2012 IEEE 19th International Conference and Workshops on Engineering of Computer-Based Systems

On the Extraction and Analysis of a Social Network with Partial Organizational Observation

Sean Whitsitt, Abishek Gopalan, Sangman Cho, Jonathan Sprinkle, Srinivasan Ramasubramanian, Liana Suantak, Jerzy Rozenblit
Electrical and Computer Engineering
University of Arizona
Tucson, Arizona, USA
Email: {whitsitt,abishek,csmcho}@email.arizona.edu, {sprinkle,srini,liana,jr}@ece.arizona.edu

***Abstract*—The behavior of an organization may be inferred based on the behavior of its members, their contacts, and their connectivity. One approach to organizational analysis is the construction and interpretation of a social network graph, where entities of an organization (persons, vehicles, locations, events, etc.) are nodes, and edges represent varying kinds of connectivity between entities. This paper describes a transformation based approach to the extraction of a social network graph, where the original data comprising (partial) observation of the organization are embedded on a graph with a different ontology, and with many entities and edges that are unrelated to the organization of interest. Social network extraction allows the inference of implied relationships, and the selection of relationships relevant for intended analysis techniques. The analysis of the resulting social network graph is based on organizational and individual analysis, in order to permit an advanced user to draw conclusions regarding the behavior of the organization, based on established social network graph metrics. The results of the paper include a discussion of the complexity of analysis, and how the observation data graph is pruned in order to scale the application of analysis algorithms.**

***Keywords*-Social Networking; Network Analysis; Graph Translation**

I. Introduction

Social networking sites are growing in popularity, and have rapidly become a fixture in popular culture. Interestingly, the dynamics of criminal organizations have long been investigated by understanding the connectivity of key and peripheral players in the organization. In the case of inferring the behavioral dynamics of an organization, as well as all realistic scenarios, only *partial* observation of the social network is possible.

The key to all techniques that provide some understanding of the dynamics of an organization based on the social network approach is the *type* of connectivity between entities. This motivates a domain-specific consideration of the graphs used for analysis: namely that the ontology (or, metamodel) used for the analysis should formalize the necessary classifications for the nodes and edges of the graph. The resulting social network graph—suitable for analysis—is a colored multigraph, which can be considered as conforming to a metamodel.

Once the social network graph is created, it is possible to perform two fundamental kinds of analysis: organizational-level analysis, and individual-level analysis. Organizational-level analysis provides insight into the form, efficiency, and cohesion of the organization. Individual-level analysis characterizes the members of an organization, in order to identify key individuals.

Source data, however, may be distributed across multiple databases, and use ontologies not intended for graph-based analysis. Therefore, it is necessary to develop techniques to transform well-formed data from the source ontologies, to the destination ontology, preserving as much information as possible along the way, and inferring missing information.

This work assumes that the magnitude of the source data may exceed the ability of a single analyst to comprehend the connections between entities. Under this assumption, it is imperative that tools exist to scope the source and destination graph structures, such that intelligent growth of the datasets (due to successive queries, for example) is possible. This motivates tools and techniques to explore the large network of connections to find relevant nodes and edges.

This paper describes a transformation-based approach to the extraction of a social network graph. The original data are assumed to be gathered in varying degrees of fidelity, to be a partial observation of the entire organization of interest, and to include significant amounts of data that are irrelevant to the organization of interest. Further, the original data are encoded in an ontology that is not designed for analysis of organizational dynamics.

The key contribution of this paper is the collection of techniques for extraction and analysis of a database not originally intended to be considered as a social network. The paper first describes the matching of an existing dataset's ontology to an ontology suitable for social network analysis. Then, algorithms used for organizational- and individual-level analyses are presented, along with their computational complexity, and the methods used to scale the performance of these analysis techniques on large datasets. Finally, the paper discusses results on the dynamics of the network with respect to time, by showing changes in network density.

978-0-7695-4664-3/12 $26.00 © 2012 IEEE
DOI 10.1109/ECBS.2012.33

IEEE computer society

II. Background

Analysis of social network graphs is a domain-specific application of general graph algorithms to a specific coloring of a graph.

A. Established Social Network Graph Practice

Recent military manuals (e.g., Petreaus) [1] describe how military analysts use the notion of social network graph to describe organizations of interest as part of an insurgency. An important observation is that the organizational expert is responsible for determining graphs and patterns of interest: it is not possible to infer graphs and patterns of interest based solely on large example datasets, due to problems of partial observation. Inference must be used at some point, and such inference must be validated by the expert.

The metrics and algorithms that this paper will discuss are as follows:

- Social Network Extraction
- Clustering
- Closeness Centrality
- Degree Centrality
- Betweenness Centrality
- Information Brokers
- Central Players
- Peripheral Players

These metrics are based chiefly on their definition in [1], and in that reference, the structure used to express the social network has a specific ontology. This paper later discusses how to transform various data sources into such an ontology for analysis.

B. Examples of Realistic Data Sources

The research in this paper is focused on providing tools to analyze large intricately connected networks. Fortunately, social media websites provide a perfect source of very large and interconnected networks. As a result, this research makes use of data collected from users on Facebook. We note that the participants in this research were asked to provide data to test the speed of algorithms, and not to do any human subjects research. Participants opted in using the standard Facebook interfaces to our application. Participants understood that data from their news feeds and on their friends would be collected and used for the testing of these algorithms.

Using Facebook as a data source for social network analysis is not a new concept. For instance, [2] mined Facebook data to provide examples of a large-graph structure for research into parallel computing models. [3] is a presentation of a system meant to manage privacy information in a social network (specifically Facebook). The FAITH system developed in [3] also allows users to transform the social graph that each Facebook application can see. [4] presents data on the social properties of player-to-player interactions developed in social media applications.

C. Existing Tools and Techniques

UCINET [5] is a tool used by social network analysts to visualize and understand social networks. UCINET presents a user with many of the same or similar algorithms presented in this paper, but in a way that is more useful to experienced sociologists than to intelligence analysts. The key difference between these two groups of individuals being that sociologists understand social network analysis and why they might use the algorithms presented herein whereas the intelligence analysts have other specialties and could validate results, but may be unable to construct the graphs or algorithms.

The methodology presented in [6] presents a similar approach to centrality as is presented in this paper. However, this method could be considered a different centrality measure than the three presented in this paper. Also, the research in [6] was built around developing and presenting a novel approach to measuring the centrality of nodes in a social network while this research is geared towards presenting existing techniques for large-scale networks to intelligence analysts with no social network analysis experience.

Carley's work in [7] presents analysis techniques for networks that may have dependencies between segments of data that might impair or invalidate statistical methods of analysis. [8] demonstrates an approach to extract covert social networks from texts. Both [7] and [8] are novel concepts, but both are geared towards building an understanding of networks. Similarly, this paper is meant to present a set of tools that can be used to gain an understanding of social networks. However, the methods presented in [7] and [8] would be techniques that could be applied to the networks that are generated as a result of parts of the research presented in this paper.

D. ATRAP: An Environment for Intelligence Analysts

Asymmetric Threat Response and Analysis Program (ATRAP) is used to visualize and perform analysis on information gathered from disparate sources [9], [10]. The tool has various use cases: (i) to ingest information into a database; (ii) to query that information based on ranges of date, geography, etc.; (iii) to visualize results of a query geographically, in time, or by association of database entities.

ATRAP is used as the implementation framework for our algorithms, due to the ease of use of its databases, and its ability to expand/reduce queries based on entities and their relationships. All of the screenshots in this paper are taken from ATRAP.

III. Network Extraction

The ideal analyzable graph model is a well connected graph that can be guaranteed to contain all relevant data, and only relevant data. However, in practice, such a perfect example can never be expected from raw data. Instead, for this paper we will describe a process by which a network that is more suitable for analysis can be extracted from existing

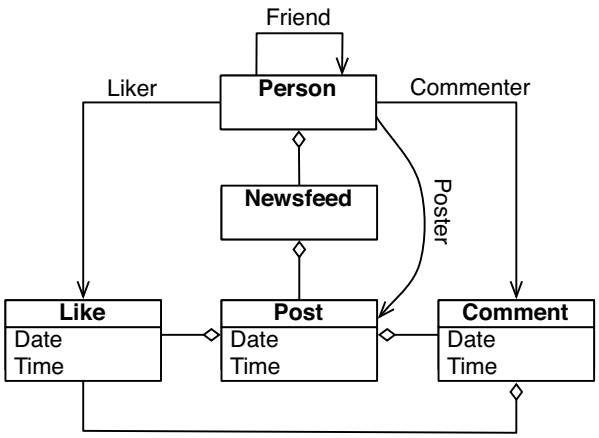

Figure 1. The metadata structure from Facebook that this paper is using as an example data set. Nodes are represented in a UML class style with open arrows representing edges and diamonds representing containment. Attributes that are collected as part of the data in the graph are noted in the classes.

data structures. In short this process involves removing explicit connections that do not contribute to the overall understanding of the network, and inserting by inference new implicit connections that do.

This process also serves a secondary function for the purposes of this research. The final application developed during the cycle of this research contains data that is of a different structure than the Facebook examples presented in this paper. For this reason it was beneficial to develop an interface for converting other data structures to the data structure that the algorithms presented herein expect. As a result the algorithms could be developed independently of the requirements of either the example data structures or the final product's data structures.

A. Graph Model

The most suitable network for the analysis methods that this paper describes is one that contains only relevant social entities (e.g. people) with relevant social connections (e.g. relationship statuses). Since the example data set being used in this research is from Facebook, the reader may expect that this data is already apparent. Unfortunately, that supposition assumes that the example data set comes from a wide variety of people that are all interconnected with minimal external connections. Also, in order to make this research relevant to more data sets than just Facebook (or other social networks) data, a more generalized approach has been developed.

Fig. 1 shows the raw data's desired metadata structure that will be translated into a more analyzable data structure. This structure shows how persons on Facebook are related by friendship and through comments and "likes" on status updates and news posts.

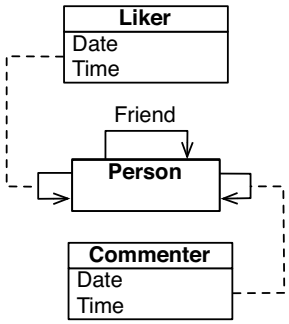

Figure 2. The end-result metadata structure that is generated from the example data set. Note that the "Liker" and "Commenter" relationships from Fig. 1 are now shown as relation classes so that the attributes that are kept from the original metadata structure can be shown.

B. Analyzable Graph Organization

Fig. 2 shows the desired metadata after extraction from the example data set. The social graph obtained from Facebook users is reduced in complexity to only show persons and the ways in which persons are related. This generalized translation process from an example data set allows the methods and algorithms described in the following sections to be usefully applied to more data sets than just one specific example (e.g. the Facebook example used in this paper). The only catch in this process is that each new example data set has to be properly translated to the metadata type described by Fig. 2.

C. Graph Translation

The graph translation algorithm for this research works by searching the original social graph for certain patterns and adding in a connection between the two end points of the pattern. A pattern in this case is itself a graph, but is limited to being a string of entities. A string of entities is a set of more than one connected entities with no loops and where each entity has at most two connections and at least one connection. The algorithm begins by examining each entity in the graph and comparing it to the first entity in the pattern. If the graph entity matches the pattern entity the algorithm then branches out to neighboring entities and compares those to the neighboring entities in the pattern. The algorithm repeats this comparison routine until the entire pattern has been matched. If the pattern has been matched a new connection can be established between the first matched entity and the last matched entity. This gives a user the ability to specify implicit connections between different entities in the graph. Extraneous non-social connections and entities can then be removed from the graph leaving behind a new graph unique to the purposes of the user.

As an example, assume there is a network that shows membership of individuals in a few organizations with

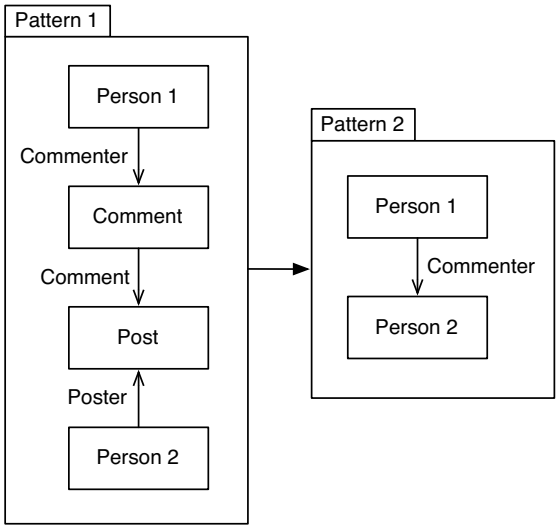

Figure 3. A graph translation pattern (Pattern 1) and the pattern to add (Pattern 2) into the graph structure. Each pattern is a list of interconnected nodes. Note that the concept of containment shown in Fig. 1 is not shown here. The network structure is such that containment in the metadata structure is represented as an edge between the contained node and the containing node. Node and edge types are denoted where appropriate, but be aware that "Person 1" and "Person 2" denote unique nodes of type Person.

only links between the organizations and the individuals (no connections between individuals). An individual may be a member of multiple organizations, and thus have connections between multiple organizations. A pattern may then be created that is defined as a person connected to an organization which is then connected to another person. Applying this pattern to the network would then create links between all of the individuals in a given organization. Then, removing the organizations and the connections between organizations and individuals would create a network of only individuals with connections to other individuals.

The reader should understand at this point that the tool developed through this research leaves to the user the tasks of defining patterns, adding implicit connections, and deciding which entities and connections to remove. Fig. 3 shows one of the patterns used to reduce the example data set (Fig. 1) down to the example social data set (Fig. 2).

This particular metadata example is not very complicated (it only requires three patterns). However, if this example is extended a bit to consider a graph where the friend connections do not exist (i.e. the data set contains no knowledge of friendships between different people) then the data it does contain can be used to at least partially reconstruct the information on friendship connections. In other words, the data on how individuals are connected together via comments left on news posts and "likes" of news posts can be used to build a graph that represents

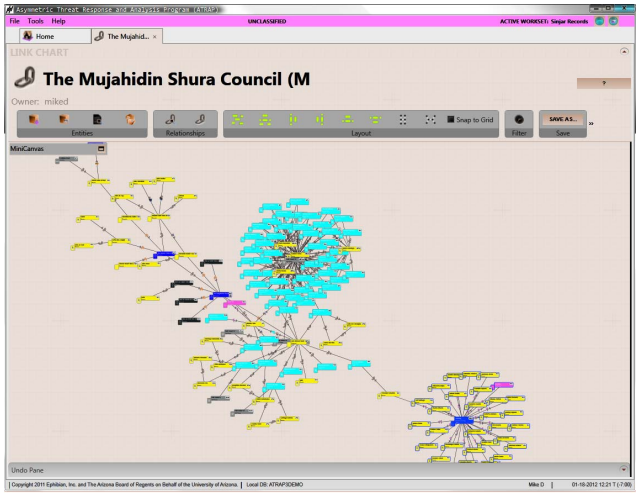

Figure 4. An example raw network in the ATRAP utility before extracting a social network. Light blue nodes are events, blue are organizations, purple are equipment and the remainder are individuals. This figure is meant to demonstrate the complexity of a large network before it gets reduced to a social network.

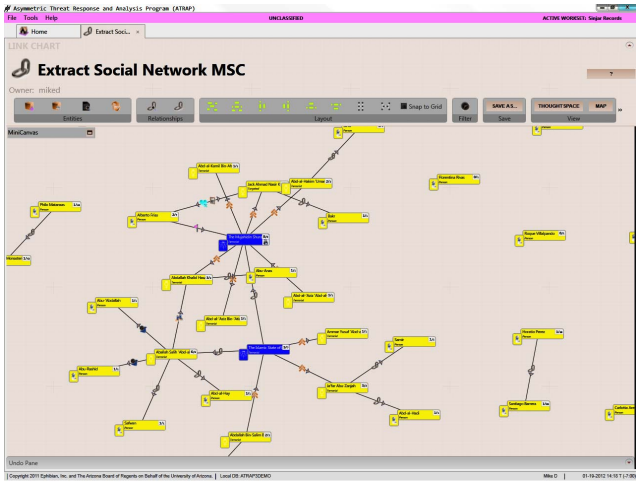

Figure 5. An example social network that was extracted from the network in Fig. 4. Yellow nodes representing individuals and blue nodes representing organizations are the only remaining nodes. This figure is meant to show the reduced complexity of the network in Fig. 4.

possible friendship relationships between those individuals.

Fig. 4 and Fig. 5 show an example network in the ATRAP utility, pre-network extraction and post-network extraction (respectively). This example illustrates the reduction in complexity that is expected with networks.

IV. Analysis

We consider an undirected network represented by a graph $\mathcal{G}(\mathcal{N}, \mathcal{L})$ where $\mathcal{N}$ denotes the set of nodes and $\mathcal{L}$ denotes the set of links. The number of nodes and links are represented by N, L respectively. We now consider two different ways; organizational and individual level of

analyzing the social network to gain insight on portions or nodes of interest. The former tries to make inference at the global structure or the entire network while the latter operates on a finer granularity in that it tries to identify individual persons or objects of interest in the social graph. Some of the schemes that follow are well understood and considered standard techniques in social network analysis, and we call attention to our novel contributions where appropriate. However, in the interest of completeness, we describe all of them briefly.

A. Organizational level analysis

1) Network Density Over Time: Network density(ρ) is a measure of the number of links in the network relative to the number of maximum possible links in the same network [1].

$$\rho = \frac{2L}{N(N-1)}$$

In a social network, relationships need not be static. For instance, a node representing a person can change associations with organizations over time. This dynamic nature resulting in several ties being severed or added over time can be directly captured by the appearance or removal of links in the social network. Then, it is of interest to study the behavior of the network density over time. This can be a powerful tool in studying the sudden changes in the importance of people, relationships etc. Assuming the presence of temporal data, namely which links are active at any instant of time, it is possible to compute the network density at several instances. The granularity at which this is required varies depending on the type of changes that an analysis wishes to track.

Running time: Computing the network density at one time instant takes constant time $O(1)$. Thus, the cost of computing it at several instances (say d of them) takes time $O(d)$.

2) Clustering and Cliques: A social network that has been extracted from partial observation can still be fairly large in size. Thus, analysis that looks for patterns in the entire network can be a fairly cumbersome task. This motivates the need to identify clusters or sub-graphs in the network. Once this is done, the analyst can focus on studying the behavior and properties within this network. A good example is one of identifying a single organization among several in a social network.

The goal of clustering a social network is to group nodes that have several interconnections amongst themselves. This captures closely knit sub-groups or individuals. From a graph theoretical perspective, finding such groups can be modeled as finding cliques. However, computing a maximum clique or listing all maximal cliques in a network is NP-hard and decision versions of the problem are known to be NP-complete [11]. To this end, we employ an algorithm that is simple and fast to execute while compromising on the quality of the clusters. Further, we provide a handle on this quality which can be used by the analyst to look at clusters of desired sizes.

We define a cluster as a function of two variables K and F. K is termed as the clique approximation factor while F denotes the cluster size factor. Both variables are fractions that range between 0 and 1.

A cluster is a sub-graph satisfying the following properties:

1) Each node in the sub-graph has at least $K(N-1)$ neighbors
2) The number of links in the sub-graph is at least $F(N(N-1)/2)$

As an example, to check if the the given network is a complete graph, one can simply set both K and F to 1 and check if a cluster can be found. A more useful scenario is one in which an analyst wishes to identify a densely connected sub-graph[1] or several sub-graphs of a certain size. The handle on K, F help achieve exactly this.

Running time: Computing the clusters for a given K, F can be done in $O(N+L)$. This is achieved by computing all nodes that satisfy the neighbor constraints in $O(L)$ followed by computing the connected components on this restricted sub-graph in $O(N+L)$ using a tree traversal algorithm such as breadth first search (BFS) [12]. Finally, it only remains to check if the number of edges in each of these components satisfy the second condition for a cluster which can be done in time $O(L)$.

Remark: A more interesting analysis ensues when an analyst wishes to find the largest sub-graph in which each node has a certain degree. That is, we wish to compute the largest F for a given K. There are a total of potentially L discrete values for the cluster size[2]. By doing a binary search[3] on the range of F we can compute the largest sub-graph for a given K in $O(logN)$.

B. Individual level analysis

We now discuss algorithms to identify nodes in the network that are likely to be influential and are often the centers of information flow. It is standard practice in social network analysis to study centrality scores for each node in the graph to achieve this.

1) Degree centrality: The degree centrality of a node is the number of links incident on the node.

$$C_d(v) = degree(v)$$

Running time: The score for all nodes can be computed in overall time $O(L)$ by maintaing an adjacency list representation for the graph.

[1]The maximum clique ideally.

[2]$L \leq N(N-1)/2$

[3]We can compute the smallest value of $F(N(N-1)/2)$ for which there is no cluster. The largest integer smaller than the value above is the cluster size.

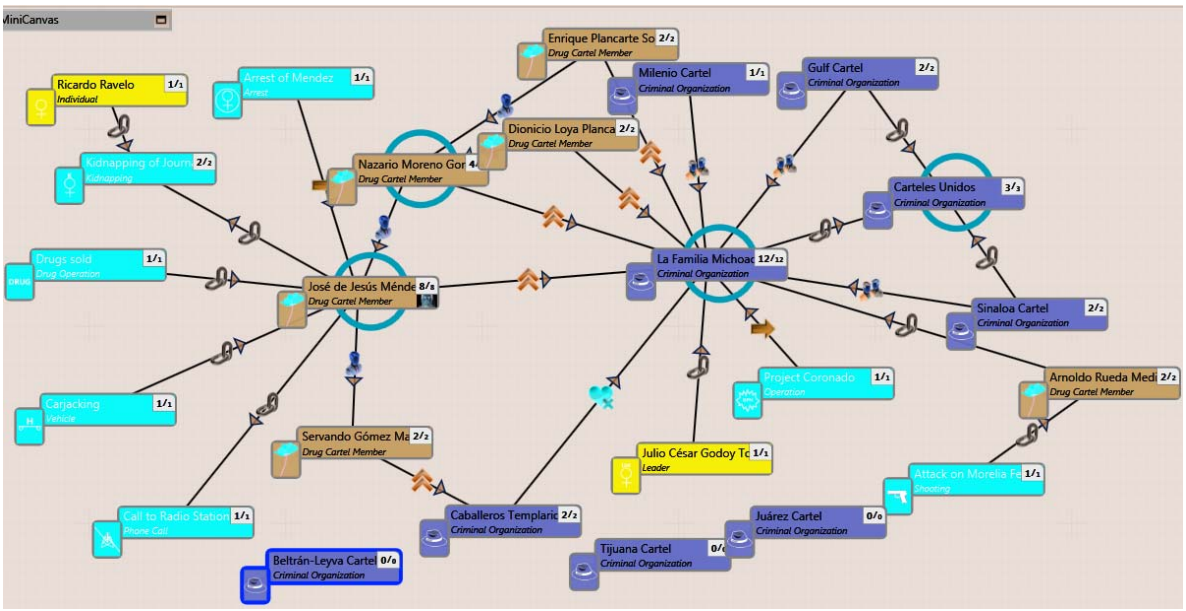

Figure 6. After running the Central Players algorithm, the calculated central players are indicated by circles. Note that the names and events in this network are fictional.

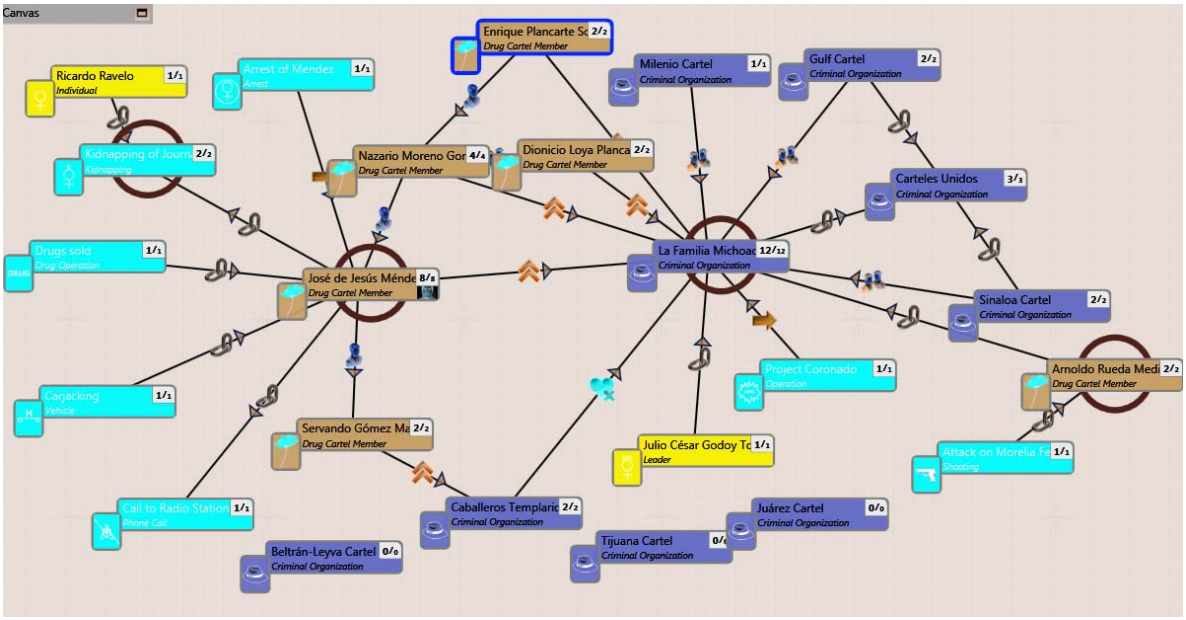

Figure 7. After running the Brokers algorithm, the calculated information brokers are indicated by circles. Note that the names and events in this network are fictional.

2) Closeness centrality: In the context of a social graph, the measure of closeness is quantified in terms of shortest path distances. Let $d(u, v)$ denote the shortest path distance from node u to node v in the network. Then the closeness centrality for a node v is given by

$$C_c(v) = \frac{N-1}{\sum_{\forall x \in \mathcal{N}} d(v,x)}$$

This score for a node v represents the average shortest distance cost from v to any other node in the network.

Running time: For a single node, the shortest distances to all other nodes can be computed by running Dijkstra's algorithm in $O(L+NlogN)$. Thus computing centrality for all nodes in the network takes time $O(NL + N^2logN)$. The same asymptotic running time can also be achieved by running Johnson's algorithm [12] to compute all-pairs shortest paths on the graph. In case of unweighted graphs, a BFS is sufficient to discover shortest paths from a node. The overall time complexity in that case is $O(NL + N^2)$

3) Betweenness Centrality: The betweenness of a node is a measure of the number of shortest paths in the network that pass through the node. Let σ_{st} denote the number of shortest paths from s to t and $\sigma_{st}(v)$ denote the number of shortest paths from s to t that pass through v. Then the betweenness centrality for a node v is given by

$$C_b(v) = \sum_{s \neq t \neq v \in N} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Running time: Brandes [13] algorithm can be employed to compute the betweenness centrality for all nodes in $O(NL)$ for unweighted graphs and in $O(N^2logN)$ for weighted graphs.

4) Central Players: A central player as the name suggests is someone who has several connections to other nodes in the network. These nodes are likely to indicate individuals that are important to the network in question, but not to elements outside of the network. They are possibly centers of information flow and likely to control large portions of the network. Among the centrality scores discussed above, both degree centrality and closeness centrality have a positive correlation for a node to be identified as a central player. Higher scores are indicative of nodes being more central.

From an analyst's perspective, sorting these scores (degree, closeness) in decreasing order gives an insight of the most central players in the network. Depending on the nature of the application, it could be of interest to remove or protect such key players.

Fig. 6 shows a small example network with four central players indicated by circles as seen in ATRAP.

5) Information Brokers: Brokers are nodes that tend to control the flow of information across different regions of the network. Brokers typically have high betweenness scores since the centrality is a measure of the number of (shortest) paths through a given node. Thus, these liaison nodes see a lot of information flowing through them.

As an analyst, one might want to identify such points of interest for spying or control purposes. It is also possible that the analyst would wish to remove such brokers to limit or damage the communication with the organization.

Fig. 7 shows a small example network with four brokers indicated by circles. This is the same network as shown in Fig. 6 except after running the Brokers algorithm. The difference between information brokers and central players may not be entirely evident from this small example, but note that the central players in this network appear to be surrounded by many other nodes while the brokers tend to be the main points at which information will travel in the network. For instance, the broker circled at the top left of Fig. 7 is a broker because that node is between one other node and the rest of the network. This particular broker is not central in the network, but any information flowing between another point in the network and the yellow node labeled "Ricardo Ravelo" must travel through this broker.

6) Peripheral Players: Peripheral players are quite the opposite of central players in a network. These nodes tend to have limited connections and control over the network in question. These are identified by looking up nodes that score low on degree, centrality and betweenness. However, an important aspect for an analyst to consider is the role of such nodes outside the scope of the network in question. Peripheral players are likely to be brokers in unmapped networks [1] and could potentially control flow of information across different networks. Thus, they could be key players at a coarser scale of analysis.

V. Conclusion

This paper described methods of extracting a social network from data structures not originally intended to be analyzed using metrics for the behavior of an organization. Novel approaches to graph analysis algorithms are used in order to permit scalability of common metrics, and these approaches leverage the concept of the social network to reduce the size of the graph under consideration. Several examples were shown which demonstrate the ability of the ATRAP tool to integrate these social network concepts into its repertoire of use cases, and to utilize its existing visualization engines. With these results, analysis of the behavior of an organization is not fully automated: rather, the domain expertise of an analyst is used to reduce the workload required to consider the social network of players in an organization.

Acknowledgments

ATRAP is based upon work supported by the G9 Prototype under Task Order Numbers 9T7ZDAIS705,

9T8ZDAIS803, 9Q9SDAIS903, and 9Q0SDAIS003. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s), and do not necessarily reflect the views of the G9 Prototype or the US Army Intelligence Center of Excellence.

REFERENCES

[1] D. Petraeus, "The U.S. Army and Marine Corps Counterinsurgency Field Manual," Department of the Army, Tech. Rep. FM 3–24, MCWP3–33.5, Dec. 2006, Appendix B.

[2] B. Wu, Y. Dong, Q. Ke, and Y. Cai, "A parallel computing model for large-graph mining with mapreduce," in *Natural Computation (ICNC), 2011 Seventh International Conference on*, vol. 1, Jul. 2011, pp. 43 –47.

[3] R. Lee, R. Nia, J. Hsu, K. Levitt, J. Rowe, S. Wu, and S. Ye, "Design and implementation of faith, an experimental system to intercept and manipulate online social informatics," in *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, Jul. 2011, pp. 195 –202.

[4] B. Kirman, S. Lawson, and C. Linehan, "Gaming on and off the social graph: The social structure of facebook games," in *Computational Science and Engineering, 2009. CSE '09. International Conference on*, vol. 4, Aug. 2009, pp. 627 –632.

[5] B. MacEvoy and L. Freeman, *UCINET: A Microcomputer Package for Network Analysis*. Mathematical Social Science Group, 1987.

[6] T. Crnovrsanin, C. Correa, and K.-L. Ma, "Social network discovery based on sensitivity analysis," in *Social Network Analysis and Mining, 2009. ASONAM '09. International Conference on Advances in*, Jul. 2009, pp. 107 –112.

[7] K. Carley and D. Banks, "Nonparametric inference for network data," *The Journal of Mathematical Sociology*, vol. 18, no. 1, pp. 1–26, 1993.

[8] J. Diesner and K. Carley, "Using network text analysis to detect the organizational structure of covert networks," in *Proceedings of the North American Association for Computational Social and Organizational Science (NAACSOS) Conference*, Jul. 2004.

[9] M. Valenzuela, C. Feng, P. Reddy, F. Momen, J. Rozenblit, B. ten Eyck, and F. Szidarovszky, "A non-numerical predictive model for asymmetric analysis," in *Engineering of Computer Based Systems (ECBS), 2010 17th IEEE International Conference and Workshops on*, Mar. 2010, pp. 311 –315.

[10] E. Chan, J. Ginsburg, B. Ten Eyck, J. Rozenblit, and M. Dameron, "Text analysis and entity extraction in asymmetric threat response and prediction," in *Intelligence and Security Informatics (ISI), 2010 IEEE International Conference on*, May 2010, pp. 202 –207.

[11] R. Karp, "Reducibility among combinatorial problems, complexity of computer computations," *Proc. Sympos. IBM Thomas J. Watson Res. Center*, pp. 85–103, 1972.

[12] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.

[13] U. Brandes, "A faster algorithm for betweenness centrality*," *Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001.