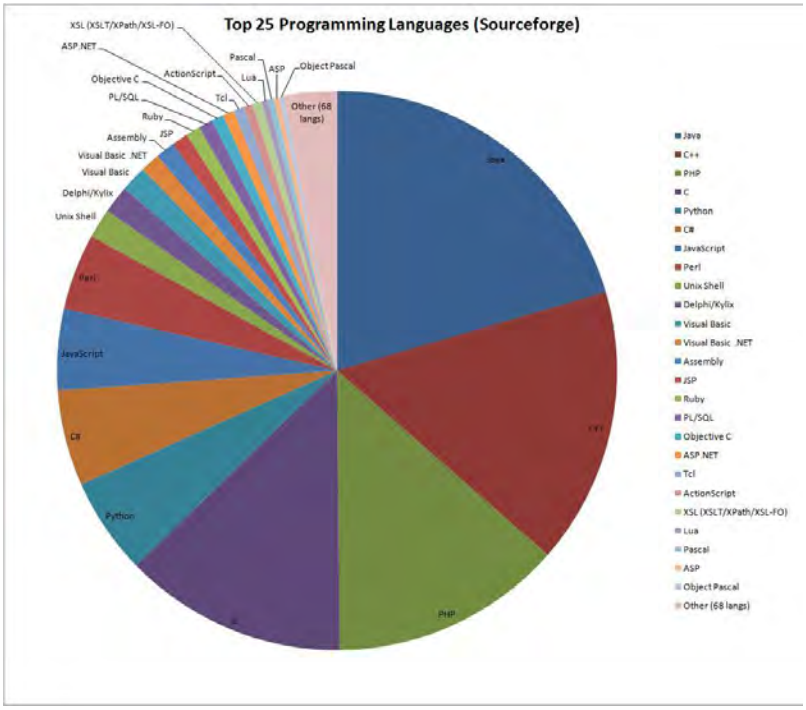

Fig. 8. The shares of 25 most popular programming language among developers that provide access to their portal projects through SourceForge.net.

The winning argument in favor of Java as a project programming language was the availability and accessibility of Java API in a projects at Stanford University (USA) Protege-OWL, as it was Stanford Research Center of Biomedical Informatics,

who has become the practical studies flagship in the field of development tools, knowledge base and ontology editing and teaching with knowledge representation language OWL [6-19]. Projects developed in Java as well:

- Gate [http://gate.ac.uk/] -set of text document processing tools to identify new knowledge.
- owlapi.sourceforge.net - another Java-project, which is a library of Java-classes with broad functionality of OWL- document processing.
- Pellet [http://clarkparsia.com/pellet/] - a software tool - the Java inference machine to implement arguments (knowledge creation) from knowledge base in OWL 2.0 language.

Since there was no unified system for online content search that would bear the value and novelty, the goal was to create one. In other words to implement a system which would use user key words and return only relevant content as result for further processing. The value of such a system lies in scholar labor saving during search of necessary material or document that clearly increases his productivity.

In the service of the aim the following technical challenges were placed [7]:

- Creation of unified information search system, which includes relevant content;
- The system must be implemented as a program (independent module), which should be written in a modern programming language;
- The program must include components for information web source communication (scientific web resources, libraries, archives, data storage, etc.);
- The program must include a component that gives web page content;
- The program must include a component which provides the article content reading and processing and related information from a remote Web resource;
- The program should be built on such a design pattern that allows to expand its capabilities without significant changes of the existing code base;
- The program must include technical documentation that gives an insight into operating principle and simplify further development;
- The program must meet the basic criteria of the Basics of Occupational Safety and general accepted development standards;
- Program should be cross-platform and contain minimum dependencies on third-party libraries of chosen programming language.

During the system creation process the syntactic analysis mechanism of Java integrated web page content was evaluated. There are two such mechanisms or rather two interfaces: DOM (Document Object Model) and SAX (Simple API for XML). When employing DOM document, wherein analysis is necessary, it is systematized in tree-type (the tree hierarchy).The elements of such hierarchy are conveniently accessed and processed. Also DOM element search possesses fast response, but the interface mechanism requires more memory than SAX.

SAX search for relevant information takes place with help of iterative method in other words enumerative technique of all elements in document from the first to the last. That is, per one iteration loop only one element is given for processing and it is possible to refer to it only when iteration reaches the last element of the document and start a new cycle. SAX mechanism has a much lower response speed than DOM, but uses less memory. So, for the system implementation the DOM mechanism was selected as it is much faster to refer to necessary elements in document if they are

classified in the DOM model. There are a lot of implementations and add-ons for DOM. Main of them are: Jericho HTML Parser; Java XML Parser; JTidy; HTML Parser. Within this set of analyzers Jericho HTML Parser was chosen, as its main advantages are:

- Not valid HTML document does not cause errors during the analysis;
- HTML document is being analyzed even if it contains server's tags;
- Option analysis is held with help of StreamedSource class, which allows memory to process large files efficiently. It provides additional functions that are not available in other on-stream analyzers;
- The row and column number of each position in the original document are easily available.

Jericho HTML Parser is a Java open source library, distributed on two licenses: Eclipse Public License (EPL) and the GNU Lesser General Public License (LGPL). So you can use it for commercial purposes in accordance with the license agreement conditions. Javadocs contain comprehensive information on the entire API.

Classes, methods and fields used to create a system:

- Source is class, the object of which contains HTML document.
- fullSequentialParse () is class method Source, which analyzes the output tags.
- Segment - class, the object of which parses the Source object content into segments.
- getAllElements (StartTagType) is class method Segment, which creates a DOM source document hierarchy according to specified conditions.
- getTextExtractor () is method, which returns the clean text and removes all tags in a given document.
- HTMLElementName is a class that contains static methods for choosing right tags.
- getAttributeValue (String attributeName) is method, which returns the decoded attribute value with the specified name.
- getParentElement () is method, which returns the parent towards given in the DOM hierarchy.
- getName () is method, which returns the name of the element.
- getChildElements () is returns a list of the direct descendants of this element in the document hierarchy.

The system consists of three main modules (interfaces) and auxiliary interfaces that provide iteration procedure (Fig. 9a). These modules are: iConnectionProvider, iWebInfoParser, iWebInfoSource. Auxiliary interfaces that provide iteration: Iterable <Publication>, Iterable <String>. IConnectionProvider module is an interface, which is implemented through StraightConnection or ProxyConnection classes depending on chosen connection option (direct inclusion or inclusion with server authentication).

StraightConnection class includes two major public method that implement the iConnectionProvider interface: connect (URL), isConnectible (URL). The method connect (URL) is provides a connection to the web information source. The method isConnectible (URL) is checks for a connection. ProxyConnection class includes fields that contain the data required to make server authentication authorization (Fig. 9b): proxyHost (server authentication name in the form of domain or IP address); proxyPort (server authentication port); proxyUserName (authentication server user login); proxyUserPassword (authentication server user password). All of

these fields are closed due to encapsulation concept. They are accessed using the methods set and get. Also ProxyConnection class includes two major public method that implement the iConnectionProvider interface: connect (URL), isConnectible (URL). The method connect (URL) is provides a connection to the information web source. Method isConnectible (URL) is checks a connection. Also ProxyConnection also includes the following key methods: connectAuthProxy (URL), connectNoAuthProxy (URL), testAuthProxy (), testNoAuthProxy (). IWebInfoSource component consists of classes (Fig. 10): AbstractWebInfoSource, ScienceDirectWIS, CiteSeerXWIS, WileyOnlineLibraryWIS. AbstractWebInfoSource class is an abstract class and implements the interface iWebInfoSource. Classes ScienceDirectWIS, CiteSeerXWIS and WileyOnlineLibraryWIS implement receiving of "raw» not analyzed data from web sources http://www.sciencedirect.com, http://citeseerx.ist.psu.edu and http://onlinelibrary.wiley.com accordingly.

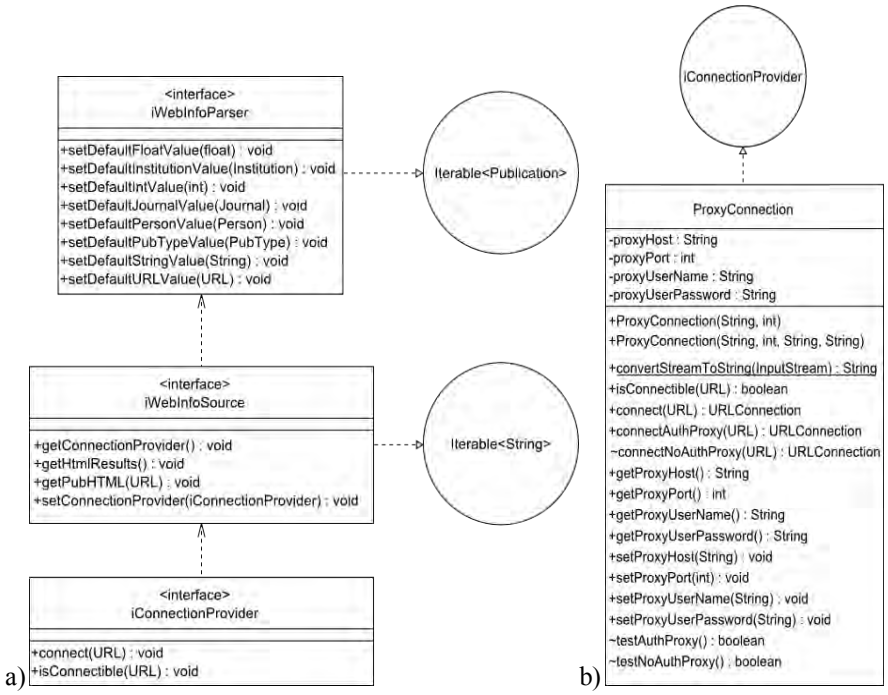

Fig. 9. a) General diagram of the parser structure and b) ProxyConnection UML-class diagram

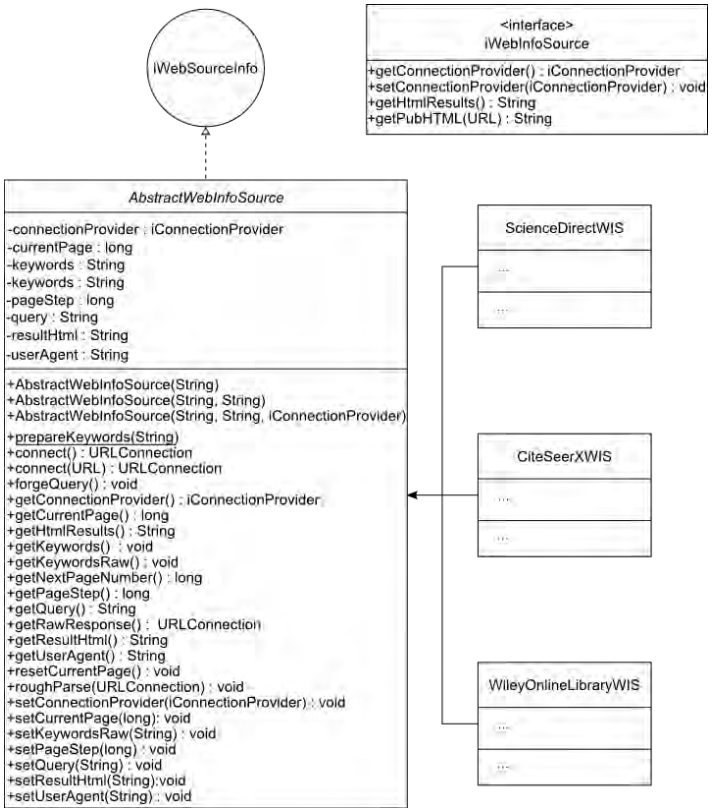

Fig. 10. Schematic diagram of iWebInfoSource component

IWebInfoParser module consists of the following classes: AbstractParser, ScienceDirectParser, CiteSeerXParser, WileyOnlineLibraryParser. AbstractParser class is an abstract class and implements the interface iWebInfoParser.
Classes ScienceDirectParser, CiteSeerXParser WileyOnlineLibraryParser analyzes raw» not analyzed data from component iWebInfoSource.

5 Conclusions

The tremendous growth rates of internet and volume of content stored on its servers requires the creation of tools for automatic content analysis and processing. Intelligent systems for content processing aim to simplify and automate such tasks as content analysis and classification, finding keywords and building digests, distributing content according to specified criteria. This paper presents the results of a study of patterns, characteristics and dependencies in automatic text processing of content. Data and information flows in processes of content transformation are elucidated and formalized. The methods of linguistic analysis are proposed for automation of all operations of content processing. A content management system built using

developed methods is constantly monitoring content from various sources, gathers and integrates content and distributes it to customers. The implementation of proposed methods and procedures allows effectively create and distribute content for targeted social audience and individual customers.

Content forming is implemented as a content-monitoring system collecting the content from various data sources. This enables the creation of a database of content information according to user needs. The result of collecting and primary processing of initial content is new content reduced to a single format, classified according to certain rubricator, and having attributed some descriptors and keywords.

The subsystem of content support provides the formation of information snapshots; the detection of thematic plots; the building of content dependencies tables; the calculation of content rating, the identification of new events in content flows, their tracking and clustering. The analysis of maintenance process of content allows us to determine the causes of the target audience formation using the set of parameters. By adjusting the themed set of content, its uniqueness, its formation efficiency and providing the adequate management according to the individual needs of regular users, the boundaries of targeted social audience and the number of unique visitors from search engines can simulated.

The article describes following results:

- Has been analyzed the development problem of ontology-based information resources processing intellectual systems.
- Has been proposed a new evaluating method of a text document relevance according to the information needs of the information system client, which is based on building information need model in the form of intelligent agent optimal strategy, assessment of expected usefulness and its change due to refinement of the plan by adding information from the investigational document.

The automated ontology synthesis information technology was implemented as software CROCUS, which can be used for proposed method of relevance assessment in human language text information search and knowledge extraction systems.

References

1. Vysotska V., Chyrun L., Lytvyn V., Dosyn D. (2016). Methods based on ontologies for information resources processing : Monograph. LAP Lambert Academic Publishing. Saarbrucken, Germany.
2. Lytvyn V., Pukach P., Bobyk I., Vysotska V. (2016). The method of formation of the status of personality understanding based on the content analysis, *Eastern-European Journal of Enterprise Technologies*, no5/2(83), 4–12.
3. Bisikalo O.V., Vysotska V.A. (2016). Identifying keywords on the basis of content monitoring method in ukrainian texts, *Journal «Radio Electronics, Computer Science, Control»*, No 1, Zaporizhzhya National Technical University, 74-83, Access mode: http://ric.zntu.edu.ua/article/view/66664/0.
4. Lytvyn V., Vysotska V., Veres O., I Rishnyak., and Rishnyak H. (2017). Classification Methods of Text Documents Using Ontology Based Approach, *Advances in Intelligent Systems and Computing* 512, Springer International Publishing AG: 229-240.
5. Ourania Hatzi, Dimitris Vrakas, Nick Bassiliades, (2010). Dimosthenis Anagnostopoulos, and Ioannis Vlahavas. The PORSCE II Framework: Using AI Planning for Automated

Semantic Web Service Composition the Knowledge Engineering Review, Cambridge University Press, Vol. 02:3, 1–24 p. (In English)
6. Lytvyn V. (2013). Design of intelligent decision support systems using ontological approach, *An international quarterly journal on economics in technology, new technologies and modelling processes*, Krakiv-Lviv, Vol. II, No 1, 31 – 38 (In English).
7. Lytvyn V., Dosyn D., Smolarz A. (2013). An ontology based intelligent diagnostic systems of steel corrosion protection, *Elektronika*, Lodzj. – No. 8. – 2-13. – Pp. 22-24 (In English).
8. Lytvyn V. (2011), The similarity metric of scientific papers summaries on the basis of adaptive ontologies , *Proceedings of VIIth International Conference on Perspective Technologies and Methods in MEMS Design*, Polyana, Ukraine, pp. 162. (In English)
9. Link Grammar – Carnegie Mellon University, available at: http://bobo.link.cs.cmu.edu/link.
10. Qiu Ji, Peter Haase, and Guilin Qi (2008). Combination of Similarity Measures in Ontology Matching using the OWA Operator, In Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Base Systems.
11. Gruber T. A. (1993). Translation approach to portable ontologies. Knowledge Acquisition, № 5 (2):199–220.
12. Guarino N. (1995). Formal Ontology, Conceptual Analysis and Knowledge Representation. International Journal of Human-Computer Studies, 43(5-6):625–640.
13. Sowa J. (1992). Conceptual Graphs as a universal knowledge representation. In: Semantic Networks in Artificial Intelligence, Spec. Issue of An International Journal Computers & Mathematics with Applications. (Ed. F. Lehmann), № 2–5:75–95.
14. Montes-y-Gómez M. (2000). Comparison of Conceptual Graphs. Lecture Notes in Artificial Intelligence, Vol. 1793. – Springer-Verlag, Access mode: http://ccc.inaoep.mx/~mmontesg/publicaciones/ 2000/ComparisonCG.
15. Muller H.M., Kenny E.E., Sternberg P.W. (2004). ―An Ontology-Based Information Retrieval and Extraction System for Biological Literature". PLoS Biol. 2(11):e309. doi:10.1371/journal.pbio.0020309.
16. Knappe R., Bulskov H., Andreasen T. (2004). Perspectives on Ontology-based Querying // International Journal of Intelligent Systems, Access mode: http://akira.ruc.dk/~knappe/publications/ijis2004.pdf.
17. Jacso, Peter. (2010). ―The impact of Eugene Garfield through the prizm of Web of Science,". Annals of Library and Information Studies, Vol. 57, p. 222.
18. Christoph Meinel Serge Linckels (2007). Semantic interpretation of natural language user input to improve search in multimedia knowledge base, Information Technologies, 49(1):40–48.
19. Giorgos Stoilos, Giorgos Stamou, and Stefanos Kollias (2005) A String Metric For Ontology Alignment, Proc. of the 4rd Int. Semantic Web Conf. (ISWC), vol 3729 of LNCS, p. 624–637, Berlin. Springer.