Web Resource Changes Monitoring System Development

Lyubomyr Chyrun[0000-0002-9448-1751]1, Yevhen Burov[0000-0001-8653-1520]2,
Bohdan Rusyn[0000-0001-8654-2270]3, Liubomyr Pohreliuk[0000-0003-1482-5532]4,
Oleh Oleshek[0000-0002-2222-8773]5, Aleksandr Gozhyj [0000-0002-2641-5947]6,
Igor Bobyk[0000-0002-0424-1720]7

[1]IT Step University, Lviv, Ukraine
[2,5,7]Lviv Polytechnic National University, Lviv, Ukraine
[3-4]Karpenko Physico-Mechanical Institute of the NAS Ukraine
[6]Petro Mohyla Black Sea National University, Nikolaev, Ukraine
chyrunlv@gmail.com[1], yevhen.v.burov@lpnu.ua[2], rusyn@ipm.lviv.ua[3],
liubomyr@inoxoft.com[4], oleshek95@gmail.com[5], alex.gozhyj@gmail.com[6],
igor.bobyk@gmail.com[7]

Abstract. The detection of changes in the web content and notifying users about them in timely manner is an important service for both the owners of the web sites and ordinary users. The system created automatically captures and monitors the differences between the content of the web page at a given moment of time and compares them with saved earlier. The system periodically parses the contents of files on a server, available for public access according to the given pattern. It compares its contents with previously stored version and in case if discrepancies are discovered, immediately notifies the user. Thus, the user will not have to monitor manually the specified Internet resources.

Keywords: Web resource, content, SEO technology, information technology, web technology, changes monitoring, web server, intelligent system, operating system, web page, automated monitoring system, neural network, content downloader, automated monitoring, software product, monitoring system, trademark office, computer science, neural network, intelligent automated monitoring system, user interface, global network, internet resource, web service, web site, regular expression pattern, apache http server, electronic content commerce system, server part

1 Introduction

With every year, the information technology industry gains more and more rapid development. Smart systems for monitoring, analyzing and managing things are being introduced everywhere. However, the development of information technology is not limited only to this area. All related industries are also developing at a rapid pace. In order to increase productivity and improve the quality of the final product, the human labor is replaced by automation [1]. A circle is creating: the development of the IT sphere stimulates other sectors of employment, which in theirs turn stimulate the development of IT. That is why this process is becoming increasingly widespread with every year. It is logical that during this rapid development, we have a desire to keep up with the new trends and to develop at the same pace. To meet these needs, we need to constantly "consume" and "absorb" a huge amount of fresh information from multiple sources. Unfortunately, the development of our abilities to perform this process cannot be improved by purchasing new, more powerful computers [2]. For today, it is not enough to read just published books or magazines to keep a finger on the pulse of the modern world [3]. The main reason of this is the huge amount of resources, needed for the publication of even one textbook, the main of which is time. The time that is spent on the design, printing and distribution of this publication. Fortunately, we have the Internet for more rational use of opportunities. Exactly thanks to this the global web has gained such popularity. As the main source of information the Internet opens for us lots of data venues. Similarly, the content sites are trying to attract the user with all possible palettes of web site design and functionality [4]. For this purpose, owners of Internet resources don't spare the forces and facilities. In the face of intense competition, there is a need to constantly maintain the quality of the product at the proper level [5]. To achieve this aim it is necessary to invest not only in the development of something new, but also in preserving the already acquired that is protecting the site from unwanted changes and supervising the correct the work of all its components [6]. As an option we can design a lot of data processing, create a bunch of checks to protect against penetration of the unauthorized code into a page. But none method or even a combination of them all will allow to be sure for one hundred percent in the security of the initial data [7]. That is why the logical decision is not to going blindly for the unachievable aim, but to take care of possible solutions of inevitable problems as they appears [8]. An intelligent system of automated monitoring of changes in web resources serves this purpose.

2 Analytical review of literary and other sources

2.1 Subversion (SVN)

The first systems that provided the tracking of changes in the documents were the version control systems [9]. One of them is Subversion or SVN. Officially it was introduced in 2004 by CollabNet and represents a free, centralized version control system. It was built on the basis of CVS, but without its errors and disadvantages. At that time CVS was an unofficial standard among developers [10].

The work with SVN is not much different from the work with other centralized version control systems. Users copy files from the central repository, works with these copies and synchronize them with those that are left in the main repository, commenting on new edits. The ability of two or more users working with the same files is a useful feature. This is provided by creating local copies of the original files. However, this product also has the function of blocking access to a particular file, so that only one user could work with it. Also, the smart decision of this system is the use of the mechanism of delta-compression – the saving of not the original and final contents of the file, but only the differences between them [11].

The key features that SVN system provides for its users [12]:

- smart tracking of changes of all in a given directory – the file transfers, their renaming or attributes editing are fixed;
- convenient mechanism of synchronization of the proper files;
- emulation procedure:
  - the making of new branches takes place by copying the old ones and adding new edits;
  - automatic finding of differences and their unification during branches merging;
- edits, no matter how large are they, are fixed in a single repository in the atomic transactions representation;
- transferring and processing only the differences between files, but not themselves;
- the same qualitative processing of both binary and text files;
- two storage formats:
  - repository – database;
  - repository – a set of regular files;
- the ability to write commits in any language;
- the ability to create a reflection copy of the repository;

Although the SVN system is based on CVS, however, there is little left from it. Here is a list of the major differences [13]:

- the way of working with text and binary files was improved;
- boundaries in which the system tracks changes were increased – now it can control not only files but also directories;
- a set of properties of each file was added to the control;
- it saves network traffic during transferring files to the repository that are not completely modified but only different parts;
- the ability to lock files during work was created, so that nobody can makes changes in parallel.

For a better understanding of what a version control system represents and what possibilities it offers, see Fig. 1 [14].

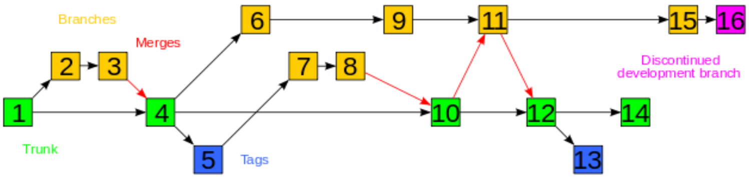


Fig. 1. Visualization of a simple Subversion project

This figure shows the way of project development. It resembles the structure of a tree with a multitude of branches. Exactly from here the name for the mechanism for creating various temporary versions of one software product is created. The trunk of the project is marked by green color, by yellow – the branches. Branches are created at the time of development for some additional functions in the system. In the case of their successful realization additional branches are merged into the trunk. This process is called synchronization and is indicated by a red arrow in this figure. Blue – branches of the project that will no longer change, are called tags. Discontinued development branch is marked by purple [15]. The process of user interaction with the SVN system can be divided into the following parts [16]:

1. Connect to a central repository.
2. Download the latest version of the project from the repository or creation a new one.
3. If the project was created earlier – make the necessary changes to the working copy. If this is a new project – work with it as with any other project, not tracked by this system.
4. Commit of accomplished actions and providing a textual description of what was made.
5. Transfer the completed commit to the repository.

Due to the strict observance of this plan the history of the project development is created with the possibility at any time to return to a certain stage of development and go to a different way [17].

2.2 SiteLock

SiteLock is protecting web pages in real time [18]. This product was introduced in 2009 as the start-up of two programmers – Neville Feather and Scott Lovell, which eventually turned into an advanced security in the field of web security. This product was developed to expand the ability to protect resources written in WordPress CMS. It extends the set of standard features by its own, which will be useful to both the web developers and the owners of Internet resources. The main advantage of this software is the enormous variety of website scanning tools that prevent most problems for appearing. SiteLock provides its users with the following scanning capabilities [19]:

- Applications scanning – checking for outdated, unstable or unsafe applications;
- SMART scanning – access to the root folder of the site is going through the FTP protocol and runs a full scan to identify weak spots or malware. Files are checked in binary format, which makes it impossible to miss even the least errors;
- A regular malware search is a simpler version of SMART validation that uses less server resources and is suitable for fast site crawling;
- Providing the information about domain – regular checking who the owner of domain is;
- SSL certificate validation for relevance;
- The detection of SQL injections – if rogue software attempts to transfer SQL code through the form of the site into the database. In case of a successful injection the database will be restored from backlog copy and the owner will receive a detailed report by what method it was realized;

- Spam protection is search for a domain in the largest spam databases (check whether the e-mail address declared on the site has not been broken and is not used for the distribution of spam);
- XSS testing is similar to SQL injections – but detecting an attempt to introduce redirects of end-user requests;
- Searching for open network ports;
- Scanning of recommendations – checking external redirects, cookies, etc.

SiteLock also can boast with its own centers of data processing that are strategically located to provide the shortest response time to the request and the ability to cache user sites on their own servers. Also this system interacts through leading network providers of level 1 to provide its customers with high speed servers with high capacity [20]. All this enables SiteLock owners to give their services equally to users regardless of their location. The capabilities of caching data on system servers are as follows [21]:

1. Dynamic – this method constantly updates the information on the website and its display. Depending on which content of the site is changing and how it changes, the forecast for future changes is made to establish a balance between caching data for a quick response and the relevance of these data.
2. Static – simple caching of static content of a web resource. It includes caching HTML files, a JavaScript resource and images.
3. Caching on the user's side – keeping as much data as possible in the browsers of the site visitors for even faster display of content and offline viewing of web resources.
4. Loading pages from memory – pages with the largest number of requests are cached in RAM. This can be done without working with the file system and buffers.

Particularly useful for owners is the ability to self-control which data should be cached, and which should not. So, in the case of site design changes or some other global changes, SiteLock allows to delete all or only certain parts of the cached content. Another strong side of this product is the availability of support. It works in 24/7 format, that enables users to get an answer to the question at any time. SiteLock works well with its main aim – protecting users' web resources and providing means to improve their end products – that is why it became so popular. However, its disadvantage is limited support of CMS systems [22].

2.3 Content Downloader

The Content Downloader software is a parser. With its help it becomes possible to save on computer the files with text and graphics presented on the site. In order to work with it the knowledge of programming is not required, which is an advantage, because it allows to include in the list of customers the ordinary users. This software product, like its counterparts, is often used to quickly fill the new web page or to download and compare different sites [23]. This software is not a new one, it has been in the market for a long time and has proved to be a reliable and convenient tool with a lot of useful features. One of the main criteria for choosing this tool among many others is the simple, intuitive interface that is an important argument in terms of attracting new customers and keeping old ones [24]. The user only needs to assign the address of the resource from where the data is parsed, the size of data and choose the format in which to save the results of the work. Parser has a large set of analysis methods, which allows to collect data effectively from entire web pages or only necessary parts of them [25]. Content Downloader can be used for [26]:

- Phone numbers, email addresses or other contact information parsing;
- Downloading information (descriptions, prices, images) from online stores and saving results to a CSV table;
- The specialized part of the web resource (tables, headers, inserts of the code, conclusions) parsing;
- Local storage of XML-maps of the sites;
- The attached files parsing;
- Downloading of the hidden data, access to which is possible only after authorization or clicking on the relevant items (this feature is only available during purchasing the maximum version of the product, which extends the basic Content Downloader functionality with the WebApp application that actually simulates a click on a button);
- Any content that matches the filters from all declared data sources parsing.

Content Downloader qualitatively differs from other parser programs with the ability to fine tune the system to any requirements, to ensure comfortable work of users having different levels of understanding of the system and with various tasks complexity. Among its many features one should pay attention to the following [27]:

- The ability to set ways of authorization on the resources from which parsing will occur;
- The ability to configure cookies files so that the user do not have to constantly relogin;
- Set the pause time with which server requests will be realized;
- Adjust the number of parsing threads;
- Saving resources received during parsing as:
  - local resource – into the system where the Content Downloader is installed;
  - remote resource – cloud storage or FTP access – that is adjusting the process of automatic downloading data;
  - any SQL database;
- Simultaneous parsing of different information from one source, and breakdown of the results into different categories;
- The ability to connect own PHP scripts for more deeper and more detailed processing of the data;
- The ability to set any User Agent, and its replacement during the work of the parser;
- Setting a custom proxy server (one or more);

- Ability to download site resources through a browser to connect JavaScript scripts;
- The ability to “competently” parse and store data in different encodings;
- Setting the multithreading work of the system (both during data parsing and saving);
- A large selection of data extensions (CSV, TXT, HTM, PHP, MySQL).

Content Downloader is a product specifically designed to download content from websites [28-31]. On the one hand, it is his advantage, on the other – a drawback. The advantage is a specified category of classes, with which this product works perfectly. The disadvantage is the supplying only the raw data, so in order to provide certain statistics or in the case of analytical comparison of data from different sources, the user will have to do it manually, or to look for a different system [32-35].

2.4 Advantages and disadvantages of considered software products

The software products considered above are not a novelty in the software market. Each of them perfectly proved itself and became entrenched in its “realm”. Due to the long time they have been in the market they have formed a clear range of services that are needed for their users. After analyzing these services, as well as those which, due to certain circumstances, are left without proper attention from the considered systems, we were able to formulate requirements for automated monitoring system. In the Table 1 the advantages and disadvantages of reviewed systems are described [30].

Table 1. Advantages and disadvantages of considered software products

Product	Advantages	Disadvantages
Subversion	Ability to view all changes that the document have undergone. Centralized data storage. Saving only changes, not files.	Imperfect mechanism of simultaneous work. The impossibility of completely deleting data. To identify changes the user needs to add new data to the repository.
SiteLock	A various web site scanning procedures. Automatic response in the event of a threat. Ability to improve the basic characteristics of servers.	Orientation on the CMS system. Few services are included in the basic product kit. Restrictions on monitoring
Content Downloader	The possibility of multi-type parsing with a large number of custom parameters. Ability to plan regular parsing without user participation.	Limitation of the system only for data parsing.

3 Systems analysis and the statement of the problem

The aim of this work is to create intelligent automated monitoring system, so let’s first state the definition of “intelligent system” and “automated monitoring” notions. Intelligent system is a system of technical or programmatic nature that can independently solve its tasks from a certain subject area, creating a new knowledge based on the acquired earlier. The process of functioning of such a system usually results in constant decision making taking into account the current state of external and internal factors in order to achieve certain results. The functioning of such systems includes certain stages:

1. Clarification of the initial state at the time of task assignment to the system.
2. The comparison of this state for all factors and tasks with all previous states and tasks. As a result, there is a check on the possibility of finding a quick solution for the given task.
3. In the absence of the possibility to finding the response immediately, the system proceeds with decomposition of the task into a series of smaller ones, and repeats the actions described in paragraph 2, for each of these tasks.
4. The collection of the results from step 3 into one unit and the work results formation.
5. Saving the information about the task, the initial state of external and internal factors, the results of its realization to the base of the previous tasks.

The automated monitoring is a process of continuous collection of certain data (indicators, results of calculations, etc.), their analysis and the formation of further behavior depending on the results of the analysis. Also the key feature of this process is the complete absence of human interference. So, the intelligent automated monitoring system is a system that is created to further reject the need for a human participation in the process of gathering information, processing it and shaping the proper behavior in the particular subject area. The main use cases of automated monitoring system are shown on fig. 2. There are actors such as the User and the System. The User has two kinds of options:

- The receiving of notification as the message sent in a user-friendly format including the report about the monitoring results.
- Formation of the query. This option is somewhat wider and includes the more specific ways of interacting with the system.

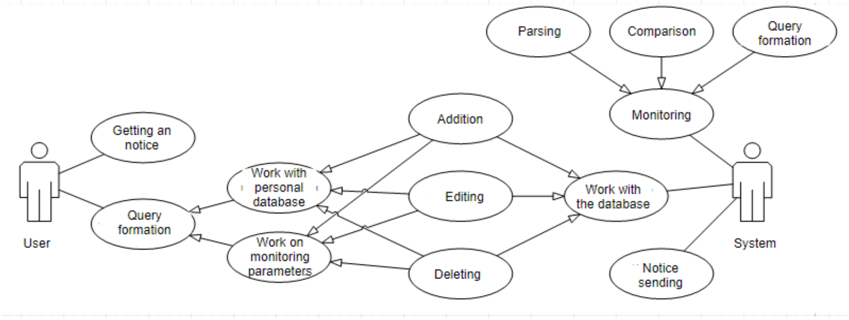


Fig. 2. The use case diagram

The case “Request Formulation” can be divided into two sub-cases. The user can either work with a user profile (email address, password, site name) or work with monitored parameters (creating new patterns and resources for tracking, changing existing validation rules, removing unnecessary functionality). The actor System has the following interaction options:

- Sending the report on the results of monitoring of relevant resources for the selected users.
- Monitoring case includes following sub-cases:
  - Parsing is the process of downloading the content from a web resource.
  - Comparison is the process of identifying differences with the content saved in the database earlier.
  - Report formation is presentation of differences (if there are any) in a user-friendly format.
- Work with the database case is recording, reading, editing the data stored there.

Activity diagram (Fig. 3) is used to display the sequence of all user actions with the system. The activity diagram is constructed in the form of an oriented graph. The actions are indicated by rounded rectangles. Decisions are marked by a diamond. If there is a set of actions that are executed in parallel, then the beginning and the end of parallel actions are indicated by a straight horizontal line. The black circle means the beginning of the process, and the circled black circle means the end. Activity diagrams are very similar to the flowcharts, so they can be considered as such. In the activity diagram it can be noted that the user of the system can perform a certain sequence of operations – registration, or logging-in, correction of available data in the system or the assignment of new ones.

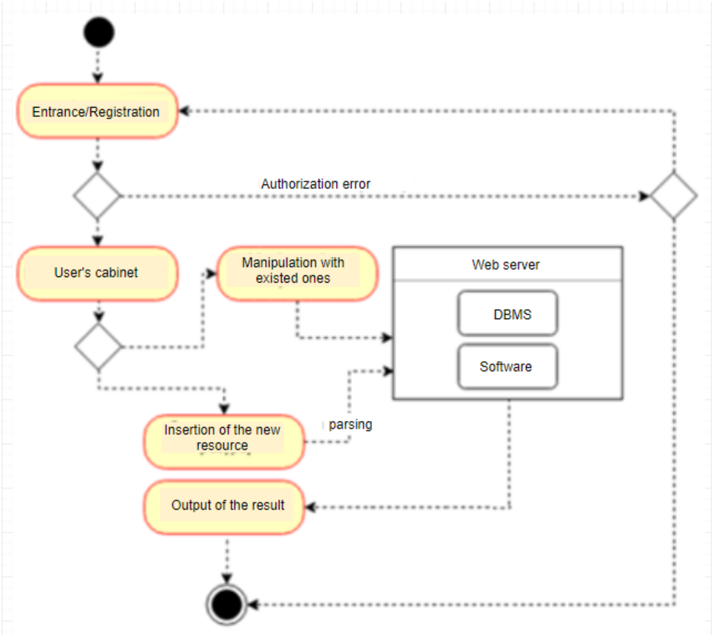


Fig. 3. Activity diagram

While in working in his personal cabinet, the user can see a list of all available web resources that he added to the monitoring list. It can also perform certain actions with them (such as deleting, editing, checking, pausing, etc.) or adding more resources to monitor. During the adding a new page, the user will be offered to select specific patterns of what is needed to track. He may also create his own. Those patterns represents static elements of the system (classes, their content and attributes). In the class diagram the package designation may also be additionally displayed. Also it allows to place some procedural elements, but this dynamics is better shown on other types of diagrams. A class diagram is intended to display the static structure of the system in terms used in object-oriented programming. It is best not to show anything on this diagram, except classes, interfaces, objects and their relationships. The class depicted in this diagram is determined by the attributes and methods of a particular set of instances. All instances of class are characterized by a common behavior and the same set of attributes (each instance of the class has its own set of values). Sometimes the name "class" can be replaced by "type", but it is worth to note that these two names, in fact, characterize different things – the type is considered more general definition. Classes are depicted by rectangles signed on the top.

As a result of the process, the object can change its class or state. These two types of changes do not have radical differences from each other: objects belonging to any one class can firstly convert to a parent or branch class. In Fig. 4 the diagram of states of the object of the automated monitoring system is shown.

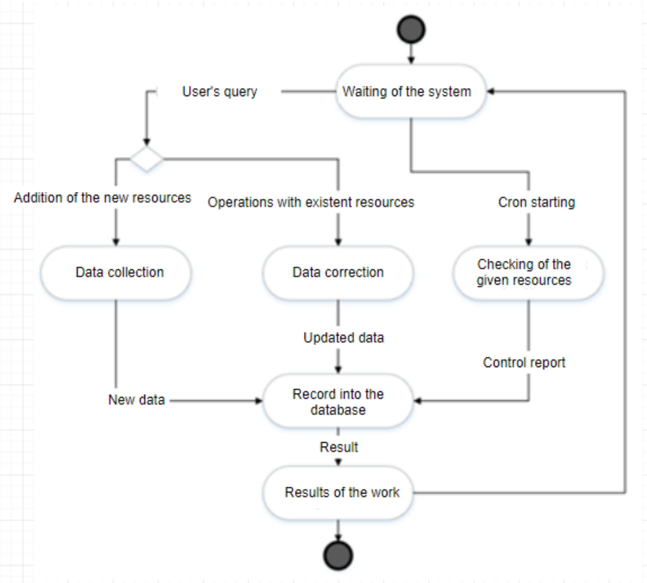


Fig. 4. State Diagram

Each state diagram has two distinct states of the object: the beginning and the end. The peculiarity of these two states is that it does not have any event that could force an object to change to its original state and does not have any such event that could return the object from the state of the end after it has been reached. Looking at the state diagram, it becomes clear that the monitoring system has two modes of operation:

1. The mode of execution of user queries – when the user is logged in and through the graphical interface interacts with the system.
2. Passive Query Mode – when the user does not specify directly what the monitoring system should do, it only receives a notification of the results of its work, according to the previously configured task.

The deployment diagram is intended to display the computing nodes of the system during its operation, as well as components of these nodes, which are also involved in the process.

The nodes of the deployment diagram are rectangular boxes with artifacts inside, depicted by rectangles. To represent a hierarchy of nodes, it is allowed to insert boxes. One node on such diagram can show a large number of physical nodes, as a cluster of all servers with databases (Fig. 5). The nodes are of two types:

- The device node is the physical computing node. It has its own memory and other means to run the software;
- The runtime environment node is a virtual resource that runs on an external basis and is designed to execute other software elements.

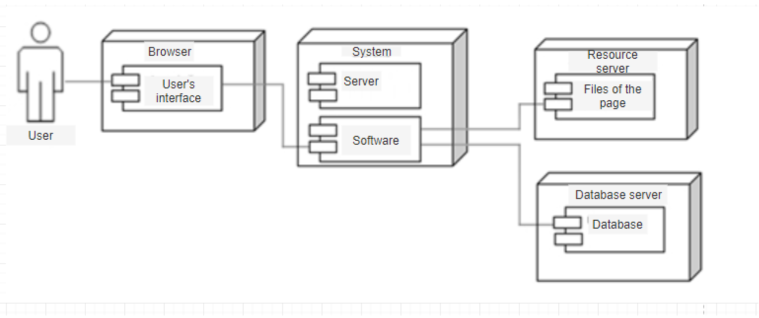


Fig. 5. The deployment Diagram

The deployment diagram for the automated monitoring system contains four nodes:

- The user’s browser through which interaction occurs.
- The system itself (its hardware and software).
- The resource server which provides monitored data.
- Database server.

Since the system is should preserve the results of monitoring and compare them with previous results, the system requires a large amount of memory. On the early stages of monitoring system usage it would be enough to use one server to execute software queries, but with the load growing, the servers will be distributed. Also, the advantage of this solution will be to spread the database at different nodes to ensure reliability and stability of the work.

4 The implementation of changes monitoring system

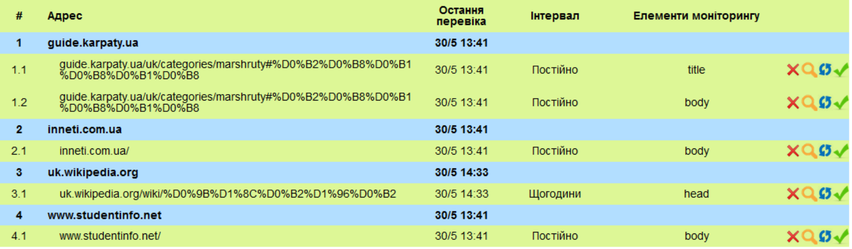
The working of monitoring system is based on the server-side utility for UNIX-like operating systems cron. It serves to execute scripts with a certain periodicity or for one-time execution at some specified moment. Under its command, the script cron_check.php is performed, located in the root file of the project, which in its turn goes through all the resources added by the users and determines whether it is time to re-verify them. If the time comes, the url adress of the resource and the regular expression by which the data on the page should be found is taken. The obtained data is then compared with those stored in the database, and in the case of detecting changes the user who added the resource for verification is informed.

To enable automated monitoring system to fulfill its purpose it needs to set several initial parameters:

1. Address of the resource to be monitored.
2. A regular expression pattern to specify what exactly needs to be monitored. The user can specify his own pattern or select from the list of available.
3. The time period for which the system should monitor the given site.

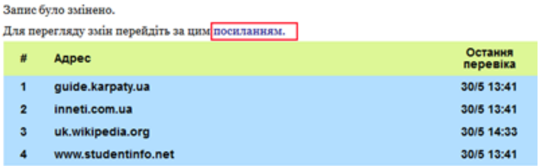
All these items can be specified on the Add page. The main element of the user interface of the monitoring system is a table with a list of the web resources added to it, general information about them and a set of functions for the work (Fig. 6). Since each page in the global network has its own domain there is an additional grouping of custom links behind it to visually separate the various resources. On this page, the user has the following set of functions:

- ✗ (delete) removes links from the table if resources with such domains do not exist, removes and deletes also all data from the database about it;
- 🔍 (scan) performs an immediate check of a given resource based on the specified regular expression (the result of this check is not stored in the database). If changes were detected, the system will provide links by which these changes could be reviewed (Fig. 7);
- 🔄 (update) performs parsing by pattern in order to create the basis for future comparisons. It will be especially useful in case of finding the differences that are permissible for the user and he would like to put them into the system;
- ⓘ✓ (change status) contains one of the two icons. The user switches them depending on whether he wants the resource to automatically pop up or not.



#	Адрес	Остання перевіка	Інтервал	Елементи моніторингу	
1	**guide.karpaty.ua**	**30/5 13:41**			
1.1	guide.karpaty.ua/uk/categories/marshruty#%D0%B2%D0%B8%D0%B1%D0%B8%D0%B1%D0%B8	30/5 13:41	Постійно	title	✗ 🔍 🔄 ✓
1.2	guide.karpaty.ua/uk/categories/marshruty#%D0%B2%D0%B8%D0%B1%D0%B8%D0%B1%D0%B8	30/5 13:41	Постійно	body	✗ 🔍 🔄 ✓
2	**inneti.com.ua**	**30/5 13:41**			
2.1	inneti.com.ua/	30/5 13:41	Постійно	body	✗ 🔍 🔄 ✓
3	**uk.wikipedia.org**	**30/5 14:33**			
3.1	uk.wikipedia.org/wiki/%D0%9B%D1%8C%D0%B2%D1%96%D0%B2	30/5 14:33	Щогодини	head	✗ 🔍 🔄 ✓
4	**www.studentinfo.net**	**30/5 13:41**			
4.1	www.studentinfo.net/	30/5 13:41	Постійно	body	✗ 🔍 🔄 ✓

Fig. 6. User resources table



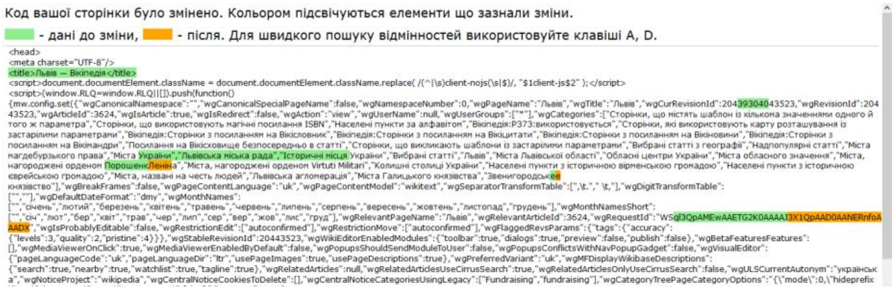
Запис було змінено.

Для перегляду змін перейдіть за цим посиланням.

#	Адрес	Остання перевіка
1	guide.karpaty.ua	30/5 13:41
2	inneti.com.ua	30/5 13:41
3	uk.wikipedia.org	30/5 14:33
4	www.studentinfo.net	30/5 13:41

Fig. 7. Link to the differences page

If the user decides to activate the link, a page similar to that in Fig. 8 will open.



Код вашої сторінки було змінено. Кольором підсвічуються елементи що зазнали зміни.

🟩 - дані до зміни, 🟧 - після. Для швидкого пошуку відмінностей використовуйте клавіші A, D.

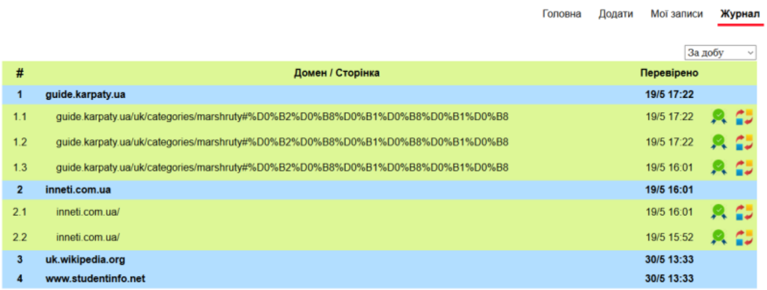
```
<head>
<meta charset="UTF-8"/>
<title>Львів — Вікіпедія</title>
<script>document.documentElement.className = document.documentElement.className.replace( /(^|\s)client-nojs(\s|$)/, "$1client-js$2" );</script>
<script>(window.RLQ=window.RLQ||[]).push(function()
{mw.config.set({"wgCanonicalNamespace":"","wgCanonicalSpecialPageName":false,"wgNamespaceNumber":0,"wgPageName":"Львів","wgTitle":"Львів","wgCurRevisionId":20439304043523,"wgRevisionId":204
43523,"wgArticleId":3624,"wgIsArticle":true,"wgIsRedirect":false,"wgAction":"view","wgUserName":null,"wgUserGroups":["*"],"wgCategories":["Сторінки, що містять шаблон із кількома значеннями одного й
того ж параметра","Сторінки, що використовують магічні посилання ISBN","Населені пункти за алфавітом","Вікіпедія:P373:використовується","Сторінки, які використовують карту розташування із
застарілими параметрами","Вікіпедія:Сторінки з посиланням на Вікісловник","Вікіпедія:Сторінки з посиланням на Вікіцитати","Вікіпедія:Сторінки з посиланням на Вікіновини","Вікіпедія:Сторінки з
посиланням на Вікімандри","Посилання на Вікісховище безпосередньо в статті","Сторінки, що викликають шаблони із застарілими параметрами","Вибрані статті з географії","Надпопулярні статті","Міста
магдебурзького права","Міста України","Львівська міська рада","Історичні місця України","Вибрані статті","Львів","Міста Львівської області","Обласні центри України","Міста обласного значення","Міста,
нагороджені орденом ПорошенкЛеніна","Міста, нагороджені орденом Virtuti Militari","Колишні столиці України","Населені пункти з історичною вірменською громадою","Населені пункти з історичною
єврейською громадою","Міста, названі на честь людей","Львівська агломерація","Міста Галицького князівства","Звенигородськее
князівство"],"wgBreakFrames":false,"wgPageContentLanguage":"uk","wgPageContentModel":"wikitext","wgSeparatorTransformTable":[",\t."," \t,"],"wgDigitTransformTable":
["",""],"wgDefaultDateFormat":"dmy","wgMonthNames":
["","січень","лютий","березень","квітень","травень","червень","липень","серпень","вересень","жовтень","листопад","грудень"],"wgMonthNamesShort":
["","січ","лют","бер","квіт","трав","чер","лип","сер","вер","жов","лис","груд"],"wgRelevantPageName":"Львів","wgRelevantArticleId":3624,"wgRequestId":"WSql3QpAMEwAAETG2K0AAAAJ3X1QpAAD0AANERnfoA
AADX","wgIsProbablyEditable":false,"wgRestrictionEdit":["autoconfirmed"],"wgRestrictionMove":["autoconfirmed"],"wgFlaggedRevsParams":{"tags":{"accuracy":
{"levels":3,"quality":2,"pristine":4}}},"wgStableRevisionId":20443523,"wgWikiEditorEnabledModules":{"toolbar":true,"dialogs":true,"preview":false,"publish":false},"wgBetaFeaturesFeatures":
[],"wgMediaViewerOnClick":true,"wgMediaViewerEnabledByDefault":false,"wgPopupsShouldSendModuleToUser":false,"wgPopupsConflictsWithNavPopupGadget":false,"wgVisualEditor":
{"pageLanguageCode":"uk","pageLanguageDir":"ltr","usePageImages":true,"usePageDescriptions":true},"wgPreferredVariant":"uk","wgMFDisplayWikibaseDescriptions":
{"search":true,"nearby":true,"watchlist":true,"tagline":true},"wgRelatedArticles":null,"wgRelatedArticlesUseCirrusSearch":true,"wgRelatedArticlesOnlyUseCirrusSearch":false,"wgULSCurrentAutonym":"українськ
a","wgNoticeProject":"wikipedia","wgCentralNoticeCookiesToDelete":[],"wgCentralNoticeCategoriesUsingLegacy":["Fundraising","fundraising"],"wgCategoryTreePageCategoryOptions":"{\"mode\":0,\"hideprefix
```

Fig. 8. Page with the presented differences

The differences between two texts are highlighted The page headline gives a brief guide to the correct interpretation of what is shown below. Also in the top is the list of hotkeys for quick navigation is shown.

In case of acceptance of changes, the user updates the record by clicking on the correspondent button on the resource page. In the case if the detected changes are not caused by user actions or are not approved, the user proceeds to removing them.

It is also possible that the changes occurred, the system found them and informed the user, but due to certain circumstances, he failed to react to them. In this case it would be helpful to view the history of all changes on the register page (Fig. 9).



#	Домен / Сторінка	Перевірено
1	**guide.karpaty.ua**	**19/5 17:22**
1.1	guide.karpaty.ua/uk/categories/marshruty#%D0%B2%D0%B8%D0%B1%D0%B8%D0%B1%D0%B8	19/5 17:22
1.2	guide.karpaty.ua/uk/categories/marshruty#%D0%B2%D0%B8%D0%B1%D0%B8%D0%B1%D0%B8	19/5 17:22
1.3	guide.karpaty.ua/uk/categories/marshruty#%D0%B2%D0%B8%D0%B1%D0%B8%D0%B1%D0%B8	19/5 16:01
2	**inneti.com.ua**	**19/5 16:01**
2.1	inneti.com.ua/	19/5 16:01
2.2	inneti.com.ua/	19/5 15:52
3	**uk.wikipedia.org**	**30/5 13:33**
4	**www.studentinfo.net**	**30/5 13:33**

Fig. 9. The register of all changes to each page

Also, this page provides an additional functionality:

- (compare with the standard) is the comparison to the state of the resource at the moment of its adding to the system.
- (view recent changes) shows the changes that occurred on the site at the time and those that were detected during the pre-check.

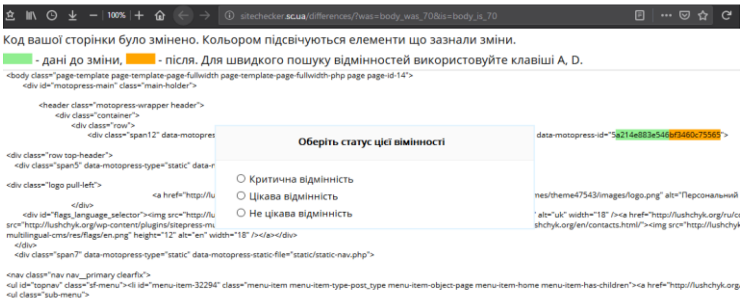
The intelligent system of automated monitoring of changes in web resources is called the intellectual exactly because of the ability to analyze user behavior and based on the collected data to predict its future actions and partially perform them instead of the user (of course, with his prior agreement). In this system the decision-making mechanism is assigned to an artificial neural network. The Fast Artificial Neural Network Library (FANN), a free neural network library with an open code that realize multilayer artificial neural networks on C was taken to perform this operation. This resource provides artificial neural networks more than on 20 programming languages. Below is a list of features of this library, being the reason they were chosen for development:

- the ability to develop a topological training that dynamically builds and trains an artificial neural network;
- easiness in use – gives the ability to create, train and run a network using three functional calls);
- high speed performance;
- possibility of setting on the fly with a lot of parameters;
- well documented;
- realized several different activation functions;
- open source, but can still be used in commercial applications (licensed by LGPL);
- language binding for a large number of different programming languages;
- widely used (approximately 100 downloads per day).

The programming language which is used for the interaction with this library and for the monitoring system is PHP. The neural network learns to recognize the following types of differences:

- critical – in case of detection which the user must be immediately notified; data about it is kept (required for tracking);
- interesting – the system notifies the user, but don't add information about it in the register of changes (their monitoring may be disabled at the request of the user);
- not interesting – is ignored

On Fig. 10 the learning process of the neural network is shown


100%
sitechecker.sc.ua/differences/?was=body_was_70&is=body_is_70

Код вашої сторінки було змінено. Кольором підсвічуються елементи що зазнали зміни.

🟩 - дані до зміни, 🟧 - після. Для швидкого пошуку відмінностей використовуйте клавіші A, D.

```
<body class="page-template page-template-page-fullwidth page-template-page-fullwidth-php page page-id-14">
    <div id="motopress-main" class="main-holder">

        <header class="motopress-wrapper header">
            <div class="container">
                <div class="row">
                    <div class="span12" data-motopres                    data-motopress-id="5a214e883e546bf3460c75565">
<div class="row top-header">
  <div class="span5" data-motopress-type="static" data-r

<div class="logo pull-left">
                                    <a href="http://lu                    nes/theme47543/images/logo.png" alt="Персональний
                    </div>
    <div id="flags_language_selector"><img src="http://lu                    ' alt="uk" width="18" /><a href="http://lushchyk.org/ru/c
src="http://lushchyk.org/wp-content/plugins/sitepress-mu                    shchyk.org/en/contacts.html/"><img src="http://lushchyk
multilingual-cms/res/flags/en.png" height="12" alt="en" width="18" /></a></div>
  </div>
  <div class="span7" data-motopress-type="static" data-motopress-static-file="static/static-nav.php">

<nav class="nav nav__primary clearfix">
<ul id="topnav" class="sf-menu"><li id="menu-item-32294" class="menu-item menu-item-type-post_type menu-item-object-page menu-item-home menu-item-has-children"><a href="http://lushchyk.org
<ul class="sub-menu">
```

Оберіть статус цієї вімінності

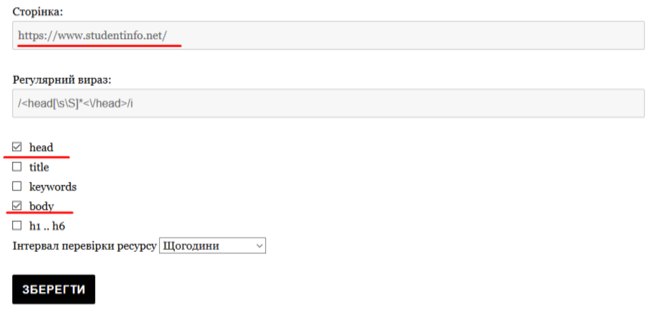
○ Критична відмінність
○ Цікава відмінність
○ Не цікава відмінність

Fig. 10. The teaching of the neural network

It should be noted that any differences which the system will find for the first time will be marked as critical. When viewed, the user assigns one of the proposed statuses. Further, when neural network has learned to the satisfactory level, it will be able to independently review the differences and determine whether or not to notify the owner of this resource. In the early stages of the network, each distinction will necessarily need to be approved for some status several times, to ensure correct weight adjustment.

5 An example of resource change monitoring system operation

To ensure that all components of the system function properly, will tested it on variety of web resources, making changes and verifying whether the system can detect them. As an example we use the main page of the site sudentinfo.net, and a pattern for monitoring its entire contents. The first step is to add this web resource into the system (Fig. 11). Next, go to the My records page to confirm the success of creating the reference resource for the resource added the previous step. In Fig. 12 it can be seen, that the resource will be monitored by the system every hour.



Сторінка:

https://www.studentinfo.net/

Регулярний вираз:

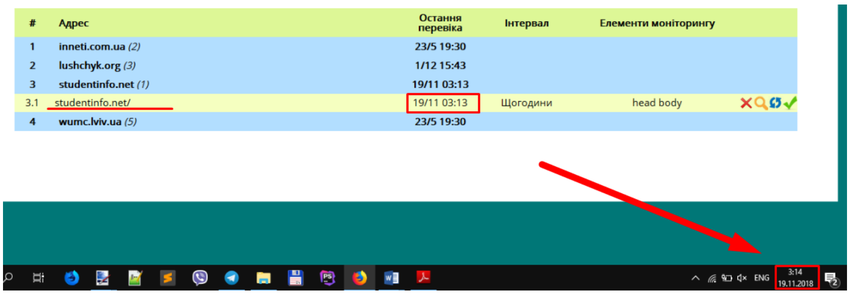
/<head[\s\S]*<\/head>/i

☑ head
☐ title
☐ keywords
☑ body
☐ h1 .. h6

Інтервал перевірки ресурсу Щогодини

ЗБЕРЕГТИ

Fig. 11. Adding resource to the system

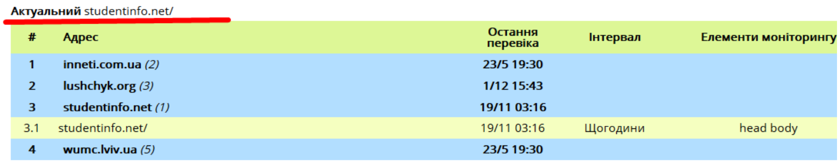


#	Адрес	Остання перевіка	Інтервал	Елементи моніторингу	
1	inneti.com.ua *(2)*	23/5 19:30			
2	lushchyk.org *(3)*	1/12 15:43			
3	studentinfo.net *(1)*	19/11 03:13			
3.1	studentinfo.net/	19/11 03:13	Щогодини	head body	✕ 🔍 ⟳ ✓
4	wumc.lviv.ua *(5)*	23/5 19:30			

ENG 3:14 19.11.2018

Fig. 12. Confirmation of adding a new resource

Next, we checked the newly added page for data dynamics – whether there’s a plug-in or extension that constantly changes its content, such as a clock or a visitor count. To do this, we clicked on the “magnifying glass” in the line of this entry and saw the result in Fig. 13.

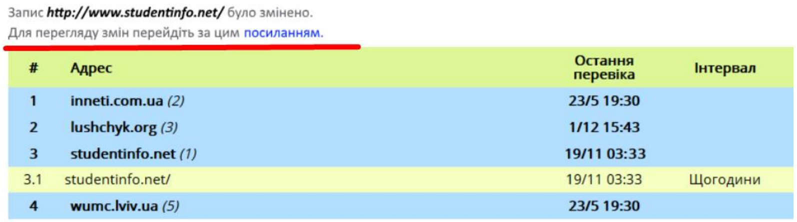


Актуальний studentinfo.net/

#	Адрес	Остання перевіка	Інтервал	Елементи моніторингу
1	inneti.com.ua *(2)*	23/5 19:30		
2	lushchyk.org *(3)*	1/12 15:43		
3	studentinfo.net *(1)*	19/11 03:16		
3.1	studentinfo.net/	19/11 03:16	Щогодини	head body
4	wumc.lviv.ua *(5)*	23/5 19:30		

Fig. 13. Checking the dynamic data

Now let’s fire the index.php file, which is responsible for the main page of studentinfo.net and add the text “sitechecker.sc.ua” to it. Put this text in the <head> tag – it hides it in the user’s browser, but in the case of monitoring, the system should detect these changes. Then again we will test in manual mode, so that we do not wait for an hour. The system detected changes (Fig. 14) and organized a link for their viewing.

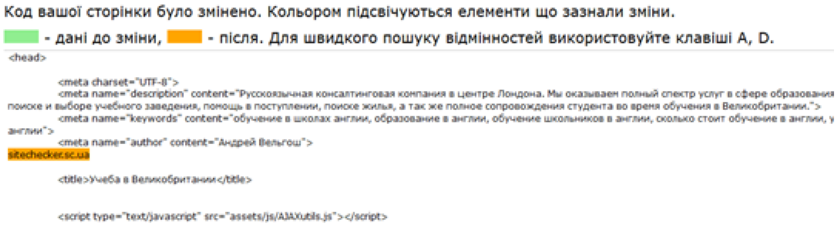


Запис ***http://www.studentinfo.net/*** було змінено.
Для перегляду змін перейдіть за цим посиланням.

#	Адрес	Остання перевіка	Інтервал
1	**inneti.com.ua** *(2)*	**23/5 19:30**	
2	**lushchyk.org** *(3)*	**1/12 15:43**	
3	**studentinfo.net** *(1)*	**19/11 03:33**	
3.1	studentinfo.net/	19/11 03:33	Щогодини
4	**wumc.lviv.ua** *(5)*	**23/5 19:30**	

Fig. 14. The changes found

By clicking on the link the user can see that the system correctly recognized the changes made and showed them as it was intended (Fig. 15). So the system worked correctly at all stages of the test. It certainly can’t be considered a hundred percent guarantee of its work, because in real conditions it will have to do dozens or even hundreds of more complicated inspections per second, but according to the results of testing we can say that the development of the system goes the right way.



Код вашої сторінки було змінено. Кольором підсвічуються елементи що зазнали зміни.

🟩 - дані до зміни, 🟧 - після. Для швидкого пошуку відмінностей використовуйте клавіші A, D.

```
<head>

          <meta charset="UTF-8">
          <meta name="description" content="Русскоязычная консалтинговая компания в центре Лондона. Мы оказываем полный спектр услуг в сфере образования
поиске и выборе учебного заведения, помощь в поступлении, поиске жилья, а так же полное сопровождения студента во время обучения в Великобритании.">
          <meta name="keywords" content="обучение в школах англии, образование в англии, обучение школьников в англии, сколько стоит обучение в англии, у
англии">
          <meta name="author" content="Андрей Вельгош">
sitechecker.sc.ua

          <title>Учеба в Великобритании</title>


          <script type="text/javascript" src="assets/js/AJAXutils.js"></script>
```

Fig. 15. Showing the changes that were made

6 Conclusion

After analyzing three different software products (different in structure and specialization, but similar in the functions), which have long been on the service market and are well known, we summed up their strengths and weaknesses and highlighted the functionalities necessary for created system. So:

- Subvertion is a great means for finding differences in files and demonstrating them to the end user. It is also worth noting the mechanism for storing and transmitting these differences when not all file contents are transmitted. This saves resources;
- Content Downloader is a multifunctional parsing software that has a large set of custom settings for downloading both entire pages and their small elements. These patterns will be reused in developed monitoring system;
- SiteLock is an application with a large set of different scanning algorithms that allows it to track even the smallest changes and make decisions independently depending on the type of changes.

As the result of this work the practical implementation of software system monitoring the changes in web content was created. We used following software platforms for system. The software was written in PHP; HTML and CSS; JavaScript. It was executed using Apache HTTP Server; MySQL; Debian – the operating system on the server under which the web server will be deployed; Webmin – a tool for managing the operating system through a web interface. This paper contains detailed description of initial analysis, system's functions and created software product. An example of changes monitoring system operation is also provided.

References

1. Iturrioz, J., Azpeitia, I., Díaz, O.: Generalizing the like button: empowering websites with monitoring capabilities. In: Proceedings of the 29th Annual ACM Symposium on Applied Computing, 743-750. (2014).
2. Maggi, F., Robertson, W., Kruegel, C., Vigna, G.: Protecting a moving target: Addressing web application concept drift. In: International Workshop on Recent Advances in Intrusion Detection, Springer, Berlin, Heidelberg, 21-40. (2009).
3. Christensen, J. H.: Using RESTful web-services and cloud computing to create next generation mobile applications. In: the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications, 627-634. (2009).
4. Neugschwandtner, M., Neugschwandtner, G., Kastner, W.: Web services in building automation: Mapping knx to obix. In: Int. Conf. on Industrial Informatics, 87-92. (2007).
5. Chen, L., Hind, J. R., Li, Y., Xiao, L.: U.S. Patent No. 8,613,039. Washington, DC: U.S. Patent and Trademark Office. (2013).
6. Breiter, G., Jall, D., Mueller, M., Neef, A., Reitz, M.: U.S. Patent No. 8,812,424. Washington, DC: U.S. Patent and Trademark Office. (2014).
7. Quintero, A. H., Fedor, J. S., Quan, A. G., Richardson, K., Scott, D. W., Piper, K. A.: U.S. Patent No. 6,910,071. Washington, DC: U.S. Patent and Trademark Office. (2005).
8. Leshko, I., Firstenberg, Y., Kumar, N.: U.S. Patent Application No. 13/528,873. (2013).
9. Dan, Asit, et al.: Web services on demand: WSLA-driven automated management. In: IBM systems journal, 43.1, 136-158. (2004)
10. Shelby, Z.: Embedded web services. In: Wireless Communications, 17(6), 52-57. (2010).
11. Garanina, N., Sidorova, E., Kononenko, I., Gorlatch, S.: Using Multiple Semantic Measures for Coreference Resolution in Ontology Population. In: International Journal of Computing, 16(3), 166-176. (2017)
12. Colton, P., Sarid, U.: U.S. Patent No. 8,880,678. Washington, DC: U.S. Patent and Trademark Office. (2014).
13. Colton, P., Sarid, U.: U.S. Patent No. 7,596,620. Washington, DC: U.S. Patent and Trademark Office. (2009).
14. Colton, P., Sarid, U.: U.S. Patent No. 8,291,079. Washington, DC: U.S. Patent and Trademark Office. (2012).
15. Colton, P., Sarid, U.: U.S. Patent No. 8,954,553. Washington, DC: U.S. Patent and Trademark Office. (2015).
16. Alipanah, N., Parveen, P., Khan, L., Thuraisingham, B.: Ontology-driven query expansion using map/reduce framework to facilitate federated queries. In: Proc. of the International Conference on Web Services (ICWS), 712-713. (2011).
17. Lytvyn, V., Sharonova, N., Hamon, T., Vysotska, V., Grabar, N., Kowalska-Styczen, A.: Computational linguistics and intelligent systems. In: CEUR Workshop Proceedings, Vol-2136 (2018)

18. Vysotska, V., Fernandes, V.B., Emmerich, M.: Web content support method in electronic business systems. In: CEUR Workshop Proceedings, Vol-2136, 20-41 (2018)
19. Kanishcheva, O., Vysotska, V., Chyrun, L., Gozhyj, A.: Method of Integration and Content Management of the Information Resources Network. In: Advances in Intelligent Systems and Computing, 689, Springer, 204-216 (2018)
20. Lytvyn, V., Vysotska, V., Chyrun, L., Chyrun, L.: Distance Learning Method for Modern Youth Promotion and Involvement in Independent Scientific Researches. In: Proc. of the IEEE First Int. Conf. on Data Stream Mining & Processing (DSMP), 269-274 (2016)
21. Korobchinsky, M., Vysotska, V., Chyrun, L., Chyrun, L.: Peculiarities of Content Forming and Analysis in Internet Newspaper Covering Music News, In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 52-57 (2017).
22. Lytvyn, V., Kuchkovskiy, V., Vysotska, V., Markiv, O., Pabyrivskyy, V.: Architecture of system for content integration and formation based on cryptographic consumer needs. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2018 – Proceedings 1, 391-395 (2018)
23. Euzenat, J., Shvaiko P.: Ontology Matching. In: Springer, Heidelberg, Germany, (2007).
24. Maedche, A., Staab, S.: Measuring Similarity between Ontologies. In: Knowledge Engineering and Knowledge Management, 251-263. (2002).
25. Xue, X., Wang, Y., Hao, W.: Optimizing Ontology Alignments by using NSGA-II. In: The International Arab Journal of Information Technology, 12(2), 176-182. (2015).
26. Martinez-Gil, J., Alba, E., Aldana-Montes, J.F.: Optimizing ontology alignments by using genetic algorithms. In: The workshop on nature based reasoning for the semantic Web, Karlsruhe, Germany. (2008).
27. Vysotska, V., Lytvyn, V., Burov, Y., Gozhyj, A., Makara, S.: The consolidated information web-resource about pharmacy networks in city. In: CEUR Workshop Proceedings (Computational linguistics and intelligent systems), 2255, 239-255. (2018).
28. Lytvyn, V., Vysotska, V., Burov, Y., Veres, O., Rishnyak, I.: The Contextual Search Method Based on Domain Thesaurus. In: Advances in Intelligent Systems and Computing, 689, 310-319 (2018)
29. Lytvyn, V., Vysotska, V.: Designing architecture of electronic content commerce system. In: Computer Science and Information Technologies, Proc. of the X-th Int. Conf. CSIT'2015, 115-119 (2015)
30. Lytvyn, V., Vysotska, V., Uhryn, D., Hrendus, M., Naum, O.: Analysis of statistical methods for stable combinations determination of keywords identification. In: Eastern-European Journal of Enterprise Technologies, 2/2(92), 23-37 (2018)
31. Vysotska, V., Chyrun, L.: Analysis features of information resources processing. In: Computer Science and Information Technologies, Proc. of the Int. Conf. CSIT, 124-128 (2015)
32. Vasyl, Lytvyn, Victoria, Vysotska, Dmytro, Dosyn, Roman, Holoschuk, Zoriana, Rybchak: Application of Sentence Parsing for Determining Keywords in Ukrainian Texts. In: Computer Science and Information Technologies, CSIT, 326-331 (2017)
33. Vysotska, V., Hasko, R., Kuchkovskiy, V.: Process analysis in electronic content commerce system. In: Proceedings of the International Conference on Computer Sciences and Information Technologies, CSIT 2015, 120-123 (2015)
34. Lytvyn, V., Vysotska, V., Pukach, P., Vovk, M., Ugryn, D.: Method of functioning of intelligent agents, designed to solve action planning problems based on ontological approach. In: Eastern-European Journal of Enterprise Technologies, 3/2(87), 11-17 (2017)
35. Su, J., Sachenko, A., Lytvyn, V., Vysotska, V., Dosyn, D.: Model of Touristic Information Resources Integration According to User Needs. In: International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT, 113-116 (2018)