UDC 621.93

TREE STRUCTURE DATA CHANGE DETECTION METHOD

Yuriy M. Molchanov, Larysa S. Globa, Nikolay O. Alexeyev

National Technical University of Ukraine “Kyiv Polytechnic Institute”, Kyiv, Ukraine

The new method, increasing efficiency and reliability of change detection in three structures in the Internet data under indetermination of data structure (DTD, XML-Schema) is proposed in this paper. The Boolean linear programming problem was solved in two exact methods – modified Balazs with filter and modified DP method and A modified method for selecting a neural network architecture was proposed. There is also considered publish/subscribe system description enhanced with core module, which provides notifications of changes to subscribers only in case they occurred.

Introduction

Nowadays in the Internet environment such interaction scheme between publisher and consumer is spread when every person, who has access to the Internet, at any time may act both as a publisher and as a consumer. This causes information changes tracking issues even within some particular resource or by particular topic as result of continuous unmanageable information update. The need to process large amount of data is required from users to search useful information only because of such conditions. Since it is assumed that the total amount of information on the Internet will increase, the user is forced to continuously increase the time to search for the necessary information.. Obviously, the tool that allows decreasing the amount of processed information while search is required. This should be achieved through automated useful information changes detection and tracking, which user is interested in the Internet

Besides, there are automated systems, which preform mentioned processing based on continuously changing data at regular time intervals, e.g. system of economic processes forecasting based on exchange rates or value of the shares. The approach based on assessing the relevance of data at certain time intervals leads to complication of hardware and software systems of track changes, increased load on the resources that provide information and does not always allow obtaining satisfactory results. That is especially actual for systems that are critical to the delay on the information update.

Mentioned issue is being solved within event-based publish/subscribe systems development, which provide total division in time, space and synchronization between publisher and subscriber. In such systems the information consumer notifies about his interest in some information or about changes at some resources and system concerns the issue of tracing, detection and notification about changes. Although currently publish/subscribe systems have the following disadvantages:

- insufficient informational object definition flexibility – not possible to define separate object or group of objects, which should be tracked, insufficient flexibility of changes representation for end user, limited set of parameters, which can be passed in request, only particular type of changes can be tracked and detected (e.g impossibility to detect insert or update nodes into the document);
- some systems are not Internet-oriented and require predefined information about system topology;
- structured information sources (i.e. (HTML, XML) support lack;
- necessity of additional software installation on publisher and subscriber sides;
- incorrect or confusing presentation of detected changes in documents, etc..

The core of Web-oriented publish/subscribe system is the module change detection in hierarchically structured information, which also provides correct presentation of changes in XML documents (also in HTML as particular case of XML). As generally XML document of any structure can be passed as an input, considered change detection algorithm should operate without information of documents structure (DTD, XML-schema) and support correct detection of insert, delete, update nodes operations, as well as change of structure. The change of one level nodes order is also should be taken into account when considering structure charges, which is not always implemented in existing publish/subscribe systems. Besides, publish/subscribe system requires fast change detection algorithm, which can provide minimal delays in detecting and notification about changes in some particular information.

Thus the objective of change detection approach in hierarchically-structured XML documents in the Internet development is actual, which should include corresponding method, models and algorithms and also flexible technology and tools of program modules of change detection creation for implementation in practice. This will allow us to meet the needs of timely and convenient way of user interest information changes notifications.

Discovery

The publish/subscribe scheme is perspective for development of distributed telecommunication systems, as it proposes 3 distribution levels between publisher and subscriber: in time, in space and synchronization. It was determined that communication system allows reduction of processed information amount and time required for data search, which correspond particular use requirements and can change anytime. Such systems are efficient in the Internet and in distributed information-telecommunication systems, which include large amount of unstructured information as unstructured documents.

The most perspective system to be used in the Internet is content-oriented publish/subscribe system, which does not require additional software, protocols and does not limit number of subscription sources.

Analysis of the functionality of existing publish/subscribe systems shows that the main core of the system is a subsystem of detecting changes in XML documents based on the completion of a certain algorithm that ensures correctness, availability and time to reflect changes in XML documents to the final consumer.

In the analysis of the existing algorithms for detecting changes in XML documents were defined the main types of algorithms: algorithms, which compute minimal editing sequence between all possible the editing scripts [1,2], an algorithm based on hash signatures [4] and semantic algorithms [3, 5]. The conclusion that the considered algorithms have the following disadvantages has been done:

focus on performance, time complexity of changes detection in operations and optimization of received "Delta", but tools and models for the effective implementation of these algorithms in publish / subscribe systems are out of attention;

- do not support all possible types of change detection in structured documents, thus present the resulting changes incorrectly.
- do not support change detection in undefined structured (DTD, XML-schema) documents.
- are time complex for the analysis of large amounts of XML documents.

Thus, it can be concluded the need to develop a detecting changes algorithm in the treelike structures under conditions of uncertainty of tree structure, which is devoid of the above disadvantages and the need to develop publish/subscribe systems based on developed algorithm.

Tree structures change detection

The method for quantitative comparison of nodes in tree structures, taken into account selected parameters of nodes matching. The method essence consists in finding the integral matching criterion of two nodes in tree structures:

$$CS(a_1, a_2) = -1 + 2 \cdot (\alpha \cdot P_{con}(a_1, a_2) + \beta + \gamma \cdot P_{dist}(a_1, a_2)) \quad (1)$$

where α, β, γ - weights,

$P_{con}(a_1, a_2)$ - matching parameter of nodes a_1 and a_2 content:

$$P_{con}(a_1, a_2) = \frac{|\, con(a_1) \cap con(a_2) \,|}{|\, con(a_1) \cup con(a_2) \,|} \quad (2)$$ where ,

$con(a_i)$ - node a_i content .

$P_{att}(a_1, a_2)$ - matching parameter of nodes a_1 and a_2 attributes, which defined as follows:

$$P_{att}(a_1, a_2) = \frac{\sum at_i \in \{at(a_1) \cap at(a_2)\}}{\sum at_i \in \{at(a_1) \cup at(a_2)\}} \quad (3),$$

where $at(a_i)$ - set of node a_i attributes;

$P_{dist}(a_1, a_2)$ - matching parameter of nodes a_1 and a_2 location, which can be found as follows:

$$P_{dist}(a_1, a_2) = \frac{suf\,(index(a_1), index(a_2))}{\max(index(a_1), index(a_2))} \quad (4),$$

where $index(a_i)$ - parameter, which characterizes node a_i location in T tree structure, suf shows length of common suffix for attributes of a_1 i a_2 node, which define location of node in tree hierarchy – between $index\,(a_1)$ i $index(a_2)$ and max defines maximal length of attribute for $index(a_1)$ and $index\,(a_2)$.

The problem of change detection can be solved by finding a "good matching" between the tree structures, i.e. such correspondence between the parts of the original and updated tree structure that the transformation of the original tree in the updated tree requires minimum number of elementary-term operations - insert, delete, move, update of nodes content. The method of "good matching" between elements of tree structures as sum integral criteria maximization problem was suggested. This method reduces the problem of finding "good matching" to the problem of linear Boolean programming, solution of which allows to compare two tree structures to each other even in the face of undefined tree structure.

The mathematical model of finding "good matching" was formulated as follows:

maximize the total value of the integral matching criteria function of two tree structures:

$$\max_{x \in B^n : Ax \le B} \langle C, X \rangle$$, where

$$\langle C, X \rangle = c_{11} \cdot x_{11} + c_{12} \cdot x_{12} + c_{13} \cdot x_{13} + c_{14} \cdot x_{14} +$$
$$+ c_{21} \cdot x_{21} + c_{22} \cdot x_{22} + ... + c_{(n-1)m} \cdot x_{(n-1)m} + c_{nm} \cdot x_{nm}$$

(5)

when:

$$a_{11}x_{11} + a_{12}x_{12} + ... + a_{1m}x_{1m} = b_1$$

…..

$$a_{n1}x_{n1} + a_{n2}x_{n2} + ... + a_{nm}x_{nm} = b_n \quad (6)$$

$$a_{(n+1)1}x_{11} + a_{35}x_{21} + ... + a_{(n+1)(m+1)}x_{n1} \le b_{n+1}$$

….

$$a_{(n+m)1}x_{1m} + ... + a_{(n+m)(n+m)}x_{nm} \le b_{(n+m)}$$

where $\{c_{11},..,c_{nm}\}$ - set of values of integral matching criteria for all nodes of tree structures;

x_{ij} - node connectivity i and j. If i-node matches to j node, then $x_{ij} = 1$. If i does not match j ,then $x_{ij} = 0$;

A and B - conditions matrixes, which have the following values:

$$A = \begin{pmatrix} 1 & 1 & 1 & ... & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & ... & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & ... & 1 & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 1 & ... & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & ... & 0 & 0 & 0 & 1 \end{pmatrix} \quad (7)$$

$$B = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad (8)$$

The change detection method in the tree structures that includes nine steps (Figure 1), to detect the changes between the original and the updated tree structure was proposed.

The original and updated tree structures are built in the first step on the basis of the web pages. Then both versions of the tree structures can be considered as inputs for the problem of change detection. The second step is indexing of tree structures needed to take into account the order of nodes (left-right) in the resulting tree and tracking node location in the hierarchy tree.

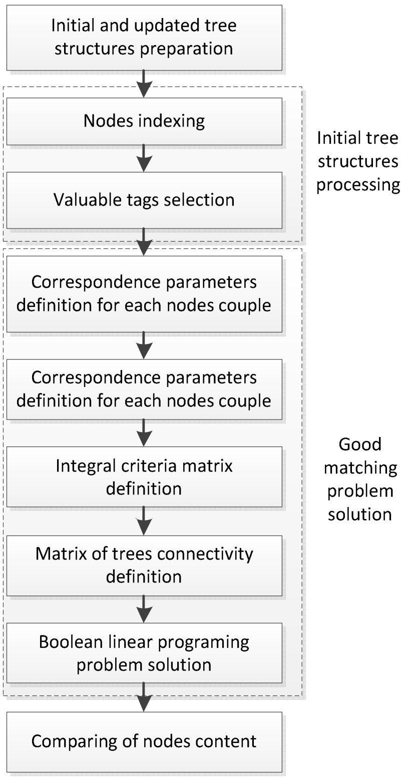

Initial and updated tree structures preparation
Nodes indexing
Valuable tags selection
Initial tree structures processing
Correspondence parameters definition for each nodes couple
Correspondence parameters definition for each nodes couple
Integral criteria matrix definition
Matrix of trees connectivity definition
Boolean linear programing problem solution
Good matching problem solution
Comparing of nodes content

Fig. 1. Trees structures change detection method

The third step is important for search nodes which are valuable for user in the resulting tree structure that will be considered in change detection algorithm. The fourth step calculates the eligibility criteria for each pair of nodes in tree structure. The matrix of integral matching criteria for all selected nodes in both versions of the XML tree is created in the fifth step, and matrix of connectivity creation is the sixth step of the method. These matrixes are used in the seventh step to formulate Boolean linear programming problem. The eighth step is the solution of linear programming problem using one of the proposed mathematical methods. The integer solution will be found during solving the problem which will satisfy all the set of conditions. This solution will determine the mutual correspondence between nodes of the old and new versions, which can be compared by contents on the ninth step.

“Good matching” search between tree structures

“Good matching” problem is formulated as follows. The set of tasks W and given relations: for each pair of tasks $\omega \in W$ some subset of its predecessors P is defined. Two trees $T_1 = T_1$ is given (a_i , N_1 , AT , $con(a_i)$)| $i = \overline{1,n}$ and $T_2 = T_2$ (a_j , N_2 , AT , $con(a_j)$)| $j = \overline{1,m}$, where N_1 , N_2 - nodes number in corresponding trees. The following problem must be solved: $\max\limits_{x \in B^n : Ax \leq B} \langle C, X \rangle$.

In considering the complexity of the solution of the linear programming problem should be noted that this problem requires a solution only in Boolean variables. Thus, it is necessary to look for solution of the Boolean linear programming problem (BLP). These tasks are formulated as the problem of additional restrictions LP variables in Boolean domain. In general this problem belongs to NP-hard problems. Thus, use of sorting schemes to solve this problem is valid. In addition, for certain types of BLP problems there are other methods of solution: accurate - branch and bound method, the method of dynamic programming, Balazs method with filter, inaccurate - the method of random search, genetic local search. However, the computational complexity of existing methods of finding solution to the BLP problem does not meet the needs of the publish/subscribe efficiency in finding corresponding nodes in trees with a large number of nodes, so it was decided to modify the known methods for solving the BLP problem by taking into account the specialties of the tree structures change detection domain to ensure minimal computational complexity with all requirements to find correspondence between nodes of the tree structures.

The Boolean linear programming problem was solved in two exact methods - modified Balazs with filter and modified DP method. Consider the basic modifications that were used in this paper for these mathematical methods.

The modified Balazs method with filter, which introduced thresholds and limit of the units number in the sorting options has complexity:

$$O\left(\frac{(mn - p - q)!}{(m - q)!(mn - p - m)!}\right) \quad (9),$$

where p - nodes number, for which integral criteria value is less then $c_{ij} < c_{\min}$, and q - nodes number, for which integral criteria value is more then $c_{ij} > c_{\max}$; m - number of nodes in updated tree i, n - nodes number in initial tree. This method is convenient to use when there are not much changes between the trees when the values p and q significantly impact the resulting complexity of this method.

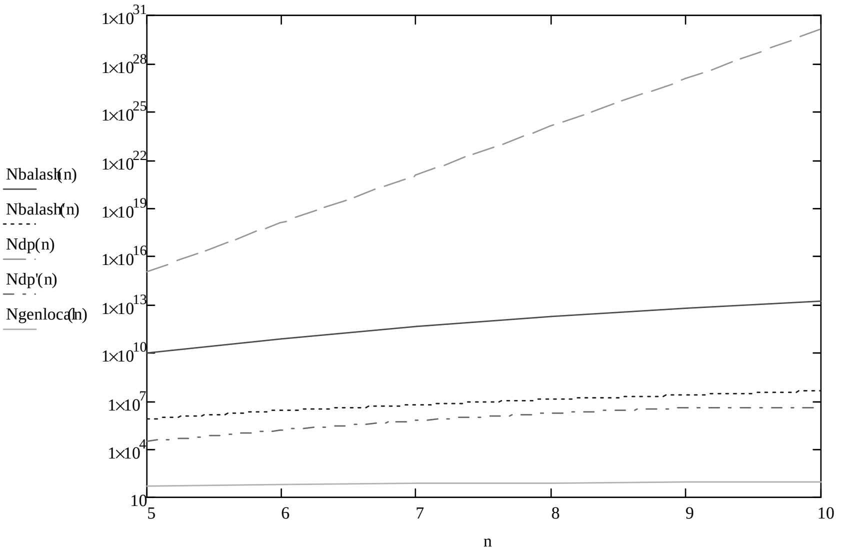
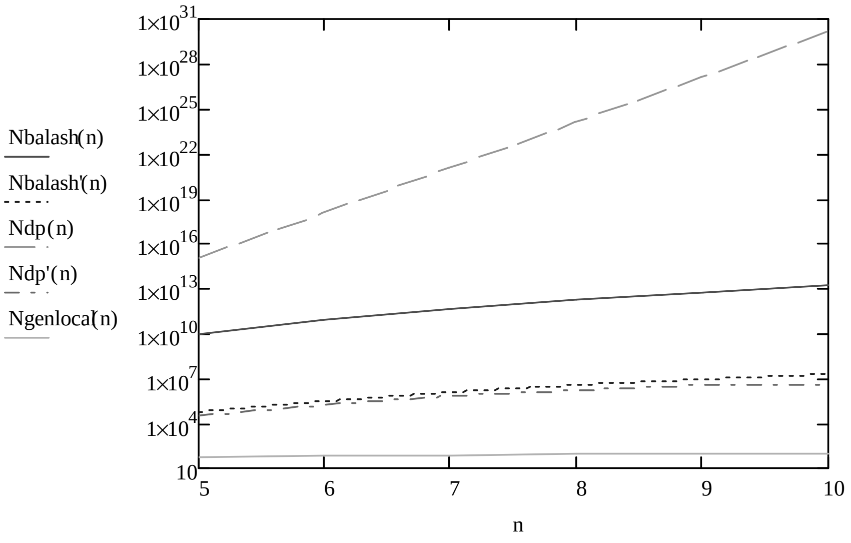
In the modified method of DP was introduced restriction of mutual nodes matching between two trees. Considering these modifications was shown that the complexity of this method is $O\left(\frac{(m)!}{(m-n)!}\right)$ (10) .

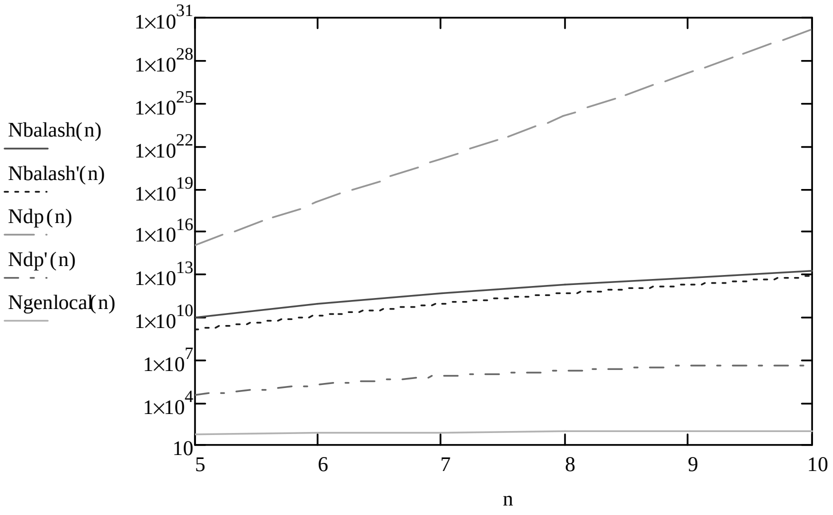
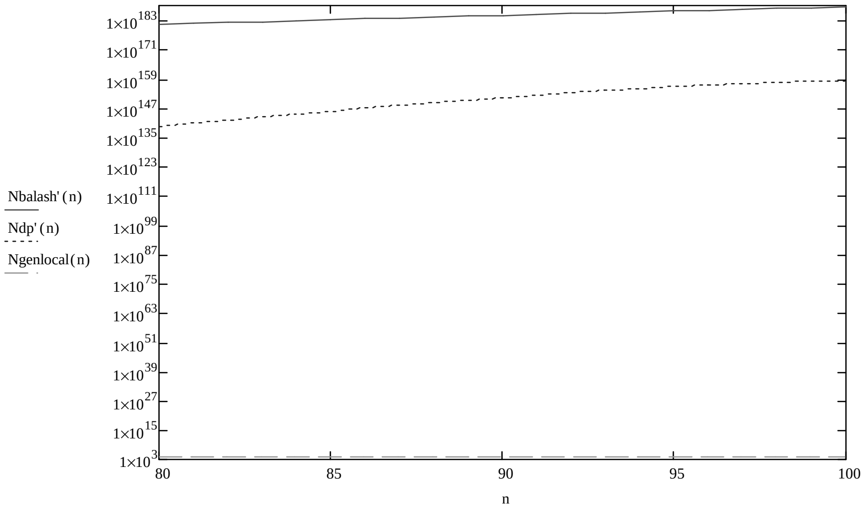
The resulting computational complexity is less comparing to the modified Balazs method with filter, but if there are many leaves in the T_1 and T_2 trees search for the acceptable solution is not possible. Accordingly, in this situation would be appropriate to use approximate methods.

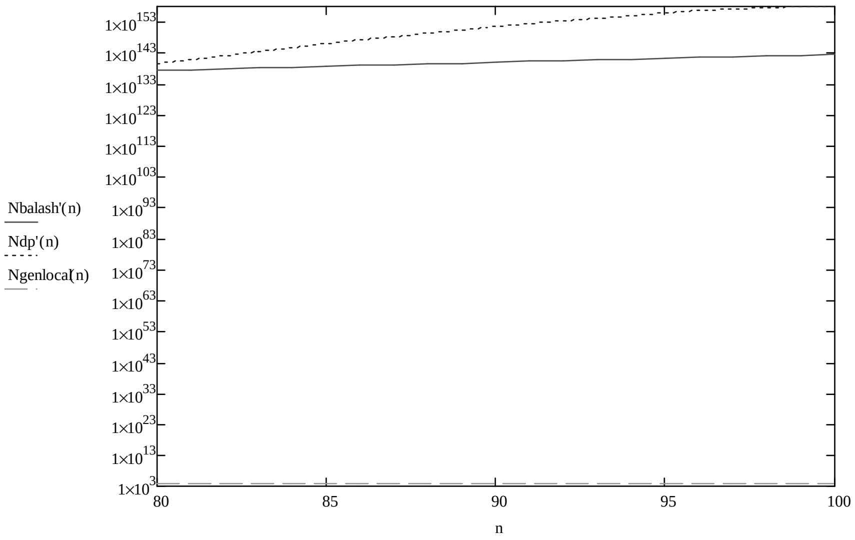
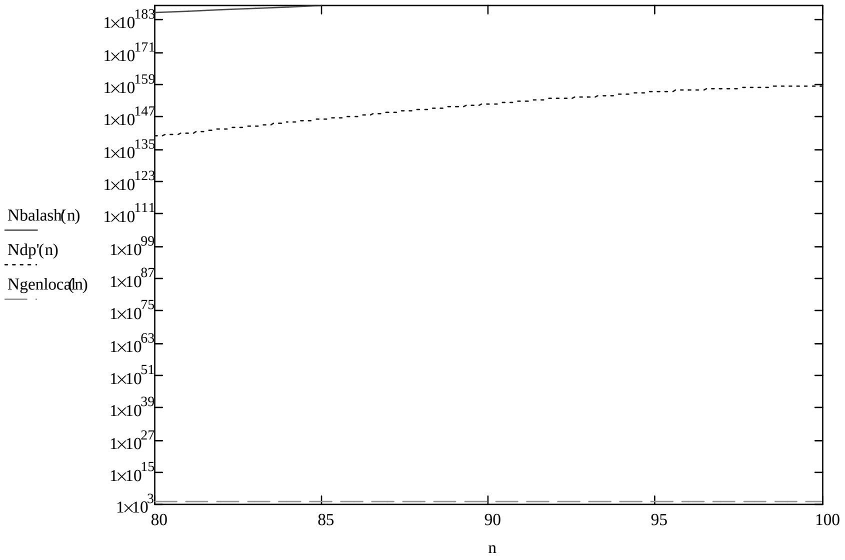
Used modifications allowed us to reduce the computational complexity of these methods, but it remained higher than the existing methods for detecting changes. Thus decision to use method was done.

The inaccurate methods for solving the BLP problem [13] are local search, random search and simultaneous use of several approximate methods, for example, the problem is solved by local search, but not once but many times to many different starting points, selectable at random. Such methods can solve the problem of large-scale linear constraints of any complexity, and within a reasonable time. Among the methods for solving subproblems locally the method of genetic local search is one of the most-effective and the most frequently used for of integer programming. Introduced algorithm is based on a genetic local search method, reduces the computing complexity to $O(m^2 + n^2)$ for the worst case.

The comparison of methods for solving the problem of finding "good matching" in the tree structure is presented in Table 1.

Tree size	Content attributes changes characteristics	Graph of the number of steps required to obtain the result depending on the number of nodes in the updated tree
Small trees (up to 10 nodes)	Random content attributes change	
	Content attributes change, not more 50%	

	Content attributes change, more 50%	
Average trees up to 100 nodes in tree)	Random content attributes change	

	Content attributes change, not more 50%	
	Content attributes change, more 50%	

Based on the comparison of the proposed change detection algorithms in tree structure, it was decided to choose a method for solving the problem based on specified conditions in every particular case. Thus, the recommendations of method selection for solving the BLP problem are as follows:

1. If the content attributes of tree node were changed less than 50% (minor changes) and small trees (up to 10 nodes), then suggested to use the modified Balazs method with filter.
2. If there is a comparison of trees of small size (10 nodes), while content attributes of nodes have been significant changed (more than 50%), suggested to use the modified DP method.
3. For the general case – when the trees contain a large number of nodes (more than 10) regardless of the number of content attributes changes to use genetic local search will be reasonable.

To compare the proposed method with the existing algorithms of change detection the method of local optimization – as the most common method of solving the problem of given problem was selected. In Fig. 3 the comparative characteristics of these methods of XML documents comparing using different approaches is shown.

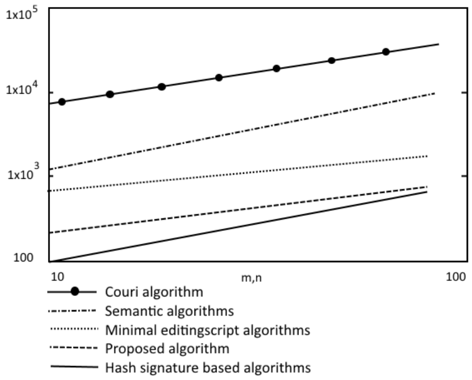


Fig. 3. Comparative characteristics of evaluated methods

Comparing the proposed method of solving the problem finding correspondence between XML documents with other algorithms has been shown that the proposed algorithm is efficient in terms of time complexity. This is achieved proofing possibility to consider the problem of finding the correspondence between XML documents as Boolean linear programming.

Compared the effectiveness of existing algorithms for change detection in the tree structures based on the criteria of time complexity and was shown that the proposed method of local optimization is more efficient than existing methods. Compared with the existing most efficient method, computational complexity has been reduced in $2nm$ steps, where n and m - corresponding number of nodes in the initial and updated tree structures.

The optimal information source request interval determination, depending on the frequency of its update.

This problem was formulated as follows: to predict the optimal requesting of page to minimize the time between page refreshes and detection of changes. The method of processing a request with correction of class was suggested, which is as follows.

Initially the publish/subscribe system receives data from sources such as page size, visiting rating of sites, the presence of certain keywords in the description of the site last updated information in the description of the site, PageRank, number of links to other resources.

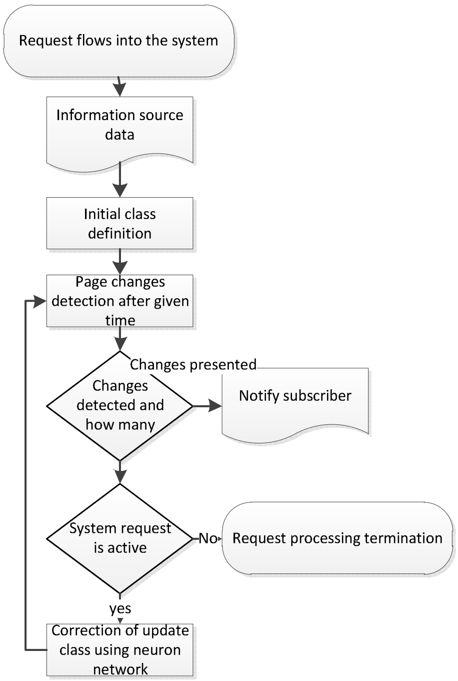

Request flows into the system
Information source data
Initial class definition
Page changes detection after given time
Changes detected and how many
Changes presented
Notify subscriber
System request is active
No
Request processing termination
yes
Correction of update class using neuron network

Fig.4. Block diagram of the processing request with correction of update class

Based on these data, a decision on the initial page update class and initial change detection interval is defined. After each subsequent change detection, the correction of updates class is performed based on preassembly and new data: the number of changes in the source of information over time, the number of changes at the last inspection of site, the number of recent "empty" inspections of sources. After correction of update class a new interval of detecting changes is defined and request goes into standby.

This procedure is repeated until the request is active in the system, i.e. request is not deleted or changed in the system. Block diagram of the method is shown in Fig. 4.

It was decided to choose the neural networks of direct distribution, as mathematical approach to solve the problem of forecasting update class. Modified method of neural network architecture choice to minimize the number of neurons in the input and inner layers while keeping a given level of learning error was proposed in this paper. This modified method will reduce time of class update correction, and therefore the processing time of one request for the change detection in in publish / subscribe system and amount of data stored for each request.

A modified method for selecting the neural network architecture consists of the following steps:

1. Building a neural network with preselected architecture using existing methods.
2. Training of the backpropagation neural network.
3. Definition of "importance" of each neuron in the input and inner layer based on the following formulas.

   The importance of the input layer neuron:

   $$I_i = \frac{\sum_{k=1}^{H}(w_{ik})^2}{N_H} (11)$$

   The importance of the inner layer neuron:

   $$I_h = \frac{\sum_{k=1}^{O}(w_{hk})^2}{N_O} (12)$$
4. Determination of minimal importance neuron and its exclusion from the neural network.
5. Determination of network errors E_{mod} on the testing sequence and compare it with E_o, determinate for given neuron. Repeat steps 2-4, until $E_{mod} \leq E_o$.

The proposed method of processing a request with update of correction class of information sources allows us to find the optimal interval for updating any sources and provide subscriber information with minimal delay

Conclusion

The efficiency of proposed modifications has been proven by analyzing the computational complexity of proposed and existing methods - detection time of changes in small trees (10 nodes) was reduced in the modified Balazs method from 181 ms to 0.22 ms, and in the modified method of dynamic programming from142 to 0.01 ms.

Genetic local search method for solving the problem of Boolean linear programming problem based on specifics subject area of change detection in tree structure, allowed satisfactory time complexity of solving the BLP problem while maintaining the reliability of the results: in the case of detecting changes in medium-sized trees (10-100 nodes) allowed to reduce the minimum time of accurate detection methods from 384 s to 6.72 ms for this method.

The efficiency of the chosen method of modified genetic local search algorithm and existing change detection methods in the tree structures based on the criterion of computational complexity was evaluated and was shown that the proposed method of genetic local search is more efficient than existing methods, especially compared to the most efficient method of Chang and Shasha, computational complexity has been reduced by 24.7% for the average web pages on the Internet.

A modified method for selecting a neural network architecture allowed by optimizing the architecture to reduce the computational complexity of training neural network by 13%, reduce by 11% the number of requests to internal and external systems, as well as the amount of information that should be stored for a query, average number of steps to train the neural network decreased by 93% and is 670 steps while maintaining the chosen level of errors.

References

[1] S. Chawathe, H. Garcia-Molina, Meaningful change detection in structured data, in: Proceedings of the ACM, SIGMOD International Conference on Management of Data, Tuscon, Arizona, May 1997, pp. 26–37.

[2] S. Chawathe, S. Abiteboul, J. Widom, Representing and querying changes in semistructured data, in: Proceedings of the International Conference on Data Engineering, Orlando, Florida, February 1998, pp. 4–13.

[3] S. Flesca, E. Masciari Efficient and affective Web change Detection

[4] Change Detection of XML Documents Using Signatures. Latifur Khan, Lei Wang and Yan Rao Department of Computer Science University of Texas at Dallas, 2003.

[5] Küster, U., H. Lausen, and B. König-Ries, Evaluation of Semantic Service Discovery—A Survey and Directions for Future Research. Emerging Web Services Technology, 2008. 2: p. 41-58.

[6] Keyvan M., Suhaimi I., Mojtaba K., Kanmani M., Sayed G. H. T.i, A comparative evaluation of semantic web service discovery approaches, Proceedings of the iiWAS2010, November 08-10, 2010, Paris, France.

[7] J. Jyoti, A. Sachde, and S. Chakravarthy, “CX-DIFF: A Change Detection Algorithm for XML Content and Change Visualization for WebVigiL,” Data and Knowledge Eng., vol. 52, no. 2, pp. 209-230, 2005.

[8] H. P. Khandagale and P. P. Halkarnikar. A Novel Approach for Web Page Change Detection System. International Journal of Computer Theory and Engineering, Vol. 2, No. 3, June, 2010.

[9] Classification of Web-based publishing systems and subscription of information resources / Yu.N. Molchanov, N.A. Alekseyev, L.S. Globa, Feldman Marius // Visnyk of V.N. Karazina Kharkiv National University, series “Mathematical Modeling. Information Technology. Automated control systems”. – Vol. 9, № 809, 2008. – 3-8pp.

[10] Larisa Globa, Mykola Alieksieiev, Iurii Molchanov, Liudmyla Kobzar /XML Documents Change Detection System, ACS, 2010.

[11] Larysa Globa, Mykola Alieksieiev, Iurii Molchanov / XML DOCUMENTS CHANGE DETECTION SYSTEM BASED ON BOOLEAN LINEAR PROGRAMMING TASK, Metody Informatyki Stosowanej, Szczecin, 2/2011 (27).

[12] A.N. Antamoshkin, I.S. Masich Efficient algorithms of constrained optimization of monotone pseudoboolean functions. Vesnik of SibSAU. Issue 4 – Krasnoyarsk: SibSAU, 2003.

Received in final form October 23, 2014