Visualization, navigation and relationship discovery in graphs

Ján Mojžiš*
Institute of Informatics
Slovak Academy of Sciences
Dúbravská cesta 9, 845 07 Bratislava, Slovakia
upsyjamo@savba.sk

Abstract

Linked data is a concept used for interlinking several data sources, often placed across the world. One of its key requirements is a link. Another is machine readable structure. But nowadays, still many of data sources on the web offer plain unstructured data. Newspaper articles, social networks or business register. Often the data is HTML formatted, where the formatting is mixed with the content, which is improper for machine reading. And the data would be very useful. We could extract information about persons, events, places and other objects. In order to extract such information from unstructured data sources, an advanced techniques of information extraction are used. Even if data structure is extracted or created, a presentation of information to the end user is crucial, because an information overload or clutter can be introduced.

In scope of our work, we focus on graph data structures, data extraction, distributed computing and graph visualization. We design, implement and evaluate a single machine system for data extraction and information retrieval, capable of using advanced graph visualization and filtering techniques. We propose a new visualization concept of pen patterns and colors. Next we define a new universal graph visualization and filtering method, usable for filtering and relationship discovery. We propose a new distributed algorithm PCMARS, intended to be used in a Pregel computing cluster for the graph relationship discovery tasks. We implement our proposal in a client, stand alone program AGECRT (Advanced Graph and Clutter Removal Tool) and distributed algorithm PCMARS. A solution is dedicated as one single architecture.

Categories and Subject Descriptors

E.1 [**Data Structures**]: Distributed data structures, Graphs and networks; H.1.2 [**Models and principles**]: User/Machine Systems—*Human factors, Human information processing*; H.3.3 [**Information storage and retrieval**]: Information search and retrieval—*Information filtering*; H.3.4 [**Information storage and retrieval**]: Systems and software—*Distributed systems*

Keywords

graph, visualization, distributed computing, parallel computing, relationship discovery, linked data

1. Introduction

From many kinds of data sources, the Web is most dynamic, dense and universal. Originally created by humans for humans, there are sources we use often on daily basis. Newspaper articles, business statistics, social networks, traffic are just a few examples. But despite the fact, that we have a vast set of data available. As humans, in order to get information, we are capable to extract and use only a small portion of such data. Indeed a machine computing power is very helpful for several reasons. First, machine can process data thousands times faster and more reliable than any human. Next, with the use of appropriate software environments we could perform information extraction and visualization tasks.

Based on website internetlivestats[1], in 2015, internet connected roughly 863 millions websites, in comparison with 2005 it grown more than 13 times. But this metric does not include web pages themselves, only base unique IP addresses. Also, worth of a note are dynamic, periodically changing websites, like news. A big contribution to the Web is the content of social networks, which are results of collaborative work of millions persons across the world. Here, a term Information Society is quite suitable.

Google Search is a popular web search engine owned by Google Inc, which also maintains web data index. Based of official Google web page[2], Google index contains more than 100 millions gigabytes of data (approximately 25 thousands of 4 terabyte discs). Google also maintained a Freebase, multi-domain database of billions of N-triple statements. Now, the project has emerged into Wikidata

*Recommended by thesis supervisor: Doc. RNDr. Michal Laclavík, PhD.
Defended at Institute of Informatics, Slovak Academy of Sciences on TODO: MONTH DD, YYYY.

[1] www.internetlivestats.com/total-number-of-websites/, 16.6.2016
[2] https://www.google.com/insidesearch/howsearchworks/crawling-indexing.html, 16.6.2016

knowledge base. Wikidata contains structured, machine readable data gathered across all the world.

News articles, blogs and social networks, together with traditional web pages (homepages, company pages), written in HTML are types of unstructured data (formatting mixed with content). They are a great part of the Web volume, many times due to their daily or periodically updated content. One of most popular social networks in the present, Facebook (FB), holds more than billion monthly active users (at least once per 30 days cycle, user is logged in). In 2004, the count was 1 million. Ending year 2015, the count grown to 1.5 billion. Users contribute to FB writing status updates, periodical submissions on "timeline" and each user maintains his/her own profile, more or less detail or public. FB is international social network, which links a broad spectrum of people across the world, not depending on language or culture. In the recent past, FB is also the space for firms, political parties or non-governmental organizations to promote themselves. FB evolution is illustrated in Fig.3.

We see Slovak Business Register (SBR) as one kind of social network, which is a public register, where subjects (natural or legal persons and companies) are listed based on particular law. In comparison to FB, SBR links are not based on friends, instead, we find Partners, Management Body, Supervisory board or Liquidators. From official statistics of Ministry of Justice SR[3], the network grown from 52 thousands subjects in 1995 to more than 254 thousands registered subjects in 2015. And, despite the fact, that SBR performs liquidation and deletion of registered entries, on the webpage of SBR, there is still possible to find and display all deleted subjects. Fig.1 displays an evolution of registered subjects in SBR database based on years.

The importance and significance of the Web, as a large evolving information space is marked by various challenges, like Semantic Web Challenge[4] or research papers at WWW [5] or ISWC[6] events. It is a kind of motivation for us to try and find relations in the data. To use a modern distributed computing models, like Pregel, where the data is represented in a graph structure, giving a new opportunities for graph algorithms.

A presentation, or a kind of an "overview" for the user is given by techniques from Information visualization field. A graph visualization, for instance, visualizes relationships providing graphic representation in vertexes and edges. Despite the rich community of researchers and many contributing solutions for many issues in graph visualization, the potential for visualization is not fully used. The research is intended to layout algorithms, clustering or drawing. Many solutions try to address the edge crossing problem on traditional basis (layouts, clustering, bundling, focus+context techniques), but we did not find the works for edge pens and colors. Only rather controversial edge visualization, where edges are rather hidden than displayed [4, 5], ultimately solving edge crossing, but

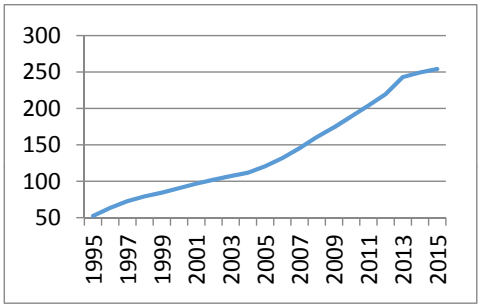

Figure 1: Slovak Business Register. The evolution of registered subjects count. Vertical axis in thousands.

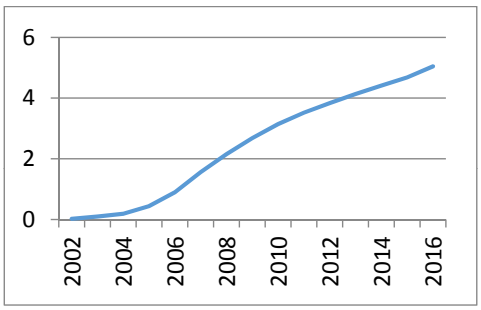

Figure 2: en.wikipedia.org. Article published count on year basis. Vertical axis in millions.

making higher uncertainty in relations.

In scope of this paper we propose a new complex solution for information extraction and processing, involving relationship discovery, resulting in visualization. We present a new and universal method, usable for relationship discovery as well as filtering in graph visualization. We identify and mark relation between graph clutter, cognitive science and psychology, from which, based on our research, we design a new pen drawing and color styles for graph visualization. Combining our solution under architecture capable of extraction, distributed processing and visualization. We also evaluate our proposal on selected data inputs.

1.1 Goals

The aim of this work is to combine relationship discovery, visualization and navigation in graph data. Design filtering techniques, offering some degree of personalization. For unstructured data of SBR propose a structure and its creation. Valid structure can serve in data integration from various sources. Progress toward the concept of Linked Data by T. Berners-lee[2] for the transparent and effective data usage in public concern. A combination of visualization and filtration techniques in data integration, navigation and relation discovery tasks could lead to interesting results. The work pick the following partial goals:

1. Design visualization techniques for better limpidity of visualizations. For 2D visualizations of graphs, several existing techniques use coloring, but pen styles or patterns are potent, yet unused. Various pen patterns can, however, equally discriminate edges as well, as colors. Patterns with colors can represent

[3]http://www.justice.gov.sk/stat/statr.htm
[4]challenge.semanticweb.org/
[5]libra.msra.cn/Conference/526/www-world-wide-web-conference-series
[6]libra.msra.cn/Conference/360/iswc-international-semantic-web-conference

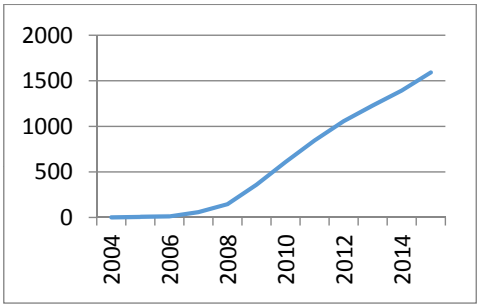

Figure 3: Facebook. Monthly active users. Vertical axis in millions.

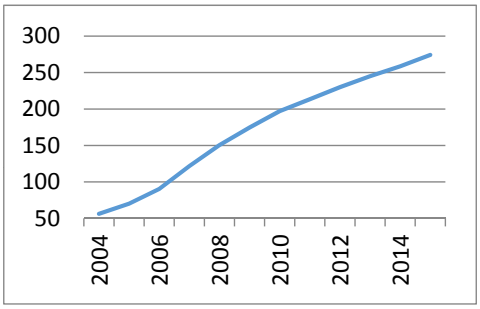

Figure 4: News articles. Article published count on year basis. Vertical axis in thousands.

particular information (the significance of relationship, type of relationship, time context, and etc.). Colors are used, but their choice is rather empirical. We offer color choice based on studies of cognitive science and psychology, especially on human visual system.

2. Design and define a new and universally usable graph filter and visualization method. Graph filtering is not only about hiding redundancies and clearing links, clustering or layouting. Filtering can progress further into semantics of graph data. It can partially address graph visualization problems. We propose a method Many-to many, intended to work with whole sets of vertexes. Filter function Many-to-many would, in those sets, watch edges between vertexes and consider properties of edges and vertexes. Relations in compliance with the conditions would pass the filter, unsatisfying remain hidden. Its usage is therefore not only for graph filtering, but is also valuable in visualization and relationship discovery tasks.

3. Design an algorithm for distributed relationship discovery. One of base criteria in distributed computing is large-scale, because of increasing data volume. When developing an algorithm for relationship discovery, we need to select appropriate data structure and distributed computing model. We also use the concept Many-to-many for the algorithm and will work with the sets of vertexes. There is no automatic reasoning, algorithm contains several conditions, executed during relationship discovery, making it more universal for general use.

4. Personalize graph filter and visualization. For methods of visualization, using colors and pen patterns, enable personalization by defining color sets and patterns, creating different user profiles. For filtering, define own sets of "interesting" vertexes, to be preferentially searched and displayed in graph visualization.

5. Combine navigation, filtration and relationship discovery. Relationship discovery is important to us. With the mind on constantly growing content of the Web, offering rather large volume of data. Thus we calculate with a distributed computing platform. Base combination is connection between distributed computing algorithm and client side application with presentation layer. In general, a solution offering filtering, visualization and interactive relationship discovery is preferred. We utilize Many-to-many together with distributed algorithm, explore based on links or semantics. Thus we propose complex large scale solution, for interactive relationship discovery and perceive them with the use of various methods of graph visualization and filtration. As we show, it is possible to implement and use it.

2. Colors and patterns of pen

In introduction we briefly explain our motivation for research. We note SBR or Billion triple challenge. Visualizing such volumes of data as a whole is not desirable, mainly to the limitations of current displays. In order to display a graph and avoid common issues regarding graph clutter, we can use methods, which are (to our best notion) currently left unused. In our research in graph visualization field, we realized, that edge coloring plays no major aspect in proposed solutions. It is despite a fact, that those papers focus on graph visualization [10, 18, 9].

2.1 Related work

Jianu [10] even discuss edge coloring, but his proposal is to color edges based on their relative positions. Thus his coloring changes with edge positions and colors are independent of semantic context. His scheme cannot express additional information possibly stored in vertexes or edges. Rusu [18] use gestalt psychology in the process of edge drawing. His primary goal is to solve (aesthetic) problem with edge crossing and he does this by terminating edge in the position of the crossing. Edge coloring is not primary. Herman [9] offers an extensive list of visualization methods. It is a bit surprising, that no place for colors (with an exception of treemaps) and edge styles is given. Perhaps, a worth noting is, rather experimental, proposal of partial edge drawing [5, 4], but we see it as rather a controversial method of edge visualization. Just with a no edge visualization, the edge crossing problem could be definitely solved, but we suppose, that increasing uncertainty is not intended for graph visualizations. As the edge crossing is quite common problem in graph visualizations, those crossing edges would all remain hidden.

2.2 Design

Graph visualization is rather broad field with many methods and finding regarding how to visualize graphs. In this field, there is also a constant research. Visualization is a final layer, easing interaction and navigation. It increase the clearness of relationships. However, we still see unused potential in this field, we define following goals for our visualization, following 1st point of our main goals:

- Define proper pen styles for edges

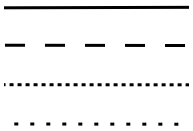

Figure 5: Pen patterns, straight lines.

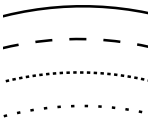

Figure 6: Pen patterns, curved lines.

- recommend proper colors for the pen with white background
- propose a persistent color representation for edges (including re-layouting)
- propose a semantics for relationship during the process of edge color and pen style selection.

We define appropriate patterns and colors for pen. Pen is an object defining visual properties of edges, while edges are painted in resulting visualization. Our design is based on information about human visual system. Based on Visual Expert[7], a difference of colors and brightness is important. A two colors of *relative high* color and brightness difference are suitable for a combination in visualization. [8] states how to calculate color difference. To calculate brightness difference, we need to calculate brightness itself. The basis are three components of RGB model. Each component has a weight assigned by the standard ITU-R in recommendation BT.601-4. Poynton [17] here notes, that weights are no longer considered accurate, as were for older CRT displays. We thus propose modified weights: $Y = 0,30078 \times R + 0,58986001 \times G + 0,10935999 \times B$. Y is a resulting brightness, normalized on the scale 0-255, R,G and B are color components of RGB model. We discovered new weights during our experiments of color to grayscale image conversions. We then implemented algorithms and proposed weights in our program BKonvert[7], which served as educative and experimental tool during our study at Matej Bel University in the Applied Informatics field.

Brightness difference is a brightness difference $Y_\Delta = Y_A - Y_B$. Color difference K_Δ is an expression of how distant given two colors are in RGB model. We rely on W3 [6], where there are formulas for brightness and color difference calculations. Color difference is given by $K_\Delta = |R_1 - R_2| + |G_1 - G_2| + |B_1 - B_2|$. A suitable high brightness difference Y_Δ is, based on W3, a constant with a value of 125. A color difference K_Δ is (according to W3) a constant with a value of 500. When the result of brightness difference is $Y_\Delta > 125$ and color difference $K_\Delta > 500$, then given two colors are in a suitable combination (according to W3). We thus propose aforementioned formulas (Y_Δ and K_Δ) for a suitable color selection, based on referencing color (background).

We propose following pen pattern styles: three terminated and one full, all in two variants; straight and curved (Fig.5, Fig.6)

Together there are eight patterns (4 straight and 4 curved), which we consider proper for drawing. Patterns are placed in a set Z, while $|Z| = 8$. More patterns could cause discrimination issues, when two similar patterns are colored by the same color. We do not recommend other patterns, like "+", because it reminds crossing, again, when two edges get near and are colored by the same. We do not recommend using bent lines, as the sudden change in edge direction angle could obfuscate edge following. These patterns (along with presented color formulas) are used for visualization in our program AGECRT, see Fig.10.

```
int pure = {true,true,true,false,false);
Color[] pureColors = new Color[]{
  new Color(0,0,128), // blue
  new Color(0,0,255),
  new Color(0,128,0),  // green
  new Color(0,192,0),
  new Color(0,0,0), // black
  new Color(155,0,0), // red
  new Color(255,0,0)
};
Color[] edgeColors = new Color[]{
  new Color(92,92,255), // blue 1
  new Color(255, 150,0), // orange 1
  new Color(160,160,255), // blue 2
  new Color(255,0,255), // comb. pink 1
};
Color colorHash(int h1, int h2){
  int pure = pureColors[Math.abs(h1 MOD cista.length)];
  if (pure = true){
    return pureColors[Math.abs(h2) MOD
                      pureColors.length];
  }
  return edgeColors[Math.abs((h2 XOR h1) MOD
                   edgeColors.length)];
}
Color color(Edge edge){
  Vertex v1 = edge.v1;
  Vertex v2 = edge.v2;
  int h1 = hash(v1);
  int h2 = hash(v2);
  if (h1 != h2)
    h1 = h1 XOR h2;
  Color color = colorHash(h1, h2);
  return color;
}
```

Figure 7: Color selection from the set, based on h1 and h2 hash values.

In Fig.7 is an algorithm of color selection. Main function is *color*. $h1, h2$ are hash values of vertexes v_1, v_2 in integral numbers. If $h1! = h2, h1 = h1XORh2$. XOR is a binary operation of bitwise xor on integral numbers. MOD is a binary operation modulo [8]. Function *colorHash* firstly evaluates $h1$ to decide whether a color should be *pure*. Pure color, in our design, is a color, having two components of RGB equal to zero and one non-zero (set $pureColors$). If $pure = true$, then color is pure and is is selected from $pureColor$. Otherwise, a color is selected from $edgeColors$. A selected color is used to draw edges between v_1 and v_2. Similarly, we select edge pattern style. Fig.5 and Fig.6 are placed in one set. Because there are two vertexes and one pen pattern set, the selection is almost identical to the color selection process. Use binary

[7] https://sourceforge.net/projects/bkonvert/

[8] in modular arithmetic, a binary operation, returning integral remainder from division.

operation XOR to combine hashes and get a number, next function modulo returns an index of pattern in the set and edge pattern is selected.

3. New and universal filtering method

The main motivation is the increasing volume of data on the Web, like we stated and illustrated in introduction. Due to the scaling Web, we have to design a large-scale solution. It bears parallel computing power and distributed resources. An interesting choice is a Pregel, one of new and potent distributed computing models. Pregel combines fault tolerance, simplicity of development on a distributed cluster and a new concept of synchronization, based on supersteps. Pregel is a brand new computing model, but with a clear potential [13].

During our research, we discovered rules for graph filtering, which could be used universally. They can be used in graph visualization for filtering and also to discover relationships in a distributed computing model.

Our goals in this section follow main goals, 2. and 3.

3.1 Base design

Let G be the graph $G = V, E$, where E is a set of edges and V set of vertexes connected by edges from E. We propose to divide the V set into two disjunctive subsets of *interesting vertexes* I and *ordinary vertexes* O. For algorithm of relationship discovery and graph visualization, set I holds vertexes, which are important or interesting. In visualization, the set contains all currently visible vertexes. In distributed computing algorithm for relationship discovery, the set contains vertexes, between which, a relation is to be discovered. Set O contains vertexes, not visible in visualization process. In relationship discovery, there are all vertexes $V - I$. With these two sets, we define rules:

$$H = \{\exists u \in O | \exists v_1, v_2, \in I : \exists e = \{u, v_1\} \wedge \exists e = \{u, v_2\}\} \quad (1)$$

$$h \in H : M = \{u \neq h | \exists e = \{h, u\} \wedge u \in I\} \quad (2)$$

Where:

H - set of all neighbors of vertexes v_1 a v_2

I - set of all interesting vertexes

O - set of all ordinary vertexes

M - set of all interesting neighbors of vertex h.

Rule 1 tells of existence of vertex u from the O set, which is in sequence $P = (v_1, e_1, u, e_2, v_2)$. Rule 2 specifies, that vertex u is excluded from the interesting set of vertexes I. In graph visualization, we see proposed rules usable for vertex hiding or expanding/collapsing (like Herman states about ghosting and hiding [9]). In the expanding process, only vertexes, which pass proposed rules filter could be identified and visualized. For details we recommend our former works[14, 15]. In algorithm of distributed relationship discovery, these rules directly qualifies the relationship.

3.2 method in distributed algorithm PCMARS

Proposed rules are used in our Pregel-based algorithm PCMARS (Pregel Computing Model And Relationship Search). In Pregel, a base component is a vertex-centric approach (Pregel type). All vertexes perform their own *compute()* function and the computing synchronization is governed by supersteps. Messages are passed in superstep i and received in superstep $i+1$. Even if computations are running parallel between nodes, *compute()* method runs sequential inside each node's thread. Messages intended to vertexes, which are located on the same node are stored in the stack of current node. Here messages wait, until they are sequentially picked and processed in *compute()* in suitable vertex on this node. Pregel usually terminates, if there are no additional messages in stack and no vertexes are active (on a given node). Important function is *compute()*, as it is the function, which is user-defined and also message passing, where user can select which values are to be passed. In PCMARS algorithm, all important control instructions are performed with the use of specialized messages and matching responses. There are two base types of messages *PING* and *INTERESTING*. Each one of two messages is intended for the matching st of vertexes. *PING* for the O set and *INTERESTING* for I set. Furthermore, there are response messages, as reactions to receiving two base types of messages. Response messages are *STORED_INTERESTING_REPLY*, *STORED_ACTIVATED_REPLY* and *STORED_REPLY_PINGER*. The choice depends on the kind of received message and on the vertex kind (set of O or I), as well as on particular properties or attributes of particular vertex, where *compute()* is currently running.

A base algorithm design is in Fig.8. A few notes to proposed algorithm. Main part of compute() method is a loop of message receiving (*hasMsgAt()*). Because there are several kinds of defined messages, each one must be handled differently. Some messages, for instance, are distributed along. If actual vertex (running *compute()*) receives *INTERESTING*, it is activated (if vertex is from O set), increments the path length and distribute the message toward its neighbors. Or the value of path length must remain the same, when the reply requires the original path length. If any vertex of O set is activated from two different I vertexes, such vertex is considered as a relationship. Each vertex has an access to its neighbors list along with particular connecting edges (here an algorithm can be modified based on edge types) and also maintains a structure of verexes, which it has received a message from. It is used in several cases during message generation. A vertex can thus, distribute messages received several steps before. When vertex from O is activated by a message from I vertex, such vertex is then capable of responding with *STORED_INTERESTING_REPLY* and activate other vertexes from O set.

Algorithm behaves differently in the first two supersteps. Vertexes of O are sending exploratory messages *PING*. Vertexes of I are sending *INTERESTING* messages. Subsequent supersteps receive these messages and react accordingly. Algorithm in Fig.8 is a sample code picked from PCMARS source code, which is more complicated and complex. Whole sourcecode can be found on supplied CD in the work.

4. Base architecture

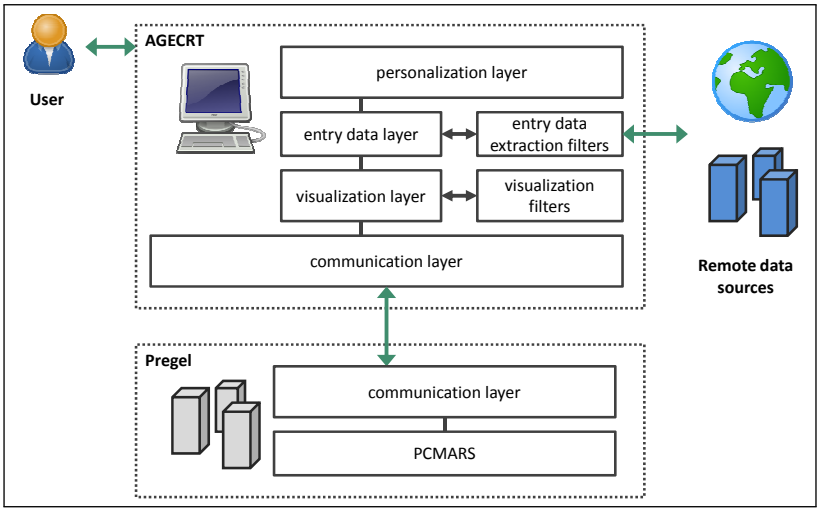

Figure 9: Base architecture.

We follow our base goals 4 and 5, presenting an base architecture, combining distributed algorithm PCMARS and our client side application AGECRT. It is a platform independent solution, capable of large-scale computing with the use of computer cluster. Also it can perform visualization and filtration as a client side application, offering interactive navigation and relationship discovery with its presentation layer. It can gather data from the Web and process them. With the help of its parsers and designed schemata, it is able to extract a structure from unstructured HTML data sources. Data can then be used in subsequent further development of Linked Data. Fig.9 depicts a combination of PCMARS and AGECRT in one system. Communication layer will be covered in implementation, for example as a console. Communication can run also on a client side (for test purposes). Then PCMARS and AGECRT can run on one client machine at the same time. PCMARS is equipped with the functionality needed. Thanks to Sedge framework, it is possible to configure worker nodes, add or remove them as needed, which is stated in its configuration file. Communication between PCMARS and AGECRT will be realized through a console interface or with sockets. Architecture is supporting both cases.

Data entry and extraction filters must be equipped with a suitable set of data processors. Parsers are used to extract data. From unstructured HTML, a structured data form will be gathered. For a structured data sources, it is simple to extract data from XML, RDF/XML, CSS or N-triples formats. Here a template or a schema is required only. Java already supports XML reading and parsing. We calculate with more complex formats, like PDF, for which, we can use one of available third party drivers, like the one with source code, pdf2HTMLEx[9]. Even it support only HTML format in output, we have discovered, that it is rather a simple form of HTML and thus suitable for parsing. We have also evaluated Tabula[11], but, today, after evaluation, we can state, that pdf2HTMLEx is more suitable and reliable. pdf2HTMLEx requires no extra interaction with the user, is is able to process many kinds of PDF files, including scientific papers and articles, pdf2HTMLEx is thus a seamless choice. With minor changes, it is possible to implement SPARQL endpoints support. SPARQL endpoints are maintained by their respective data publishers. Offering access through several structured data protocols, like JSON, POST, GET or SOAP. These protocols can easily be built into entry data extraction filters.

Visualization layer is covered solely with a Jung[10] framework. We have used this framework to define custom methods of drawing and presentation layer in AGECRT.

Internal entry data layer works with structured data, refined with one of entry data extraction filters. This layer is directly connected with visualization layer and with personalization layer.

Personalization layer is a governing layer to visualization layer and entry data layer. It holds certain definitions for personalization. Enables to define and store parameters for vertex selection and filtering, visualization settings, layout algorithms and additional information about vertexes and edges. User can create his own profile and reuse it in subsequent visualizations. For example a vertex filter. Only vertexes listed in the filter would visualize, others are hidden.

5. Evaluation

In this section we present sample evaluations, selected from our thesis.

5.1 Clutter removal

Visualization filter for graph clutter removal is working according to its design and rules (page 5). We can evaluate visualization filter based on parameters: actual reduction of visual components (vertexes) based on rules and reduction of graph density (edges and vertexes count). We use base graph $G = V, E$, which also proposed on page 5. Further we refer to vertex count as $|V|$ and edges count $|E|$, we also define an average vertex degree $\nu = \frac{\Sigma_{v \in V} deg(v)}{|V|}$. Fig.11 is a visualization of path between vertexes "Christopher Manning" and "L.Hluchy". Path was obtained using a standard Dijkstra shortest path

[9] http://coolwanglu.github.io/pdf2htmlEX/

[10] http://jrtom.github.io/jung/

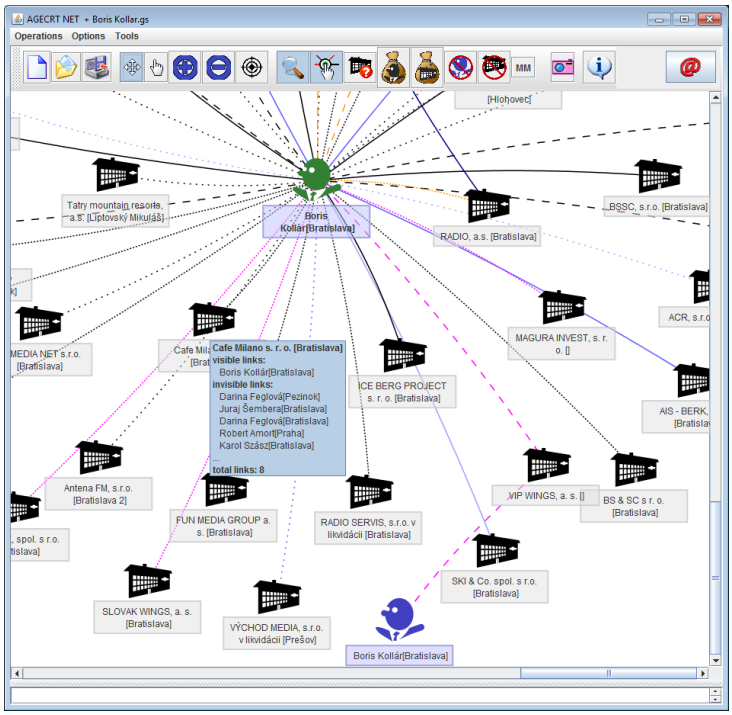

Figure 10: AGECRT main window with visualization of Boris Kollar network.

algorithm. Data source is ACM citation graph[11], where $|V| = 622,335$, $|E| = 1,334,753$, f, $\nu = 4.29$. In Fig.12 we see neighbors of vertex "Authoritative sources" without filter applied. It connects 200 neighbors. Name tags in rectangles were hidden and replaced with circles and polygons, for simplicity. Visualization A (Fig.11) depicts a graph, where $|V_A| = 9$, $|E_A| = 8$, $\nu_A = 1,78$. For comparison, B (Fig.12) is a graph $|V_B| = 207$, $|E_B| = 567$, $\nu_B = 5,47$. Number of visualized vertexes is $\Delta|V| = |V_B| - |V_A| = 198$, new count of edges is $\Delta|E| = |E_B| - |E_A| = 559$, ν is raised from $1,78$ to $5,47$. For such large number of vertexes and edges, with edge crossing, a visualization is considerably cluttered and relations are unclear. Effect of the filter is displayed in Fig.13. In comparison to A, there are new vertexes $|V_C| - |V_A| = 4$, edges $|E_C| - |E_A| = 9$, ν raised from $1,78$ to $2,62$. Based on rules (page 5) are filtered all vertexes, which do not satisfy the rules. Hidden remaining $|V_B| - |V_C| = 194$ vertexes and $|E_B| - |E_C| = 550$ edges. Average vertex degree is higher than in A (Fig.11), which is reasonable, due to newly visualized vertexes, but considerably lesser than in B (Fig.12), where no filter is applied.

In Fig.14 we can see a network of neighbors of vertex "Ground Term Confluence". The graph is a resulting visualization D, after we applied our filter two times on this vertex. Each time a filter was applied, new vertexes were found and visualized. Here is a drawback of our proposal of graph visualization filter, based on rules from page 5. Effectiveness of such filter is limited. A visualization space is eventually filled, which leads to information overload or higher uncertainty of relations. If we perceive graph clutter also as edge crossing problem and vertexes overlapping edges, where we cannot sufficiently state, where one edge ends and other is starting.

A solution actually exists. One just need to use another visualization technique, especially, edge coloring. We use our design from page 3. Colored edges with styles from visualization E are found in Fig.15. Based on new colors and edge styles, we can state, that new information was introduced. Between vertexes "Semantics and" (left of "Ground term confluence") and "PCLOS: a critic", is an edge, colored in pink color and drawn with a full pen. Edge leads under "An extensible" and "An object orien" vertexes, but due to color and edge curving, we can suppose, that edge starts in "Semantics and", ending in "PCLOS: a critic" vertex, free of any breaks. Similarly, vertexes "Sun's Link Serv" and "An object-orient" (above "Ground term confluence") and other cases. We recommend the use of transparency to offer additional possibility, following edges through overlapping vertexes.

5.2 Relationship discovery

Based on the design of PCMARS from page 5, we define two sets; ordinary vertexes O and interesting vertexes I. Tab.1 lists vertexes from I set, among which, according to rules (page 5) and algorithm design, relations are searched. We have evaluated data gathered from Freebase[12] dataset. Evaluation started on 21.8.2014 at 15:05:55 and finished 31.8 (10 days). There were 207,382 executed supersteps. A progress of message generation is displayed in Fig.16. Algorithm returned 314,668 vertexes u, which are contained in sequence $P = (v_1, e_1, u, e_2, v_2)$ from page 5. Those vertexes received *INTERESTING* or *INTERESTING _REPLY* messages at least 2× (vertexes $v \in I$). In order to visualize such a graph, we have used our AGECRT tool. Several vertex types were excluded from visualization (Gender, Male, Female, Place of Birth, Place of Death), because we were not interested in such common relations. After exclusion, graph con-

[11] http://datahub.io/dataset/rkb-explorer-acm, retreived 12.6.2016

[12] https://developers.google.com/freebase/, Retrieved 22.6.2016

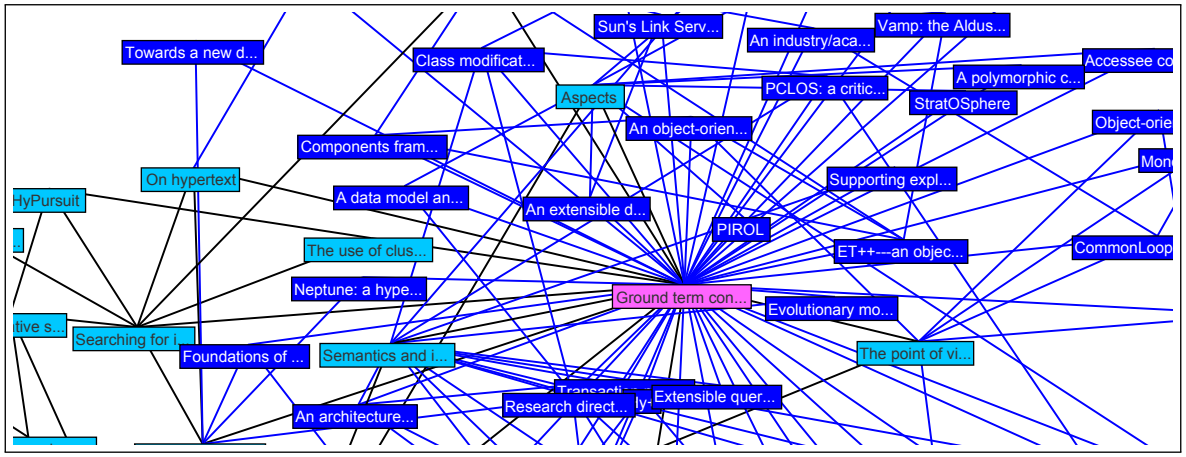

Figure 14: Neighbors of vertex "Ground Term Confluence". Blue colored are newly visualized vertexes and edges. $|V_F| = 67$, $|E_F| = 158$, $\nu_F = 4,72$.

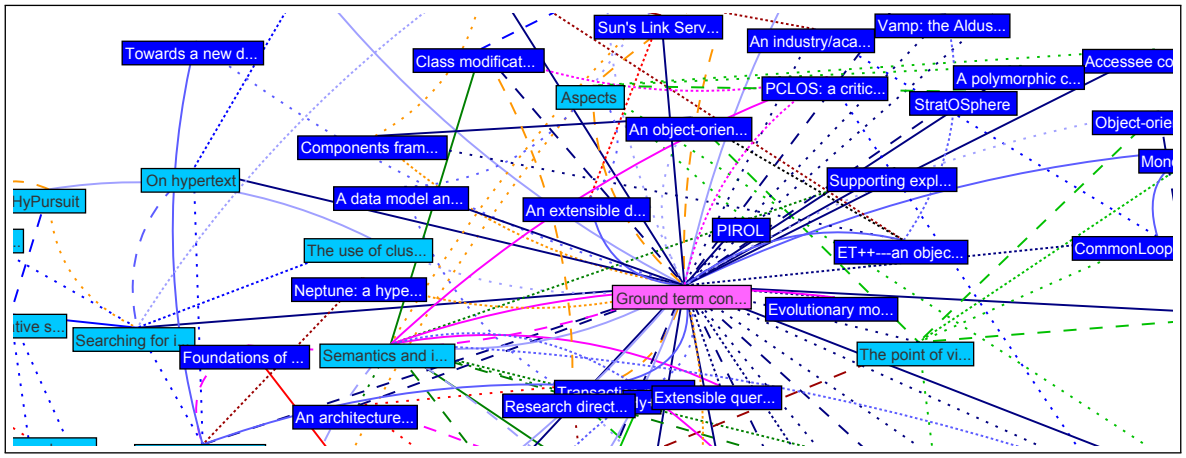

Figure 15: In spatial layout, identical visualization with Fig.14, difference is edge coloring and pen styling from page 3.

tained 234,786 vertexes and 262,279 edges. A graph was still too big to visualize, so we have used Dijkstra shortest path. In this case, shortest path was searched multiple times, because we have searched for shortest path between all vertexes from I set regarding to this notation $dijk(u, v), u \neq v, u, v \in I$. Paths were continuously added into visualization, until all vertexes from I set are searched using $dijk$ function. Obr.17 Freebase dataset is a specific one. For instance, name (Martin Lawrence) has different identifiers MID. This property of Freebase would require deeper study in its internal object representation. Another property of Freebase is, that "film" vertexes are not directly connected to their "actor" vertexes, instead, a mid-vertex v_2 is connected ($film \rightarrow v\_2 \rightarrow actor$). Vertex v_2 contains information on actor role in particular film (character name, film title). We had to increase maximum path length to 2 (max_path_length in PCMARS algorithm from page 10. Vertexes v_2 are always unique and are dedicated to particular combination of film and actor.

From the output of PCMARS and AGECRT visualization, we can find several interesting results. Expected relation between "Jackie Chan" and "Owen Wilson" is not found. Although vertexes "Will Smith" and "Karate Kid" are connected, a connection with "Jackie Chan" is missing. On the other side, we find a connection "Jet Li", "Chinese Martial Arts" and "Karate Kid". The reason behind this is, that "Will Smith" (from Z set) activated "Karate Kid" (O set) indirectly, through several vertexes (e.g. "PG (USA)", MID = m.0kprc8). Thus, "Karate Kid" was activated from vertex "Will Smith". Vertex "Jet Li" activated "Chinese martial arts". Both vertexes "Karate Kid" and "Jet Li" are directly connected to "Chinese martial arts". However, vertex "Jackie Chan" is connected through "People Choice Award" (MID.0dlskb3). This vertex is unique to whole dataset (contained only in one path). Here we could change maximum path length (currently set to 2) or use semantic information of neighboring vertexes and edges.

Additional look in Fig.17 can raise another question. "Anthony Anderson", "Tom Arnold" and "Marsha Thomason" are all connected with "Actor" vertex. Why not as well "Will Smith" or "Jackie Chan"? Are they not actors? The reason is, that "Marsha Thomason" is directly connected with "Actor", like is vertex "Will Smith". But "Marsha Thomason" did not activated vertex "Actor" directly. "Marsha Thomason" activated vertex "Scott Taylor" (MID.025zlv2), which, in turn, sent a message fur-

Table 1: Set I of interesting vertexes of Freebase dataset. This set is used in distributed computing algorithm PCMARS and also in AGECRT to discover relations and shortest path.

Freebase MID	Name	Freebas MID	Name
m.02xbw2	Gabrielle Union	m.029_l	Delroy Lindo
m.0147dk	Will Smith	m.012d40	Jackie Chan
m.0271y9f	Jaden Smith	m.05v_r84	Jackie Chan
m.01qg7c	Barry Sonnenfeld	m.01q_ph	Owen Wilson
m.01vvzb1	DMX	m.01xndd	J.J. Abrams
m.0hqly	Steven Seagal	m.042xrr	Anthony Anderson
m.0gy64rt	Samuel Steven Seagal	m.0bvb9mz	Anthony Anderson
m.02633g	Martin Lawrence	m.0451j	Jet Li
m.01th95y	Martin Lawrence	m.05qg6g	Zoe Saldana
m.01hhx1l	Willennium	m.05jpsx	Chi McBride
m.0b8xmc	Robinne Lee	m.029pnn	Tom Arnold
m.07y925	Marsha Thomason	m.05d79k	Bill Duke

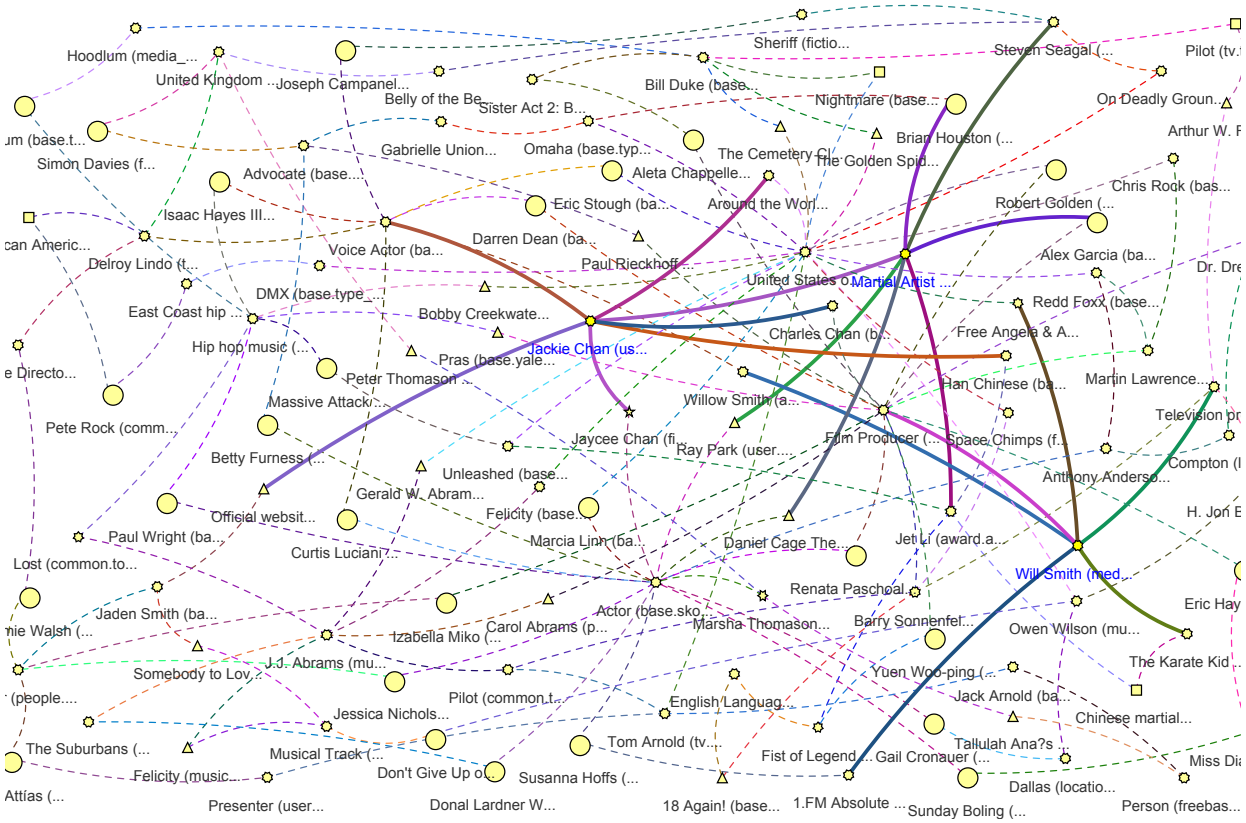

Figure 17: Resulting visualization of Freebase dataset, after processed with PCMARS. Visualization is taken from our program AGECRT. Bold styled are edges for vertexes of "Jackie Chan", "Martial Artist" and "Will Smith".

ther to "Actor" vertex. This path "Marsha Thomason"→ "Scott Taylor" → "Actor" was stored. Path length is 2 (number of edges).

6. Conclusions

In our thesis, we propose a new coloring method for graph edge crossing problem. There are many papers available, intended to graph visualization. But we have not found any, which would discuss pen patterns and edge color styles, based on cognitive and psychovisual aspects. Perhaps it is a bit shameful, as with edge styles, we can represent more different edge combinations. From Fig.5 and Fig.6 we can have eight different edge styles. Together with colors in Fig.7, we can have $8 \times 11 = 88$ different edge combinations. Even if papers discuss graph clutter, only a brief, conventional aspect of aesthetics criteria is addressed, without higher point on cognitive science or human visual system perceiving. Through our design

```
public boolean compute() {
FilterMessageNorm msg = null;
Vertex src;
int pathLength;  // message receiving
  while (hasMsgAt(getStep () - 1)) { // pick
  msg = pollMsg (); // messages
  src = msg.getSourceVertex();
  pathLength = src.getPathLength();  // based on message type
  if (msg == STORED_INTERESTING || STORED_PING ||
             STORED_DISTRIB_INTERESTING || STORED_DISTRIB_PING)
    passMessageToAllNeighbors(msg);
  else if (msg == STORED_INTERESTING_REPLY ||
                  STORED_ACTIVATED_REPLY)
      replyWithInteresting(msg, src)
  pathlength ++;
  if (msg == STORED_REPLY_PINGER)
     if (this.isActivated == 1)
       if (pathlength <= max_path_length)
         ReplyToPingerWithInteresting(STORED_REPLY_PINGER,
                                      src);
 if (msg == PING){ // is -this- vertex interesting ?
   if (getInteresting ()) { // reply with INTERESTING
     ReplyWithInteresting(msg , pathLength - 1);
   } else if (this.isActivated == 1) { // reply
                                       // INTERESTING
     if (pathLength <= max_path_length) { // reply with
       // messages, received from stored paths
       ReplyToPingerWithInteresting(PING, src);
       ReplyWithVisibleAll(PING , pathLength); // need to
       // send reply to source. In case, that outgoing
       // edge is not defined, distribute PING
       PassToAll(PING, src);
     }
   }
 }
 }
 if (getNeighbors().length > 0) { // are there any
  // neighbors?
  if (getStep () == 1) { // INTERESTING sent in
  // 1. step
  if (this.isInteresting() || this.isActivated == 1)
     // i am from Z, sending INTERESTING
     PassToAll(INTERESTING, this.getId());
  } else
     return true; // i am from O, i send PING
     // in 2. step
 }
 if (getStep () == 2) { // 2. step
   if (! getInteresting ()) { // i am from O
     PassToAll(PING, this.getId());
   }
 }
}
 return false;
```

Figure 8: PCMARS with function compute().

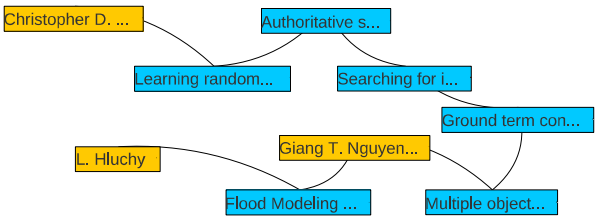

Figure 11: Sample visualization A of the shortest path between "Christopher D. Manning" and "L.Hluchy" using standard Dijkstra shortest path algorithm. There are two types of vertexes: authors (orange color) and articles (blue color). Statistics $|V_A| = 9$, $|E_A| = 8$, $\nu_A = 1,78$.

of color and pen styles, formulation of rules, distributed computing algorithm development and implementation in PCMARS and AGECRT we fulfill our base goals, stated

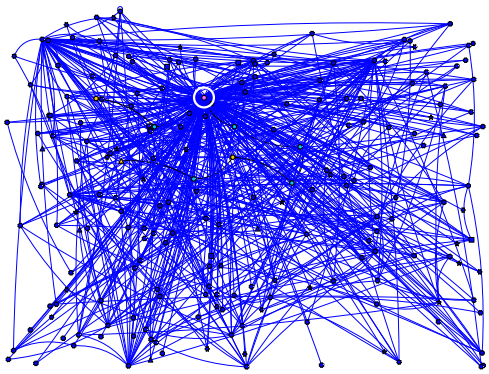

Figure 12: All 200 neighbors of vertex "Authoritative sources" (vertex in white circle). Blue colored are newly visualized vertexes and edges. Statistics $|V_B| = 207$, $|E_B| = 567$, $\nu_B = 5,47$.

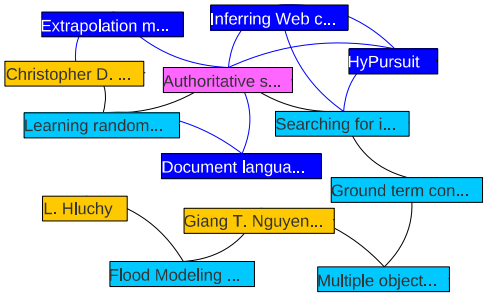

Figure 13: Visualization C is a visualization of a graph, where a filter is applied to vertex "Authoritative sources" (pink color). Blue color is for newly visualized vertexes and edges. Statistics $|V_C| = 13$, $|E_C| = 17$, $\nu_C = 2,62$.

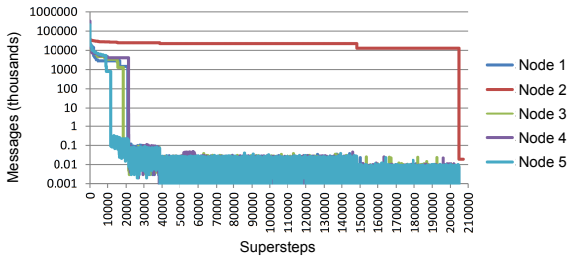

Figure 16: An evolution of message sending based on supersteps in algorithm PCMARS. Vertical axis represent message count in logarithmic scale, horizontal denote supersteps.

in our thesis.

Relations discovered with PCMARS could, theoretically be found with the use of SPARQL querying. However, a few notes must be stated. In order to explore paths in SPARQL, a SPARQL (ver.) 1.1 is needed (property paths support). One have to use either star symbol (*) to denote a variable path length, or write down a full, fixed path length, including all predicates, contained in path. It is also possible to use negation with symbol (!),

but in order to navigate properly with negation, one must write a predicate NOT contained in path, otherwise such predicate would terminate path exploration. With this information, we can use SPARQL 1.1 property paths to perform relationship discovery, similar to PCMARS. But we should carefully select SPARQL implementations, as not each one does actually support 1.1 version, or their support is partial. We recommend consulting with W3C [13] webpage, where a list of implementations, along with their support, is maintained. In several specific cases, even a performance issues could be introduced by implementations, as stated in [1, 12].

PCMARS, in its current design, can omit edge directions. This can be advantage as well as disadvantage, depending on particular situations. Edge direction omitting can be easily altered with additional condition on message receiving, to evaluate, whether an outgoing edge actually exists.

Entry data extraction filters, specified in architecture (Fig.9) can be further enhanced with new filters for new data sources. In our thesis we propose one data extraction filter for Slovak Business Register (SBR) HTML output along with a sample structured schema. Currently, there is still rather large quantity of unstructured data sources on the Web, despite proposed practices of Linked Data[3], recommendations of Tim Berners-lee [2] and W3C consortium working groups efforts[14] (particularly CSS, RDF or SPARQL groups). Although Slovak Republic is a member state of EU, it has open government data portal data.gov.sk, participates in open government projects and e-government, still, many bills, invoices or acts are published in plaint, unstructured PDF format. For details about semantic data availability in Slovakia, we recommend consulting our recent work [16].

Acknowledgements. This work was partially supported by the Slovak Research and Development Agency, project CLAN with ID APVV-0809-11 and by the Scientific Grant Agency of the Ministry of Education, science, research and sport of the Slovak Republic and the Slovak Academy of Sciences, project VEGA, ID 2/0185/13.

References

[1] M. Arenas, S. Conca, and J. Pérez. Counting beyond a yottabyte, or how sparql 1.1 property paths will prevent adoption of the standard. In *Proceedings of the 21st international conference on World Wide Web*, pages 629–638. ACM, 2012.

[2] T. Berners-lee. Linked data - design issues. `http://www.w3.org/DesignIssues/LinkedData.html/`. Retrieved 18.6.206.

[3] C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227, 2009.

[4] T. Bruckdorfer, S. Cornelsen, C. Gutwenger, M. Kaufmann, F. Montecchiani, M. Nöllenburg, and A. Wolff. Progress on partial edge drawings. In *Graph Drawing*, pages 67–78. Springer, 2012.

[5] M. Burch, C. Vehlow, N. Konevtsova, and D. Weiskopf. Evaluating partially drawn links for directed graph edges. In *Graph Drawing*, pages 226–237. Springer, 2011.

[6] W. W. W. Consortium et al. Techniques for accessibility evaluation and repair tools. *http://www.w3.org/TR/2000/WD-AERT-20000426*, 2000.

[7] V. Expert. Sbfaq part 6: Color for text and graph legibility. `http://www.visualexpert.com/FAQ/Part6/cfaqPart6.html/`. Retrieved 18.6.2016.

[8] Had2Know. How to Calculate Color Contrast from RGB Values. `http://www.had2know.com/technology/color-contrast-calculator-web-design.html`. Retrieved 15.6.2016.

[9] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24–43, 2000.

[10] R. Jianu, A. Rusu, A. J. Fabian, and D. H. Laidlaw. A coloring solution to the edge crossing problem. In *Information Visualisation, 2009 13th International Conference*, pages 691–696. IEEE, 2009.

[11] M. Laclav et al. Accuracy of person identification based on public available data. In *2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 253–256. IEEE, 2016.

[12] K. Losemann and W. Martens. The complexity of evaluating path expressions in sparql. In *Proceedings of the 31st symposium on Principles of Database Systems*, pages 101–112. ACM, 2012.

[13] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 135–146. ACM, 2010.

[14] J. Mojžiš and M. Laclavík. Graph clutter filtering based on connectivity distance and visibility. In *Science and Information Conference (SAI), 2014*, pages 153–158. IEEE, 2014.

[15] J. Mojžiš and M. Laclavík. Relationship discovery and navigation in big graphs. In *Intelligent Systems in Science and Information 2014*, pages 109–123. Springer, 2015.

[16] J. Mojžiš and M. Laclavík. Browsing semantic data in slovakia. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 6(3-4):47–59, 2016.

[17] C. Poynton. Frequently asked questions about color. `http://cyrille.nathalie.free.fr/computer%20vision/color_gamma_white_balance/ColorFAQ.pdf`. Retrieved 20.6.2016.

[18] A. Rusu, A. J. Fabian, and R. Jianu. Using the gestalt principle of closure to alleviate the edge crossing problem in graph drawings. In *Information Visualisation (IV), 2011 15th International Conference on*, pages 488–493. IEEE, 2011.

Appendix

Selected Papers by the Author

J. Mojžiš and M. Laclavík. SRelation: Fast RDF graph traversal. In Knowledge engineering and the semantic web : 4th International Conference, KESW 2013. Eds. Klinov, P., Mouromtsev, D. - Berlin : Springer, 2013, cCIS 394, p. 69-82. ISBN 978-3-642-41359-9. ISSN 1865-0929.

J. Mojžiš and M. Laclavík. Graph clutter filtering based on connectivity distance and visibility. In Proceedings of Science and Information Conference 2014. - London : The Science and Information (SAI) Organization, 2014, p. 153-158. ISBN 978-0-9893193-1-7.

J. Mojžiš and M. Laclavík. Relationship Discovery and Navigation in Big Graphs. In: Intelligent Systems in Science and Information 2014. Springer International Publishing, 2015. p. 109-123. ISBN 978-3-319-14653-9.

J. Mojžiš and M. Laclavík. Browsing Semantic Data in Slovakia. BRAIN. Broad Research in Artificial Intelligence and Neuroscience, 2016, 6.3-4: 47-59.

[13] https://www.w3.org/wiki/SparqlImplementations

[14] https://www.w3.org/Consortium/activities. Retrieved 20.6.2016