Multiple Diagram Navigation (MDN)

Hisham Benotman
Portland State University
Portland, Oregon
benotman@pdx.edu

Lois M.L. Delcambre
Portland State University
Portland, Oregon
lmd@pdx.edu

David Maier
Portland State University
Portland, Oregon
maier@pdx.edu

ABSTRACT

Domain novices learning about a new subject can struggle to find their way in large collections. Typical searching and browsing tools are better utilized if users know what to search for or browse to. Navigation systems with richer user interfaces could go beyond search and browse facilities by providing overviews and exploration features. We present Multiple Diagram Navigation (MDN) to assist domain novices by providing multiple overviews of the content matter. Rather than relying on specific types of visualizations, MDN superimposes any type of diagram or map over a collection of information resources, allowing content providers to reveal interesting perspectives of their content. Domain novices can navigate through the content in an exploratory way using three types of browsing (queries): diagram to content, diagram to diagram, and content to diagram. We present positive indications of MDN usability and usefulness we received from a preliminary user study. We also present our vision for using heuristics about diagram structures to help rank results returned by MDN queries.

Keywords

Exploratory Search; Superimposed Information; Diagrams; Wikipedia; Visual Query

1. INTRODUCTION

Navigating collections of information resources such as website pages or digital library documents has been widely supported with traditional information retrieval techniques such as keyword searching and hyperlink and menu browsing. However, with a vague information need that may span multiple concepts (pages), a domain novice may struggle to find the right search keywords or click the right hyperlink(s). For example, a domain novice can struggle to expand their knowledge about information security beyond basic concepts such as "computer viruses" and "phishing attacks". With large collections targeting different audiences, the user may spend a long time iteratively searching and browsing. To tackle this information-seeking problem, known as *exploratory search* [10], the IR and CHI research communities have joined efforts to provide interactive IR solutions.

One major body of research (discussed in Section 5) has focused on providing graphical overviews of documents and collections using dynamically created visualizations such as word clouds, time lines, and concept maps. These visualizations are induced from the underlying documents and document metadata. In this work, we superimpose existing diagrams over a collection of related web pages or documents by connecting diagram elements to the pages or documents. Because the options of induced visualizations are restricted by the available types of visualization tools and the nature of the collection and the extracted data, our approach aims to broaden the choices of graphical overviews by including any type of diagram or map (i.e., any visual structure saved in SVG or image format). This flexibility can be helpful in revealing interesting and multiple perspectives of a collection. For example, the Attack Graph (AG) shown in Figure 1 [2], including different scenarios of attacking a computer system, shows a process or a "how-to" aspect of the domain of information-security threats. The Threat Tree (TT) (excerpts[1] of the TT are shown in Figure 2) [8], shows a different perspective of the domain: a hierarchical classification of computer security threats. Content providers may easily find or create similar diagrams. Using such diagrams to provide interactive navigation to related content can be helpful to domain novices.

We present Multiple Diagram Navigation (MDN), a visual navigation system that allows content providers to superimpose multiple diagrams over a collection of information resources. By coordinating multiple diagrams, MDN provides three types of browsing (or *visual queries*): diagram to content (D2C), diagram to diagram (D2D), and content to diagram (C2D). These features can facilitate domain exploration by allowing users to smoothly switch among seeing an overview by looking at a diagram, seeing a different point of view by changing to a different diagram, and reading content after browsing to a page or a document. Figure 1 shows an MDN prototype that included the AG, TT, and related Wikipedia pages.

The paper is organized as follows. Sections 2 and 3 describe MDN visual queries and MDN components, respectively. Section 4 contains an illustration of MDN visual query processing. Sections 5, 6, 7 contain related work, im-

ExploreDB'16, June 26-July 01 2016, San Francisco, CA, USA
© 2016 ACM. ISBN 978-1-4503-4312-1/16/06…$15.00
DOI: http://dx.doi.org/10.1145/2948674.2948678

[1]Full AG and TT diagrams can be seen at our MDN prototype page http://web.cecs.pdx.edu/~benotman/mdn.html

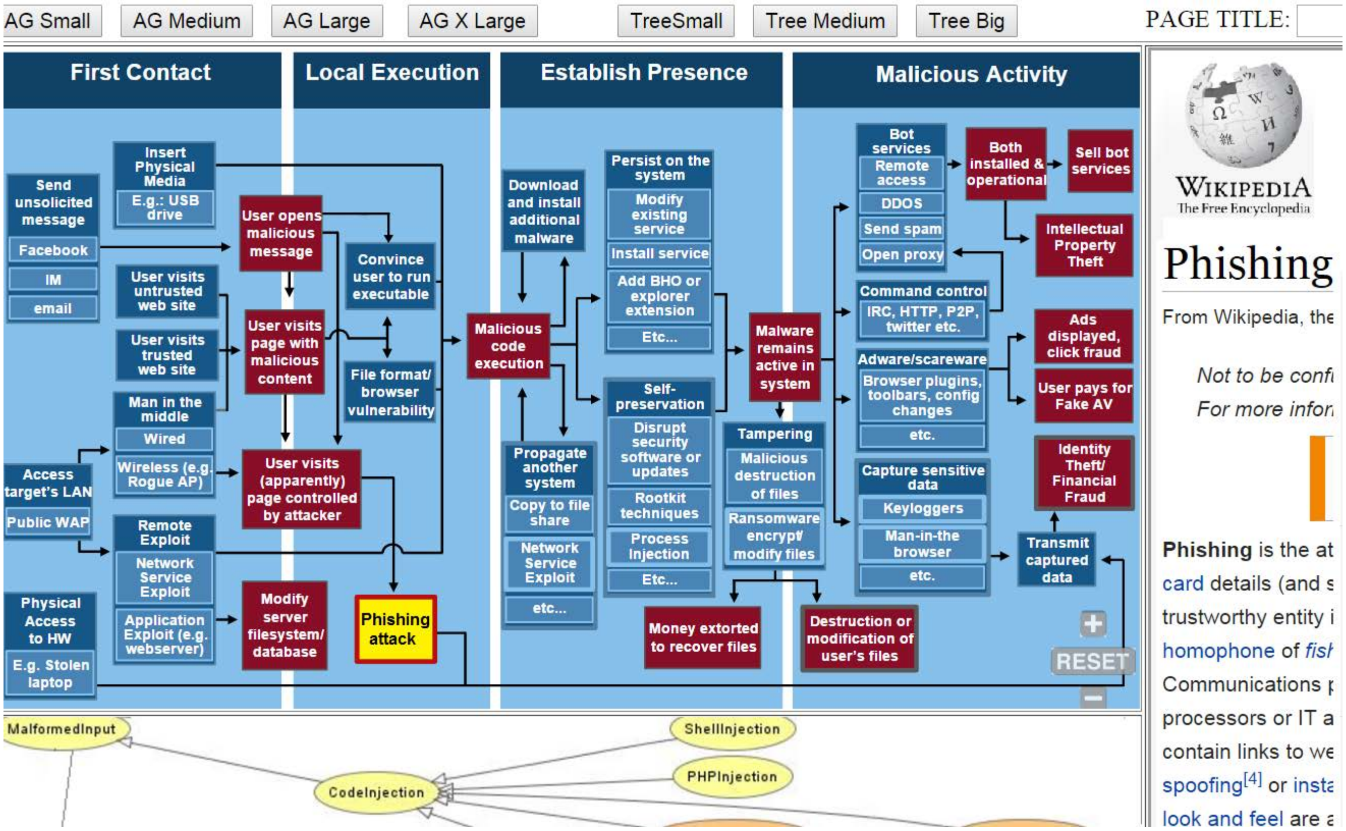

Figure 1: An MDN prototype including the Attack Graph (AG) [2] (the top left diagram with sky-blue background), the threat tree (TT) [8] (only a small part is shown at the bottom left), and a Wikipedia page on the right. The blue boxes in the AG represent steps or example threats while red boxes represent goals. This screenshot of MDN shows a content to diagram (C2D) query where a user visiting the Wikipedia page of Phishing results in MDN highlighting the phishing element (with yellow background) in the Attack Graph.

plementation, and preliminary evaluation, respectively. In Section 8, we present our vision for using heuristics about MDN diagram structures to rank results returned by MDN visual queries. We conclude in Section 9.

2. MDN VISUAL QUERY TYPES

As an exploratory navigation example, consider Tom, a student with minimal knowledge of information security. With the increase of university warnings about information security threats such as phishing, keyloggers, and identity theft, Tom wants to explore the subject of information security but he is confronted with a large number of topics, courses, and websites targeting different types of users. Tom starts by searching for 'Phishing' and reaches the phishing page in an MDN-supported Wikipedia interface. After some reading, Tom displays the AG diagram, MDN highlights an element (rectangle) in the AG (with yellow background and red outline) that is related to the phishing page, as shown in Figure 1. This is what a C2D query, where MDN considers the current page as the query parameter and highlights related elements in the diagrams. Using this query, Tom can switch from detailed content to a concise overview. As multiple diagrams can show different perspectives, Tom can start exploration using a diagram that fits his interest or information need.

When Tom selects 'Ransomware' in the AG, a D2D query in MDN highlights related elements in the TT as shown in Figure 2.A. A D2D query helps Tom see concepts of interest from different perspectives (diagrams) and explore different sets of related concepts.

The third type of visual query is D2C which allows end users to select elements in the diagrams and retrieve related pages. In Figure 2.B, Tom selects 'Sell Bot Services' in the AG and 'Backdoor' in the TT to retrieve a list of related pages from Wikipedia.

MDN visual queries can supplement both keyword search (e.g., Wikipedia search) and traditional browsing features such as hyperlinks and menus.

3. MDN COMPONENTS

MDN provides visual navigation of a collection using diagrams that cannot be generated (induced) from the collection. An MDN widget contains features that allow content providers to import and then connect external diagrams to their collection elements (e.g, pages). In the following, we describe how diagrams and collections are represented in MDN, how connections among diagrams and collections are created, and how visual queries are processed.

To facilitate the inclusion of a broad range of diagrams, MDN considers a diagram as a set of *visual elements* that

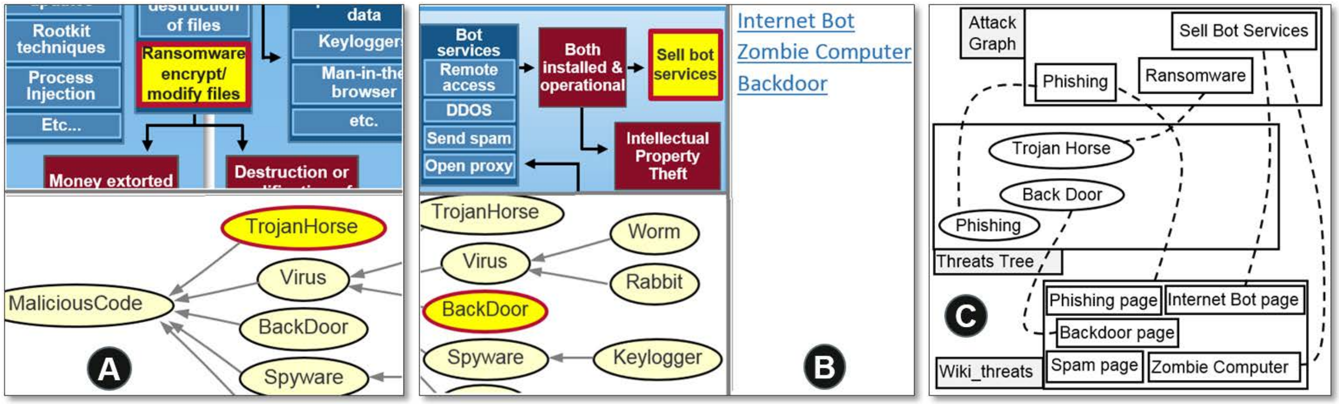

Figure 2: Part A: a Diagram-to-Diagram (D2D) visual query. A user selects Ransomware in the Attack Graph; MDN highlights TrojanHorse, a related visual element in the Threat Tree. Part B: a diagram to content (D2C) visual query. A user selects visual elements to get a list of related results on the right. Part C: An overview of the components of an MDN navigation system

represent concepts such as computer security threats included in the AG and the TT. In the current design, MDN is agnostic to the internal structure of the diagrams such as the process structure in the AG and the hierarchy structure in the TT. Similarly, MDN considers a *collection* to be a set of *content elements*, such as the Wikipedia pages related to the diagrams in the previous section. In Figure 2.C, we show an excerpt from the MDN navigation system explained in the previous section including excerpts of the AG and the TT and a set of related Wikipedia pages, Wiki_threats.

Connections connect visual elements in a diagram to related content elements. For example, 'Backdoor' in the TT is connected to the 'Backdoor' page in the Wiki_threats collection. Connections also connect related visual elements in different diagrams. For instance, in Figure 2.C, 'Phishing' in the AG is connected to 'Phishing' in the TT. MDN can use *connection by composition*. For example, 'Phishing' in Wiki_threats is connected by composition to 'Phishing' in the TT since both elements are connected to 'Phishing' in the AG. Considering all elements in MDN diagrams and collections to be vertices in an undirected graph and connections to be the graph edges, a direct connection is an edge in the graph and a connection by composition is an acyclic graph path.

Using MDN, website owner(s) can manually create connections. This task is relatively easy in small collections. Another approach is to encourage owners and expert users to collaboratively connect elements (e.g., in collections such as Wikipedia). For collections that use domain structures such as ontologies, controlled vocabularies, or classification systems, these domain structures can be connected to MDN diagrams. Then MDN diagrams can automatically connect to the new collection content by composition. For example, when connecting the Phishing visual element in the AG to the Phishing keyword in a controlled vocabulary, the Phishing in the AG can be connected by composition to all documents that reference the Phishing keyword.

The MDN technique is not restricted to a collection of pages and can work with other types of structured and semi-structured data. The only requirement is that the content (or data) elements are identifiable (e.g., using urls in website collections, XPATHs in XML, or keys in structured data).

4. MDN VISUAL QUERY PROCESSING

Query processing in MDN is path following in the MDN graph (e.g., the graph in 2.C) from selected elements in a visual query to connected (related) elements in other diagrams or collections. The graph elements (vertices) are grouped in *clusters* which represent the diagrams and collections in MDN. Each element is represented by a pair (c,e), where c is a cluster id and e is an element id; in this section, we use cluster and element titles as ids for simplicity. For instance, the 'phishing' visual element in the AG is represented by the pair ('AG','Phishing').

Algorithm 1 Get operator Algorithm

Input:(*(C, E), L, TC, VC*) where (C,E) is a selected vertex, L is the maximum path length, TC is the set of target clusters, and VC is the set of clusters visited previously.
Output: a set of vertices related to the input vertex.

1. r = retrieve all vertices *(rc,re)* directly connected to (C,E) where rc is in TC AND rc is not in VC
2.
3. $rComposition$ = empty;
4. **if** $L > 1$ **then**
5.   **for** each (rc, re) in r **do**
6. $rComposition = rComposition$ UNION
7.   Get((rc, re), L−1, TC, VC UNION rc);
8.   **end for**
9. **end if**
10. Return r UNION $rComposition$;

The *Get* query operator is used in all types of visual queries (D2D, D2C, C2D) to retrieve related visual and/or content elements. The Get operator receives four parameters as listed in Algorithm 1. For example, the first visual query shown in Figure 1 is Get(('Wiki_threats', 'Phishing page'), 1, {'AG'}, {'Wiki_threats'}). The first parameter is the selected element. The cluster of the selected vertex, Wiki_threats, is considered the *source cluster*. The second parameter specifies the maximum path length for connection by composition. The third parameter lists the *target clusters* (TC) (or the domain of discourse), specifying where to look for related vertices. The last parameter is the set of visited clusters (VC) in a path from the source cluster to a target cluster; this set is initialized to the source cluster and grows while recursively calling the Get operator; it is used in the algorithm to avoid using the same cluster twice in any path.

The query processing starts by retrieving directly related vertices (line 1). If the query allows using connection by composition (L > 1), processing recursively invokes the Get operator (lines 5-8). The query processor ensures that it does not retrieve any vertices from the current cluster rc in the recursion by adding it to the set of visited clusters.

The Get operator accepts only one selected element. For queries with multiple selected elements, we determine the union of multiple Get operators. For example, the query in Figure 2.B is formed as:
Get((‘AG’, ‘Sell Bot Services’), 1, {‘Wiki_threats’}, {‘AG’}) *Union* Get((‘TT’, ‘BackDoor’), 1, {‘Wiki_threats’}, {‘TT’}).

The MDN prototype used in our preliminary evaluation included support for only a single Get operator per query; a single element is selected by hovering the mouse in D2D or clicking in D2C. In C2D, the current page is considered as the selected content element. Only direct connections (L=1) were used. Our current implementation of MDN, developed after the preliminary evaluation, supports multiple Get operators in D2C and C2D. Future implementations can include intersection and complement, connection by composition, and explicitly selected source and target clusters.

5. RELATED WORK

Much work targeting exploratory search problem has focused on providing overviews (i.e., visualizations) induced from documents, collections, or datasets. For example, Sarrafzadeh [12] extracts a knowledge graph (concept map) from the text of a document. Hall [7] uses thesauri attached to documents in a digital library to create a map-based visualization (“resembling Google Maps”). Alonso [1] extracts a timeline from DBLP search results. Dork [6] used multiple and coordinated visualizations (CMV) [11] by providing a spatial map (based on Google Maps), word cloud, and a timeline for web-search results. Unlike MDN, induced overviews are limited to the type of visualizations provided by the used tools and also the nature of the extracted data. We argue that MDN can facilitate the navigation of collections when interesting aspects of the content cannot be visualized easily (e.g., lack of appropriate visualization tool or when manually created diagrams are more suitable). MDN comes with the cost of creating connections as opposed to induced overviews; we proposed alternative approaches to create connections in Section 3 that suit different types of collections.

The idea of superimposing diagrams builds on the work of superimposed information [9] including the following work in digital libraries. Cañas [4] linked concepts in concept maps to related information resources. Butcher [3] used strand maps, representing ordering of educational learning objectives, as an interface to access digital library documents. We extend this work by allowing different types of diagrams to be used. We also use multiple diagrams and introduce D2D and C2D queries.

6. MDN IMPLEMENTATION

After our preliminary evaluation explained in the next Section, we implemented an MDN widget as a module in the Drupal content management system. The widget supports all visual queries described in Section 2 and allows content providers to load diagrams in SVG (Scalable Vector Graphics) format and create connections. SVG provides user interaction features and is widely supported in diagram editing tools. Diagrams saved in image format can also be utilized by overlaying SVG shapes (representing visual elements) on the diagram image using any SVG editing tool.

7. PRELIMINARY EVALUATION

We conducted a preliminary evaluation through a user study to get an indication of whether MDN is useful and easy to use in exploratory navigation.

7.1 Methods

We used the MDN prototype shown in Figure 1 that included 38 connections between the AG and Wikipedia, 20 connections between the TT and Wikipedia, and 16 connections across the two diagrams. Because of the way we included pages from Wikipedia, cross-site-scripting limitation restricted C2D queries; prototype users had to paste or type a Wikipedia page title in a text box in order for the prototype to highlight visual elements related to that page.

We recruited ten Computer Science students. The user study started with a 15 minute presentation that introduced the features of MDN using two diagrams of the animal kingdom. Then, we briefly introduced the AG and the TT diagrams. The remainder of the study had five parts: the MDN prototype trial, a questionnaire with 2 questions, a first exploratory browsing task using the MDN prototype with only the AG, a second exploratory browsing task using the MDN prototype with both diagrams, and a final questionnaire. The first browsing task involved a comparison between malicious ads and identity theft, including whether malicious attachments could lead to either one. The second task involved comparing phishing and SQL injection. To emulate real exploratory browsing tasks, both tasks were ill structured and open ended as recommended by Wildemuth[13].

7.2 Results

In five questions from the Technology Acceptance Model (TAM) [5]) shown in Table 1, subjects were very positive when they rated the usefulness and ease of use of MDN. For each of these questions, eight or nine subjects answered 4 or 5 (with 5 being ‘strongly agree’ and 1 being ‘strongly disagree’); means are shown in Table 1.

In five questions, shown in Table 2, subjects assessed the different types of visual queries in MDN. The overall response was positive.

To compare the experience of using one versus two diagrams in the two sessions, we asked the subjects to rate four diagram configurations in answering the question “To what extent do you think the following approaches would help a user understand the big picture of information security”. The four methods were using: the AG, the TT, both diagrams, and three or more diagrams (assuming diagrams were useful and relevant). The averages in Figure 3, show that the subjects preferred to use either the AG or both the AG and the TT.

We also logged browsing behaviour of some users; the results from our log analysis suggest differences in browsing behaviour between more and less experienced users. Be-

Table 1: **Questions (from TAM [5])) about usefulness and usability of MDN with mean and standard deviation for the ten subjects**

Category	Question	Mean	Std. Dev.
Usefulness	MDN was useful for me to work on the given tasks.	4.4	0.49
	With the help of MDN I was able to accomplish these tasks quickly.	4.3	0.64
Usability	Interacting with the MDN was clear and understandable.	4.4	0.66
	It was easy to get the MDN to do what I wanted it to do.	4.3	0.78
	I think MDN was easy to use.	4.4	0.8

Table 2: Questions about MDN visual queries: Diagram to Content (D2C), Content to Diagram (C2D), and Diagram to Diagram (D2D) with mean and standard deviation for the ten subjects

Category	Question	Mean	Std. Dev.
D2C	It was natural to navigate to web pages by clicking on boxes and ovals in the diagrams.	3.9	0.7
	Using the diagrams to navigate to the pages helped me explore the content.	4.3	0.78
C2D	I liked seeing my current location (current page), visited pages, and not yet visited pages highlighted in the diagrams [a]	4.3	0.46
D2D	Clicking on the attack graph boxes and seeing related topics in the tree and vice versa helped me learn about information security.	3.9	0.7
	Clicking on the attack graph boxes and seeing related topics in the tree and vice versa helped me choose the next topic to read in detail.	4.5	0.67

[a]The MDN prototype only highlighted diagram elements related to the current page.

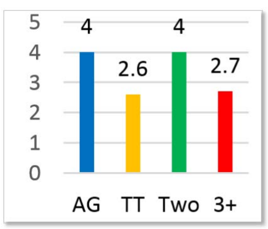

Figure 3: The average response for ten subjects about the preferred number and choice of diagrams when exploring the subject of information security. The choices are: only the AG, only the TT, both diagrams, or 3 or more diagrams.

cause of the small size of the sample and the inability to log detailed user behaviour on the Wikipedia part, we decided to investigate this aspect in more detail and report on it in future studies.

7.3 Discussion

Our findings from the questionnaire shown in Table 1 indicate that subjects perceived MDN as useful in exploring a new subject. This indication is supported by the detailed feedback about two fundamental elements of MDN: using multiple diagrams and navigating through visual queries. The two-diagrams option received a positive response as shown in Figure 3. Knowing that subjects seemed not to like the TT, we think that using a better diagram with the AG might have yielded a better rating. In addition, the high average of D2D query (the last question in Table 2) implies that seeing interesting concepts in different contexts was useful in navigation. Regarding MDN visual queries, the subjects' positive responses on the questions in Table 2 indicate that subjects found D2C, C2D, and D2D queries useful.

The low score for the 3+ option in Figure 3 suggests that using more than two diagrams should be facilitated with diagram browsing and selection. The low score for the TT diagram might be attributable to the layout or the large size of the diagram. This low score emphasizes the importance

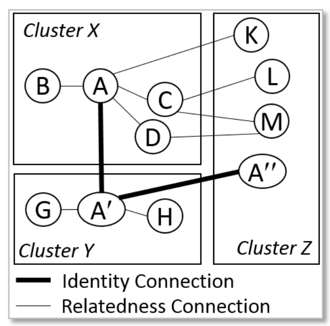

Figure 4: Adding diagram and collection internal connections to the MDN model can increase the number of results returned by MDN queries. MDN could rank these results using heuristics such as length and number of paths between selected elements and elements in the target cluster.

of the quality of the diagrams used in an MDN navigation system.

Usability questions shown in Table 1 also received positive feedback indicating that subjects were able to operate our MDN prototype easily.

8. FUTURE WORK

Our current model does not include connections between visual or content elements in the same cluster; we plan to include these internal connections to represent intra-cluster relationships (e.g., lines in diagrams and hyperlinks in collections). Using internal connections, we can retrieve additional elements from target clusters. For example, in Figure 4, when a user selects element A to retrieve elements from cluster Z, the result includes K, L, M, and A″ where L and M are retrieved using connection by composition through internal connections (A to C and A to D). Including internal connections is very important for elements such as B and G, with no connections to a target cluster (e.g. cluster Z in this case). MDN can still retrieve related elements from the target cluster Z when B or G are selected and internal connections are available.

Exploiting inter-cluster and intra-cluster connections and using connection by composition may make result sets unwieldy. Therefore, we need to rank elements returned by MDN queries based on how relevant they are to the selected elements. One approach to do the ranking is to weight the length, number, and strength of the paths between the selected element(s) and the elements in the target cluster(s). For instance, assuming A is selected in the example in Figure 4, we might rank K before L since K has a shorter path to A, suggesting that K is more relevant to A than L. Also we might rank M before L because M has two paths to A. Another heuristic that we plan to explore is assigning different weights to connections based on their type. For example, we might have identity connections that connect identical concepts (e.g., 'Phishing' in the AG and 'Phishing' in Wiki_threats in Figure 2.C), and relatedness connections that connect related concepts (e.g., 'Ransomware' in the AG and 'Backdoor' in the TT); identity connections would have more weight than relatedness connections. For instance, because A is the selected element in the last example, we may rank A″ before all other elements in cluster Z because A″ has a path with identity connections to A. The final ranking for the example in Figure 4 would be: A″, K, M, then L. This ranking could be used to sort pages retrieved from collections or assign different brightness or color to highlighted elements in diagrams.

In order to combine these different heuristics, we could view a MDN graph, such as the one in Figure 4 as an electric circuit, where a connection weight is the connection's conductance. Identity connections would have more conductance (less resistance) than relatedness connections, to give preference to paths that use such connections. A selected element acts as a battery (i.e., source of current) and all elements in a target cluster are bulbs (i.e., sinks for current). The current that each bulb receives determines its brightness (rank), which is based on the conductance, length, and the number of parallel paths between the batteries and each bulb. We plan to investigate the usage of circuit-simulation tools in future MDN prototyping.

we also plan to investigate different connection types that exist in common diagrams (e.g., parent-child in hierarchical diagrams and next-step in process diagrams). Connection types can be helpful in ranking. For example, assuming clusters X and Y are process diagrams where the type of all connections is next-step (left to right); when a user selects B in cluster X and cluster Y is the target cluster, G is ranked before H because both B and G precede identical elements (A and A′) while H succeeds A′.

9. CONCLUSION

This paper introduced Multiple Diagram Navigation (MDN). MDN allows content providers to superimpose multiple existing diagrams and maps over a collection of related information resources. MDN facilitates exploratory navigation through three types of visual queries: diagram to diagram, content to diagram, and diagram to content. We have presented a design of an MDN system. Our preliminary evaluation of MDN indicated that users perceived MDN as useful and easy to use in exploratory navigation. We also discussed possible heuristics using diagram and collection structures in ranking MDN query results.

10. ACKNOWLEDGEMENT

The first author would like to acknowledge the support of the Libyan Ministry of Higher Education.

11. REFERENCES

[1] Alonso, O., Baeza-Yates, R., and Gertz, M. Exploratory search using timelines. In *SIGCHI 2007 Workshop on Exploratory Search and HCI Workshop* (2007), no. 1, pp. 1–4.

[2] Brinkley, M. Anatomy of today's threats and how to stop them–an architect's view of threats and mcafee protection, 2013.

[3] Butcher, K., Bhushan, S., and Sumner, T. Multimedia displays for conceptual discovery: information seeking with strand maps. *Multimedia Systems 11*, 3 (2006), 236–248.

[4] Cañas, A. J., Hill, G., Carff, R., Suri, N., Lott, J., Eskridge, T., Gomez, G., Arroyo, M., and Carvajal, R. CmapTools: A knowledge modeling and sharing environment. In *Concept maps: Theory, methodology, technology. Proceedings of the First International Conference on Concept Mapping* (2004), vol. 1, pp. 125–133.

[5] Davis, F. D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Q. 13*, 3 (Sept. 1989), 319–340.

[6] Dork, M., Carpendale, S., Collins, C., and Williamson, C. Visgets: Coordinated visualizations for web-based information exploration and discovery. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1205–1212.

[7] Hall, M., and Clough, P. Exploring large digital library collections using a map-based visualisation. In *Research and Advanced Technology for Digital Libraries*. Springer, 2013, pp. 216–227.

[8] Herzog, A., Shahmehri, N., and Duma, C. An ontology of information security. *International Journal of Information Security and Privacy (IJISP) 1*, 4 (2007), 1–23.

[9] Maier, D., and Delcambre, L. M. Superimposed information for the internet. In *WebDB (Informal Proceedings)* (1999), pp. 1–9.

[10] Marchionini, G. Exploratory search: From finding to understanding. *Commun. ACM 49*, 4 (Apr. 2006), 41–46.

[11] Roberts, J. C. On encouraging multiple views for visualization. In *Proceedings of the International Conference on Information Visualization* (Jul 1998), IEEE, pp. 8–14.

[12] Sarrafzadeh, B., Vechtomova, O., and Jokic, V. Exploring knowledge graphs for exploratory search. In *Proceedings of the 5th Information Interaction in Context Symposium* (2014), ACM, pp. 135–144.

[13] Wildemuth, B. M., and Freund, L. Assigning search tasks designed to elicit exploratory search behaviors. In *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval* (New York, NY, USA, 2012), HCIR '12, ACM, pp. 4:1–4:10.