The Partial Mapping of the Web Graph

Machová Kristína
Faculty of Electrical Engineering and Informatics
Technical University
Letná 9, 04200 Košice, Slovakia
kristina.machova@tuke.sk

Gaľová Lenka
Faculty of Electrical Engineering and Informatics
Technical University
Letná 9, 04200 Košice, Slovakia
galoval@gmail.com

Abstract. *The paper presents an approach to partial mapping of a web sub-graph. This sub-graph contains the nearest surroundings of an actual web page. Our work deals with acquiring relevant hyperlinks of a base web site, generation of adjacency matrix, the nearest distance matrix and matrix of converted distances of hyperlinks, detection of compactness of web representation, and visualization of its graphical representation. The paper introduces an LWP algorithm – a technique for hyperlink filtration. This work attempts to help users with the orientation within the web graph.*

Keywords. Web structure mining, web graph, web mapping, hyperlink filtration

1 Introduction

Internet seems to be the most frequently used medium and web presentations (a set of web pages devoted to some topic) become the basic source of information. This information is accessible directly from web pages, particularly from their contents. During the process of obtaining information (particularly indexing, classification and other similar operations), mainly content analyzing is performed. But content is not the only information carrier. Another information carrier can be represented by a structure of a presentation. Information can be obtained also from the way in which pages of web presentations are connected. We studied such information sources through analyzing web pages topology.

The topology analysis of web pages is used also in [1] as a supporting tool for web pages content classification. The aim is to remove those parts of a presentation, which do not contain useful information. This work does not focus explicitly on the structure of web presentations.

On the other hand, Mathieu and Viennot [2] describe the structure of the Web in two ways: by references and by a file hierarchy generated from URL analysis. Disadvantage of this access is the assumption of respecting rules for placement of files into directories according to the position in a reference topology. These rules are obviously being ignored in practice.

Thus, our approach is connected with the structure of the Internet, particularly with existing hypertext references. The technology of hypertext references forms a basis of the structure of the Web. The references and their use allow the mutual interconnection of pages. The result is the formation of a net-like structure in which any orientation is usually very difficult.

During searching the Internet, a user often visits a web page, which does not contain the required information. Thus, he/she must backtrack to some previously visited page but he/she does not remember the way back to the starting page. Therefore, we have tried to help the user with solving this situation with the aid of a partial mapping of the web graph. This partial mapping represents a map of the nearest surroundings of the actual page. This map can help the user to orientate while searching the Internet.

Naturally, the most appropriate form of this map is a graph. Nodes of this graph can be pages of a web presentation. Edges can be clicks, which enable the movement between pages (nodes). Such click can be uncovered in the form of a relevant hyperlink on the actual page. The graph can be constructed through considering all relevant hyperlinks directed from the actual web page or eventually directed to the actual web page. Considering all of these links is meaningful to only such level, in which the web graph is still transparent and readable.

We suppose the first level of nesting be represented by the opening of an actual page. Our approach is based on the identification of the URL addresses contained on the actual web page to obtain information about next pages - the nearest neighbors of it. In other words, nodes surrounding the actual page are identified. This process represents the second level of nesting (direct access). On the third level of nesting (indirect access), there are more neighbors (of neighbors) and therefore the partial web graph is larger and more complicated. It is possible to continue

to higher nesting levels - but the corresponding web graph will be so complicated that it will not help users with orientation any more.

The first thing we have to do is to preprocess data, which HTML code contains. This preprocessing stage is described in section two. Section three is devoted to generation of an adjacency matrix from the preprocessing data step, generation of nearest distance matrix and matrix of converted distances of hyperlinks as well as to the detection of compactness of web presentations. Section four introduces a visualization process to produce a graphical representation. Section five describes our experiments focused on global representation with the second level of nesting (section 5.1), complete representation with the second level of nesting (section 5.2), and local representation with the third level of nesting (section 5.3).

2 Preprocessing stage

The whole partial mapping process starts with a preprocessing stage, in which data – pages with some html code - are saved into files and filtered to obtain relevant hypertext links. We have designed an LWP method. The LWP method uses a console application "lwp-download" in the data preprocessing stage. The LWP method and generation of all needed matrixes were programmed in the programming language ActivePerl 5.8.7 Build 815.

An actual web page is expected to be defined by a user in the form http://www.tuke.sk, www.tuke.sk or http://www.tuke.sk/. Similarly, the level of nesting is the parameter of the algorithm. Subsequently, a form of a partial web graph presentation must be selected. The following three selections are possible:

1. global presentation (only pages from other domains than the actual web page domain is will be presented)
2. local presentation (only pages from the same domain as the actual web page domain is will be presented)
3. complete presentation (pages from all domains will be presented).

At the beginning of the LWP method, the URL of the actual (root) page is put into a new file, which will contain all URLs of all web pages on all levels of nesting. This file will be used for generation of an adjacency matrix. Our algorithm is looking for some specific strings (HREF, FRAME, SRC) which represent links from the web page currently being processed. In the next step, filtering of irrelevant links as well as multiple occurrence detection of links is performed. The algorithm does not consider the following links: JAVASCRIPT:, FILE, PL, C, DOC, MP3, JAR, TXT, PDF, AVI, XLS, ZIP, RAR, PPT, ICO, GIF, CSS, RTF, JPG, EXE, PNG and e-mail adresses. Page URLs with the following extensions: html, htm, shtml, php and xml are transformed into the form, to which another reference can be joined. A root page represented in different forms, for example a page in the forms www.lonearrow.com and www.lonearrow.com/index.html, is recognized as the same.

The proccess of gradual nesting is solved within a cycle. The first level of nesting is represented by opening a root page. The second and subsequent nesting levels are performed in iterations. In each iteration, all previously found decsendants (links discovered in the previous iteration) become ancestors (new pages to be proccessed).

All relevant links found on the last level of nesting are leaf pages. The next task is to detect wheter these leaf pages point to the web pages which were already preprocesed. Althout all links on leaf pages are identified, only those links are registered, which point back to the pages preprocesed before. The step must be done in order to generate a complete adjacency matrix. After leaf pages preprocessing, an output text file "odkazy.txt" is obtained. An example of such output file containing data after preprocessing which corresponds to the web page http://www.kukfuk.sk is presented in Figure 1.

```
0 http://www.kukfuk.sk/
1 http://www.kukfuk.sk/o_firme.html
2 http://www.astudio.sk
3 http://www.kukfuk.sk/certifikat.html
4 http://www.kukfuk.sk/produkty.html
5 http://www.kukfuk.sk/foto_firmy.html
6 http://www.kukfuk.sk/kontakt.html
7 http://www.kukfuk.sk/obrat.html
8 http://www.kukfuk.sk/ochranna_znamka.html
9 http://www.astudio.sk/index.sk.html
10 http://www.astudio.sk/index.en.html
```

Figure 1. Output file "odkazy.txt" corresponding to the web page http://www.kukfuk.sk

3 Matrix generation stage

A partial graph of the Web can be constructed from the output file after preprocessing. Before the web graph construction, we have to generate an adjacency matrix from the preprocessed data, nearest distance matrix and matrix of converted distances of hyperlinks. General rules for generating all these matrixes from a graph can be found in [6]. We have adapted definitions of all the needed matrixes for using in the field of the Internet in thefollowing way.

The adjacency matrix. Each matrix component can be:

x_{ij} = 1, web page i contains a link to page j,
x_{ij} = 0, web page i does not contain any link to page j,
x_{ij} = S, i=j, web page i contains link to itself

where i and j are indexes of various web pages.

The adjacency matrix serves for generation of the nearest distance matrix by Floyd – Warshall algorithm [6]. The pessimistic estimation *E* of its complexity *C* is according to [7]:

$$C(n) = E(n^3 \log_2 n) \quad (1)$$

where *C* is complexity of the partial graph, *E* is pessimistic estimation and n is the number of nodes – web pages.

The nearest distance matrix seems to be the most important matrix, which represents the shortest paths between web pages.

The nearest distance matrix. Each component of this matrix can be:

x_{ij} = k, the nearest distance from web page i to web page j is k,

x_{ij} = 0, holds for components in diagonal where i=j (but some indirect way from i to j can exist),

x_{ij} = S, holds for components in diagonal where i=j in the special case when the web page references itself,

x_{ij} = N or inf, there is no possibility to achieve page j from page i, (distance between pages is ∞),

where i and j are indexes of various web pages.

We had to modify the Floyd – Warshall algorithm, because for our purposes the value zero is not an ideal nearest distance. So, value zero was replaced with another value (5000) and value "1" was set the shortest distance. The nearest distance matrix is illustrated in Figure 2 where IDs identify unique URLs and "Inf" represents an infinite indirect access. For example, the cell (1,6) contains number 2, which means that from page 1 we can access page 6 within 2 steps along the graph's path.

ID	URL	1	2	3	4	5	6
1	http://www.cmiinsulation.com/	0	1	1	1	1	2
2	http://www.cmiinsulation.com/service/service.htm	inf	0	1	1	1	1
3	http://www.cmiinsulation.com/products/products.htm	inf	1	0	1	1	1
4	http://www.cmiinsulation.com/eeo/eeo.htm	inf	1	1	0	1	1
5	http://www.cmiinsulation.com/company/company.htm	inf	1	1	1	0	1
6	http://www.cmiinsulation.com/index.htm	inf	inf	inf	inf	inf	0

Figure 2. The matrix of the shortest paths

The matrix of converted distances is a transformed matrix of the nearest distances, where all diagonal components have value zero. Similarly, all components with value N or inf are replaced with a conversion constant – the number of pages in the nearest distance matrix.

The matrix of converted distances is essential for the calculation of the measure of the web page compactness. A standard conversion constant is the number of nodes in a web presentation - it means the number of web pages. We used the number of pages in the adjacency matrix in the role of a conversion constant in compactness calculation. The compactness of a partial web graph *Cp* is defined in the following way:

$$Cp = \frac{Max - Sum}{Max - Min} \quad (2)$$

Where:

Max is the maximum possible value of a sum of converted distances:

$$Max = d^2 * (d - 1) \quad (3)$$

d is conversion constant.

Sum is the actual sum of all distances in the matrix of converted distances.

Min is the minimum possible value of sum of converted distances:

$$Min = d * (d - 1) \quad (4)$$

The measure of the compactness can be from the interval <0,1>. The value zero represents the situation, when pages do not contain any links to other pages. Compactness equal to the value "1" represents the situation, when all pages contain links to all the other pages. A reasonable value of a web page compactness is "0,5". Usually, web presentations of various companies have compactness closer to value "1".

The output file of the matrix generation stage is a file "vystup_matica.txt" and it contains a list of web pages with indexes, adjacency matrix, the nearest distance matrix, matrix of converted distances, values needed for compactness measure calculation and the value of the compactness of a partial web graph.

4 Web graph visualization

There are many possibilities to visualize a partial web graph. We tried to utilize two available tools: Graphviz [4] and Prefuse Visualization Toolkit [5]. The outputs from the file "vystup_matica.txt" were converted into the form required by a selected graphical presentation.

4.1 Export of outputs into Graphviz

The Graphviz uses the followed schemes: *Dot* (hierarchical or layered oriented graph), *Neato and Fdp* (spring model scheme), *Twopi* (beam scheme), *Circo* (cycle scheme). The most suitable scheme for our purposes was the *Dot* scheme, because this scheme creates hierarchical or layer oriented graphs and this scheme tries to avoid edge crossing and to

reduce length of edges. We have generated names of graph nodes in two ways.

In the first case, names of nodes were created from URLs, but local links were shortened. The part of a URL, which represents root page was deleted.

Below names of nodes, also the number is visualized. This number represents the nearest distance of the node (page) from the root node (root page). The illustration of the code follows:

```
digraph G {
0 -> 1;0 [label="http://www.hrusinsky.cz/\n0"];
        1 [label="indexa.htm\n1"];
1 -> 2;1 [label="indexa.htm\n1"];
        2 [label="side.htm\n2"];
1 -> 3;1 [label="indexa.htm\n1"];
        3 [label="uvod.htm\n2"];
3 -> 3;3 [label="uvod.htm\n2"];
          3 [label="uvod.htm\n2"];
```

The attribute *label* represents the name of a corresponding node as well as the nearest distance (\n0, \n1, …).

A graphical representation of this graph in Graphviz is shown in Figure 3.

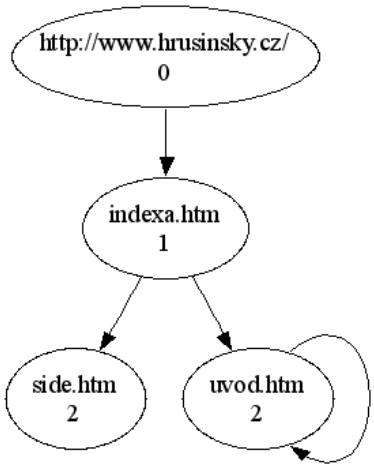

Figure 3. Graphical representation in Graphviz

In the second case, names of nodes were represented only by their indexes for more simple visualization. The illustration of the code has the following form:

```
digraph G {
    0 -> 1;
    1 -> 2;
    1 -> 3;
    }
```

4.2 Export of outputs into Prefuse VT

We tried to use another visualization tool – Prefuse Visualization Toolkit - because visualization with Graphviz does not seem to be transparent enough, mainly for large web presentations. The input format for graph visualization is written in XML. At the beginning, it was needed to set the "*directed*" value for the attribute "*graph edgedefault*". Also three data elements have to be defined: *data schema, node* and *edge*. In the following code, our definitions are defined:

```
<graph edgedefaul="directed"/>
<!-- data schema -->
    <key id="name" for="node"
    attr.name="name" attr.type="string"/>
<!-- nodes -->
<node id="1">
    <data key="name">
    http://neuron.tuke.sk/~szaboova/</data>
</node>
<!-- edges -->
    <edge source="1" target="2"></edge>
    <edge source="1" target="3"></edge>
```

The visualization process started from the page [8]. For illustration and comparison, the graph from Figure 3 is represented in another tool – Prefuse - in Figure 4.

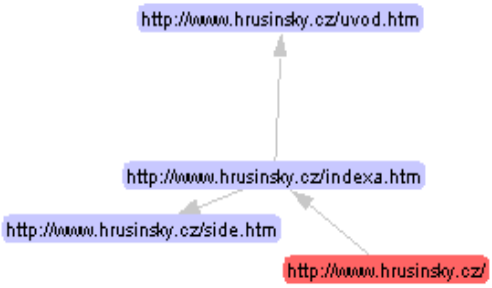

Figure 4. Graphical representation in Prefuse

5 Experiments

Our experiments were focused on the global representation of a partial web graph with the second level of nesting, the complete representation of a partial web graph with the second level of nesting, and the local representation of a partial web graph with the third level of nesting. All these experiments were provided on the base of output file "odkazy.txt"

corresponding to the web presentation http://www.kukfuk.sk, which is illustrated in Figure 1.

5.1 Global representation with the second level of nesting

The global presentation contains only pages from other domains than the domain from which the actual web page is (http://www.kukfuk.sk in our case). All pages of this presentation can be found in Figure 5.

```
0 http://www.kukfuk.sk/
1 http://www.astudio.sk
2 http://www.astudio.sk/index.sk.html
3 http://www.astudio.sk/index.en.html
```

Figure 5. List of pages of global representation

Table 1: The adjacency matrix of the global representation with the second level of nesting

	0	**1**	**2**	**3**
0	0	1	0	0
1	0	0	1	1
2	0	0	S	0
3	0	0	0	S

Table 1 illustrates the adjacency matrix of global representation generated from the preprocessed data in Figure 5. Table 2 represents subsequently generated the nearest distance matrix (part a) and the matrix of converted distances (part b). Because of simplicity, only indexes, not all names of pages are depicted in the tables.

Table 2: The nearest distance matrix (part a) and matrix of converted distances (part b) both with the second level of nesting

a)					**b)**				
	0	**1**	**2**	**3**		**0**	**1**	**2**	**3**
0	0	1	2	2	**0**	0	1	2	2
1	N	0	1	1	**1**	4	0	1	1
2	N	N	S	N	**2**	4	4	0	4
3	N	N	N	S	**3**	4	4	4	0

The following values were obtained from the matrix of converted distances:
Sum = 35
Min = 12
Max = 48
Cp = 0.36

On the base of the nearest distance matrix, the graph construction was performed using two visualization tools (Figure 6 and Figure 7).

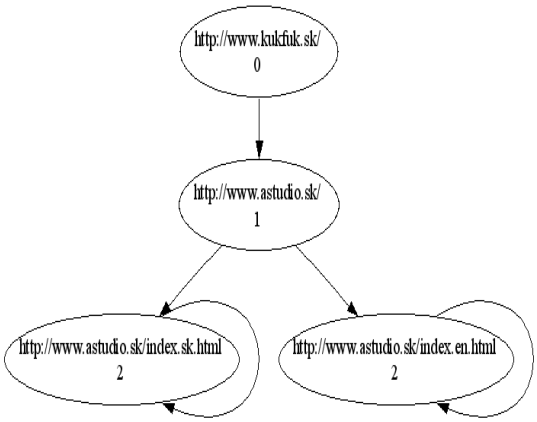

Figure 6. The graph in the Graphviz with the second level of nesting

Measure of compactness *Cp = 0.36* is lower than "0.5". So graphs in Figures 6 and 7 are not very compact.

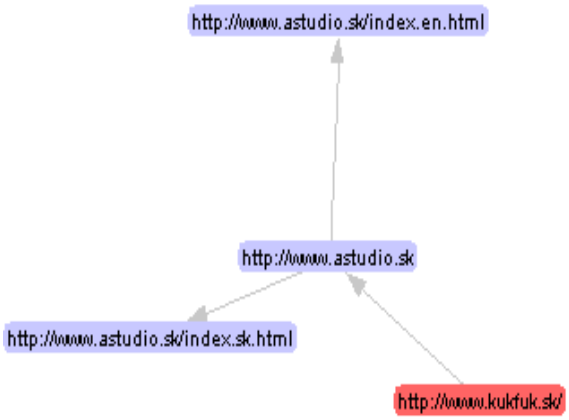

Figure 7. The graph in the Prefuse with the second level of nesting

As it is possible to see, the Prefuse visualization does not represent cycles when a page contains references to itself.

5.2 Complete representation with the second level of nesting

The complete representation contains pages from all domains. All pages of this presentation can be found in Figure 1. Table 3 illustrates the adjacency matrix of global representation generated from the preprocessed data in Figure 1.

Table 3: The adjacency matrix of complete representation with the second level of nesting

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	0	1	1	0	0	0	0	0	0	0	0
1	0	S	1	1	1	1	1	1	1	0	0
2	0	0	0	0	0	0	0	0	0	1	1
3	0	1	1	S	1	1	1	0	0	0	0
4	0	1	1	1	0	1	1	0	0	0	0
5	0	1	1	1	1	0	1	0	0	0	0
6	0	1	1	1	1	1	0	0	0	0	0
7	0	1	1	1	1	1	1	S	1	0	0
8	0	1	1	1	1	1	1	1	S	0	0
9	0	0	0	0	0	0	0	0	0	S	0
10	0	0	0	0	0	0	0	0	0	0	S

Table 4 represents subsequently generated the nearest distance matrix. The matrix of converted distances of complete representation with the second level of nesting is similar to the matrix in depicted in Table 4. The only difference is that value “N” is replaced by value “11”.

Table 4: The nearest distance matrix of complete representation with the second level of nesting

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	0	1	1	2	2	2	2	2	2	2	2
1	N	S	1	1	1	1	1	1	1	2	2
2	N	N	0	N	N	N	N	N	N	1	1
3	N	1	1	S	1	1	1	2	2	2	2
4	N	1	1	1	0	1	1	2	2	2	2
5	N	1	1	1	1	0	1	2	2	2	2
6	N	1	1	1	1	1	0	2	2	2	2
7	N	1	1	1	1	1	1	S	1	2	2
8	N	1	1	1	1	1	1	1	S	2	2
9	N	N	N	N	N	N	N	N	N	S	N
10	N	N	N	N	N	N	N	N	N	N	S

The following values were obtained from the matrix of converted distances of complete representation with the second level of nesting:
Sum = 490
Min = 110
Max = 1210
Cp = 0.654

On the base of the nearest distance matrix of complete representation with the second level of nesting (Table 4), the graph construction was done in the Graphviz visualization tool. The resulting graph can be seen in Figure 8. The compactness of this graph is quite sufficient (*Cp = 0.654)*. In Figure 8, names of local nodes, which belong to root page of the same domain, are shortened to make the graph more transparent. But names of global nodes from other domains (different from the domain of the root page) are represented in the complete form. Each node contains also a number, which represents the nearest distance of this node from the root node.

We tried to represent the graph from Figure 8 also in Prefuse Visualization Toolkit in the demo Java aplet accesible from the web page [8]. The result can be seen in Figure 9. Nodes in this graph are represented in the complete form - in order to enable double click on the nodes of the graph to open coresponding web pages in a web broser.

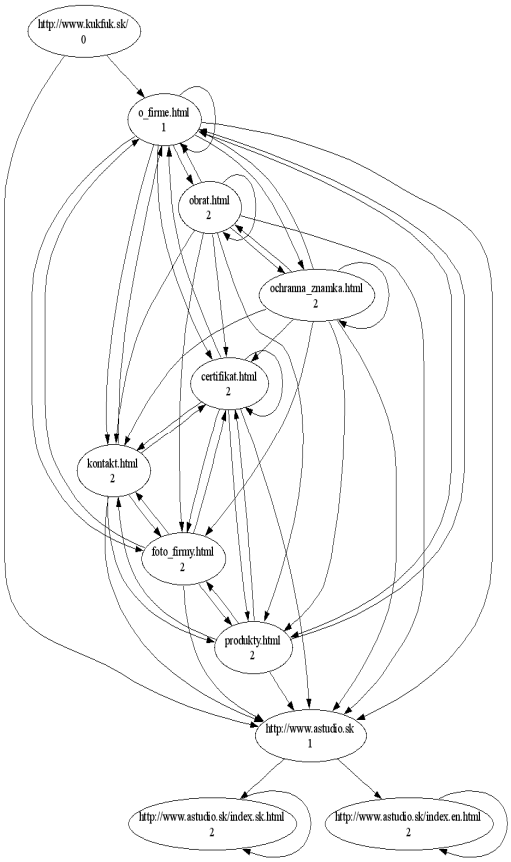

Figure 8. The graph of complete representation in the Graphviz with the second level of nesting

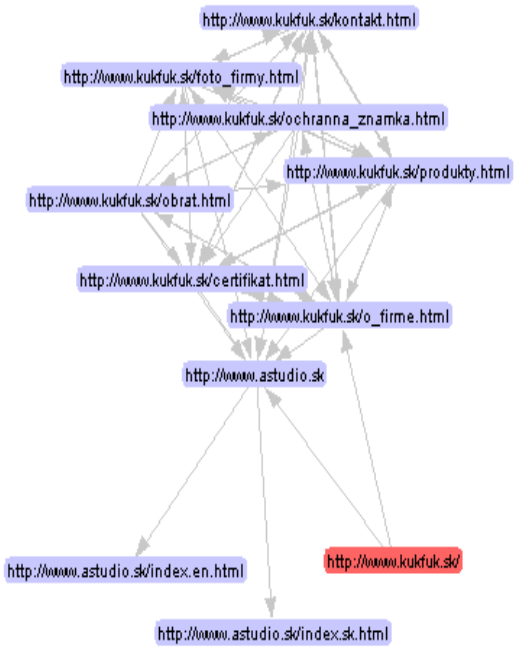

Figure 9. The graph of complete representation in the Prefuse with the second level of nesting

5.3 Local representation with the third level of nesting

For illustration, a graph in the Prefuse Visualization Toolkit with the third level of nesting was constructed. This graph is quite complicated, as it can be see in Figure 10. It is overly complicated despite the fact that only local representation is used. The question is whether it can help users in the orientation within the web graph.

The higher level of nesting, the higher time requirements and slower responding. The results introduced in [7] are similar.

6 Related work

Very interesting is KartOO system [5], which concerns the similar problem like our approach. KartOO Visual Meta Search Engine visualizes the map of the searching domain where only important pages from this domain are accessible and visible. Our approach focused on visualization of all pages but only from the nearest neighborhood of an actual page. We are intending to extend our work with drawing a history of previous accesses into the partial map of web graph.

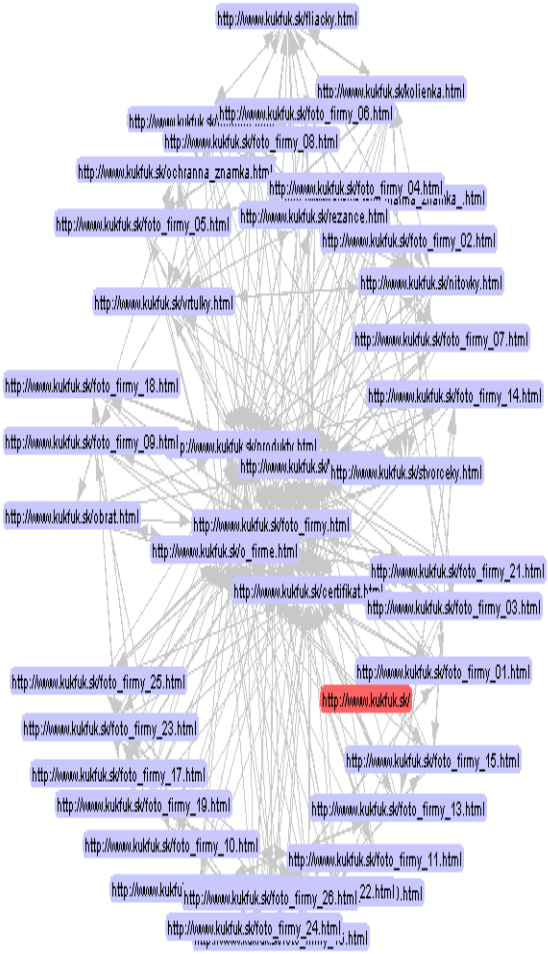

Figure 10. The graph of local representation in the Prefuse with the third level of nesting

[7] introduces the identification of a navigation structure of a web presentation on the base of a so called reference topology. But authors of this work also do not consider history of previous accesses into the partial map of web graph. They do not distinguish local, global and complete presentation, and their tests are not so extensive as test done in our work.

7 Conclusion

Our approach produces the best results for small companie's web presentations – presentations with a small number of web pages. In these cases, the compactness measure is close to "1". The global representation usually provided the worst results. On the other hand, the global representation contains more useful information about connections to other domains and how far we can go by clicks.

Our work can improve orientation in the web graph, because it enables visualization of the nearest surrounding of an actual page – page, which was opened in a browser or was provided by a searching engine. We intend to extend our application in such way that the actual page opened in a browser will obtain a new sub-window, in which the partial map graph will be visualized.

At present, the result of our work is a system for invited matrixes generation, construction of partial web graphs from these matrixes, visualization of these partial graphs, and calculating the compactness of these graphs. The matrix generation is inevitable for the partial web graph construction. Compactness of the graph represents the complexity of the graph. Graphs with high value of compactness are too complicated for users' orientation and some other type of visualization may be selected, for example global presentation.

Our work can be extended in many different ways, for example finding the seeds of the navigation structure, comparing various web maps presented in various other visualization tools or denotation the movement history in partial web graphs.

8 Acknowledgments

The work presented in this paper was supported by the Slovak Grant Agency of Ministry of Education and Academy of Science of the Slovak Republic within the 1/4074/07 project "Methods for annotation, search, creation, and accessing knowledge employing metadata for semantic description of knowledge".

References

[1] Attardi G. at al: **Automatic Web Page Categorization by Link and Context Analysis**, In: THAI-99, European Symposium on Telematics, Hypermedia and Artificial Intelligence, ed.: Chris Hutchison and Gaetano Lanzarone, Varese, 1999, s. 105-119.

[2] Mathieu F, Viennot L: **Local Structure in the Web**, In: Poster Session of the International World-Wide Web Conference, Budapest, 2003.

[3] **KartOO visual meta search engine**, available at http://www.kartoo.com/flash04.php3, Accessed: 17th Jul 2008.

[4] Low G: **Graphviz – Graph Visualization Software**, available at http://www.graphviz.org/Download.php , Accessed: 12th Jun 2008.

[5] Spencer S, Heer J: **Prefuse Visualization Toolkit**, available at http://prefuse.sourceforge.net, Accessed: 12th Jun 2008.

[6] Vejmola S: **The graph theory.** Praha, VŠE v Prahe, 1985.

[7] Volavka F, Svátek V: **The identification of the navigation structure of the web presentation on the base of reference topology**. Department of informatic and knowledge engineering, VŠE, Praha.

[8] **Visualization process starting**, available at http://neuron.tuke.sk/~galova/applet/novy.html, Accessed: 12th Jun 2008.