A layered approach to link analysis and visualization of event data

Conference Paper · August 2012

DOI: 10.1109/ICDIM.2012.6360101

CITATIONS
5

READS
36

5 authors, including:

Yain Whar Si
University of Macau
88 PUBLICATIONS **352** CITATIONS
SEE PROFILE

Simon Fong
University of Macau
584 PUBLICATIONS **2,652** CITATIONS
SEE PROFILE

Robert P. Biuk-Aghai
Software Company
89 PUBLICATIONS **444** CITATIONS
SEE PROFILE

http://dx.doi.org/10.1109/ICDIM.2012.6360101

A Layered Approach to Link Analysis and Visualization of Event Data

Yain-Whar Si[1], Se-Hang Cheong[2]

Department of Computer and Information Science
University of Macau
fstasp@umac.mo[1], dit.dhc@lostcity-studio.com[2]

Simon Fong[3], Robert P. Biuk-Aghai[4], Tat-Man Cheong[5]

Department of Computer and Information Science
University of Macau
ccfong@umac.mo[3], robertb@umac.mo[4], harry.cheong@gmail.com[5]

***Abstract*—This paper presents a layered approach to analysis and visualization of associations from events. The proposed approach provides different levels of abstraction for aggregating and analyzing events from heterogeneous data sources by using lists and customizable functions. Interfaces for creating lists and functions are also implemented for different levels of users. The effectiveness of the proposed approach is demonstrated through examples from criminal network analysis area.**

Keywords–link analysis; event mining; association visualization; criminal network

I. INTRODUCTION

Sequence of events can be used to depict a crime, the spread of an infectious disease, or day-to-day activities of a person or a business. Although different attributes can be used to describe an event, we can basically categorize them into two fundamental groups: entities and time. Examples of entity include name of the criminals, addresses, weapon used in the crimes, transaction amount, illicit drugs, name of the diseases, etc. Time can be represented as a timestamp, or an interval.

Organized crimes such as drug trafficking and money laundering involve extensive criminal networks and illicit activities are often carried beyond a nation's territories. In these criminal networks, offenders are usually connected to other members of the network via various relationships such as co-workers, friends, business partner and kinship. In addition, explicit or implicit relations can be identified among the persons involved, various entities within the networks, and the time of these events. For instance, an event on transferring money from one bank account to another by a suspect involves the names of the sender and recipient, their bank accounts, and the transaction time.

To uncover the association among criminals or criminal networks, criminal investigators often rely on link analysis tools [11]. In link analysis, investigators aggregate relevant information from raw data. The aggregated information is then processed and presented in a structured format. Visualization of any relationships identified among the entities is also provided in these tools. In recent years, a number of link analysis systems have been developed. These systems include NETMAP [1], Analyst's Notebook [2], and COPLINK[3].

Link analysis involves searching and aggregating heterogeneous databases, analyzing crime reports, and fine tuning of results based on expert knowledge. Some link analysis tools also provide interactive functions for manually visualizing the structures of criminal networks. These systems often require the users to manually input the associations between the entities. These manual tasks could be time-consuming and error-prone.

In order to alleviate these problems, a layered approach to link analysis and visualization of event data is proposed in this paper. We show a system that adopts the layered approach for processing event-based records. The system consists of modules that create lists of aggregated entities from different data sources. The system also provides user-interfaces for composing tailored-made functions. These functions are used for detection of links and calculating degree of association. A module for association network visualization is also implemented. The prototype system is implemented in JAVA and JUNG[12].

The paper is structured as follows. In Section 2, we briefly describe the overall system design. In Section 3, we describe the creation of lists for aggregating various data entities. In Section 4, tailored-made functions for link analysis are introduced. In Section 5, visualization of identified links extracted from case studies of several terrorist attacks is detailed. In section 6, we review recent work on link analysis and management of event data. In Section 7, we conclude the paper with future work.

II. SYSTEM OVERVIEW

The overview of the proposed system is given in Figure 1. The main aim of the proposed system is twofold. First, we aim to provide users with a layer of abstraction for aggregating and preprocessing data from different sources. By using SQL queries, data from different external databases are aggregated into *Lists.* This function for aggregation is provided via the programmer interface since it involves detail understanding of underlying database schemas. Second, we aim to provide users with the ability to compose tailored made functions for link analysis and visualization. To achieve this objective, an interface is provided to create commonly used fundamental functions by the programmer. The programmer interface of the proposed system is described in the lower part of Figure 1.

The next level (Function Evaluation) in the proposed system is intended for application users. These users can be criminal investigators, medical experts in infectious disease detection and control, or users who are interested in investigating relations without going into database schema. They may only use abstract functions to study aggregated data and to analyze associations. Visual mining of any discovered networks and links can be done through, a user-interface; the overall design of the system is shown in Figure 1.

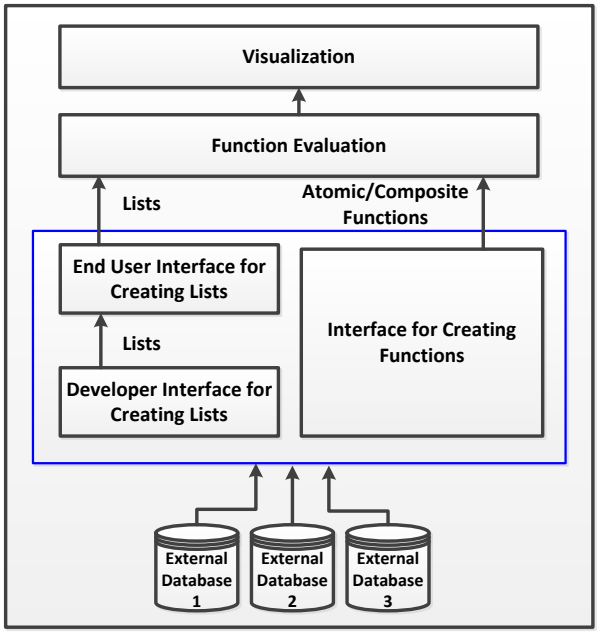

Figure 1. System Overview

III. LIST OF ENTITIES

In the proposed system, events that are extracted from different data sources can be aggregated based on a simple data structure called Lists. For instance, name of the persons involved in a specific criminal network can be aggregated from different data sources by defining a list called "Suspects". The aggregation allows the users to compose any kind of lists from existing databases. Examples of lists for criminal network analysis include "Addresses", "Crimes Committed", and "Victims". The concept of creating lists is depicted in Figure 2.

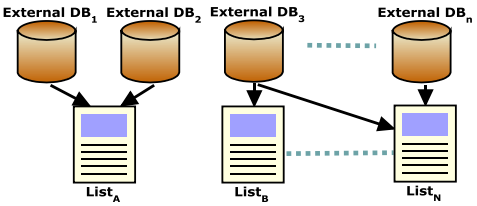

Figure 2. Creating lists from external databases

In general, Lists are aggregated from heterogeneous databases based on Java and JDBC (Java Database Connectivity). First, we use JDBC to connect heterogeneous databases. Next, we select specified columns as the entities of a List from databases whereby the data transforamtion and query translation are done by JDBC. Then, we perfrom a distributed join and merge of entities of the List if necessary. After distributed join and merge operations, the proposed system is able to compose the contents of the List.

A. *Developer's Interface for Creating Lists*

In the developer's interface, lists can be created by defining SQL queries spanning over different databases. The interface is intended for programmers or system administrators who are proficient in database programming. We assume that the developers are aware of the schema of the underlying data sources. The screen shot of the interface is shown in Figure 3.

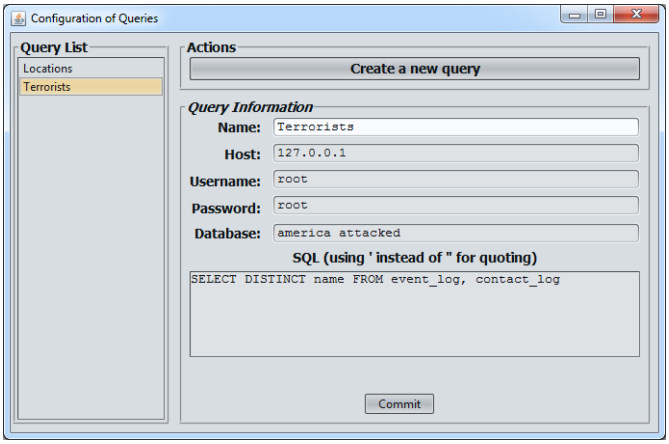

Figure 3. Developer's interface for creating lists

B. *End User's Interface for Manipulating Sets*

Lists created by developers can be further refined in the end user's interface. The screen shot of the interface is depicted in Figure 4. For instance, the list "911attackerlist" can be composed from the list "Terrorists". Through this interface, lists can be built by filtering or aggregating other lists.

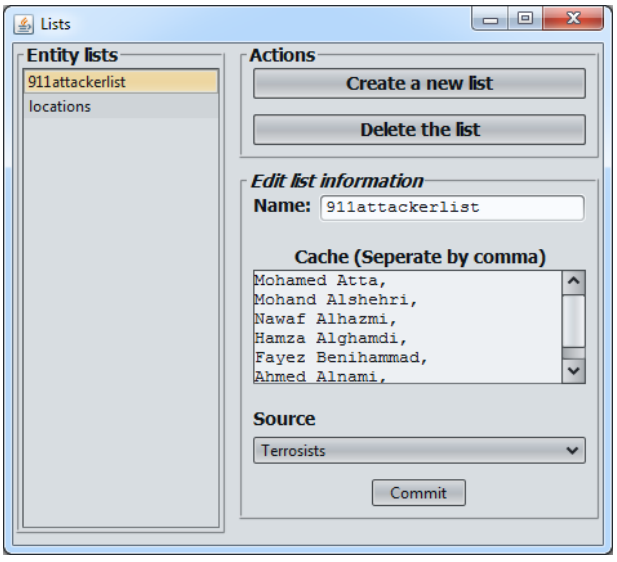

Figure 4. User's interface for manipulation of lists

IV. FUNCTIONS

Functions are used for deriving a conclusion from given data. For an instance of infectious disease detection problem, a function called "Infection Probability" can be defined to calculate the likelihood of spreading the disease from one person to another. The function can be designed to take into account the relationship between the persons, contact frequencies, and the places where they visited in the past. Functions can be designed as either atomic functions which are intended for simple calculation tasks or composite functions which are composed from multiple atomic or other composite functions and additional event data. The proposed system also allows defining functions with procedure call for executing external programs. Atomic and composite functions can be defined based on the following formats:

AtomicFunction(ListA,ListB,Operator, Ts, Te).

CompositeFunction(Function,ListA,ListB,Operator, Ts, Te).

TABLE I. PARAMETERS FOR A FUNCTION

Parameter Name	Description
ListA	Predefined list A.
ListB	Predefined list B.
Operator	For comparing the data from $List_A$ and $List_B$.
Ts	Start time of the evaluation period.
Te	End time of the evaluation period.
Function	Name of an atomic or composite function.

The supported operators are listed in TABLE II.

TABLE II. SUPPORTED OPERATORS

Operator	Description
=	Equal
>	Greater than
<	Less than
>=	Greater than or Equal to
<=	Less than or Equal to
!=	Not Equal
~=	Partially Equal

Ts and Te are used to indicate the start and end time of an analysis period respectively. For example, anatomic function called "Meet" can be defined to determine any pair of persons who have ever met at a specific location.

Name = {Alice, Bob, David,..}

Location = {HotelX, HotelY,..}

Ts = 2011-07-01 00:00

Te = 2011-07-31 23:59

Based on these input, the function Meet(Name, Location, =, Ts, Te) returns a list of records where each record contains three attributes: name of the first person, name of the second person, and a Boolean value indicating whether a meeting has been taken place during Ts to Te. The developer interface for creating customized functions is depicted in Figure 5. Syntax similar to SQL for creating functions is provided for the developer. The grammar for the syntax is depicted in the lower part of Figure 5.

During the evaluation of the function, the parameters defined in the function are substituted with their real values and the entire statement is transformed into a SQL statement for execution. For example, the function "Meet(Name,Location, =, Ts, Te)" is translated into the following SQL statement.

```
SELECT COUNT(*) > 0 FROM db_relation D1,... WHERE
D1.person[OPR]"[ListA]" AND D1.location[OPR]"[ListB]"
AND D1.starttime>="[TS]" and D1.endtime<="[TE]" ...
```

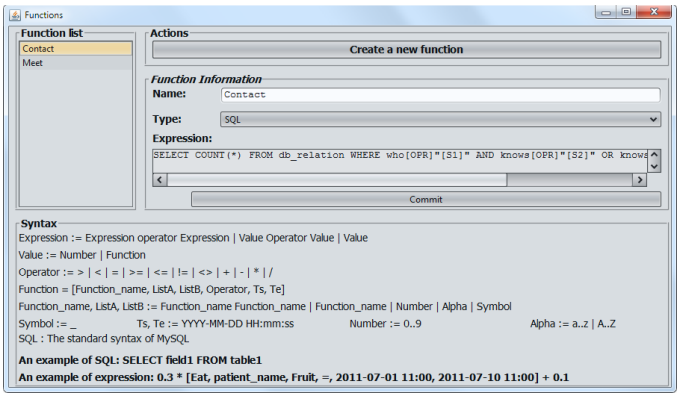

Figure 5. Interface for creating functions

Link analysis function uses input lists and other customized functions to produce required results. Moreover, it can export formatted data, such as XML for further data visualization. The interface for link analysis function is depicted in Figure 6.

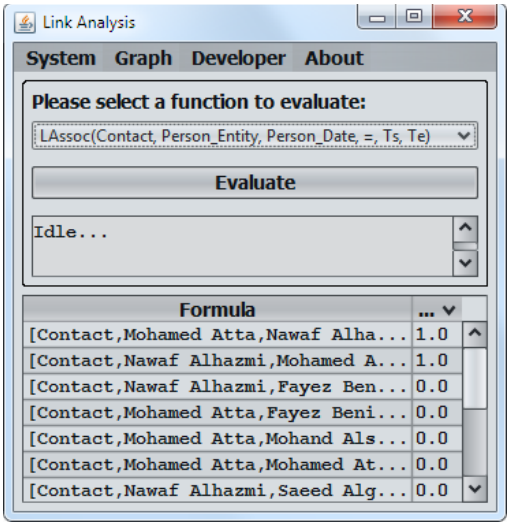

Figure 6. Evaliation of functions for analyzing September 11 attack

The function named "LAssoc" (Link of Association) for analyzing the September 11 attack can be composed as follows. First we define the lists which store names, entities, and dates that are extracted from the September 11 attack [10]. For the sake of simplicity, all information regarding actions, places, and objects involved in the attack are treated as entities.

Attackerlist = {Mohamed Atta, Alghamdi, ...}

Entitylist = {Pilot license, Driver license, ...}

Dates = {2001-09-01, 2001-01-04, ...}

Next we define an atomic function called “Contact”. The function generates an output list by performing pairwise analysis of these lists based on underlying databases. The output list contains a set of records including attackers’ names, entities and a Boolean value 1 or 0 to indicate whether an association has been detected between the person and the entity. Start and end dates are not specified in this function since we are only interested in finding the associations between persons and the entities. In contrast to the function “Meet”, the location information is also omitted in the definition. The function “Contact” can be conceptualized as follows:

Contact(Attackerlist, Entitylist, =, NIL, NIL)

The underlying SQL statement for function “Contact” is depicted in Figure 5. Likewise, we can also conceptualize an atomic function called “Do” for generating an output list by performing pairwise analysis of “Attackerlist” and “Dates”.

Do(Attackerlist, Dates, =, Ts, Te)

In this function, Ts and Te represent the start and end time of the period for performing link analysis. Next we can define composite functions “Person_Entity” and “Person_Date” as follows:

Person_Entity(Contact, Attacckerlist, Entities, =, Ts, Te)

Person_Date(Do, Attacckerlist, Dates, =, Ts, Te)

These two functions are used to find the associations between attackers versus entities and attackers versus dates. Finally, we define the composite function “LAssoc” which returns an output list containing the name of attackers and a Boolean value for association. e.g. {Mohamed Atta, Alghamdi, 1}, {Mohamed Atta, Saeed, 0}, ...

LAssoc(Contact, Person_Entity, Person_Date, =, Ts, Te)

The output list from the composite function “LAssoc” can be used as an input for other composite functions or can be saved in XML format for visualization. For instance, the degree of association between entities can be calculated by traversing the output list from the function “LAssoc” and counting the identified association. The weight of an edge in a graph for visualization is determined by the total number of counts from the list with value 1 (i.e. when an association is detected).

V. VISUALIZATION

The results of the link analysis are then used for visualization by the proposed system. A number of visualization options are provided for the users. Based on the data available from [10], visualization of the network of all entitles involved in the September 11 terrorist attack is depicted in Figure 7. These entities include name, data, place, and action. Refined visualizations from Figure 7. are depicted in Figure 8. And Figure 9. In Figure 8. time entity is omitted whereas in Figure 9. action and place entities are omitted. In these figures, links with high degree of association are represented in thick lines. Another example of visualization of links produced by the proposed system for persons and places involved in the Antrax attack in Washington [11] is depicted in Figure 10.

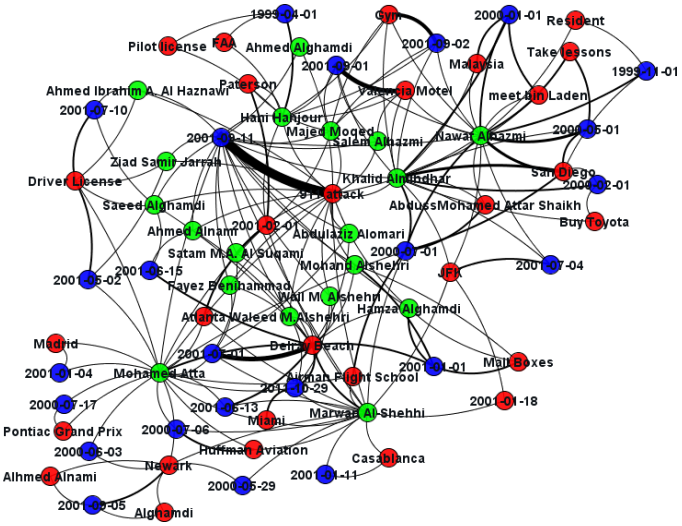

Figure 7. Visualization of the network of all entities involved in September 11 terrorist attack

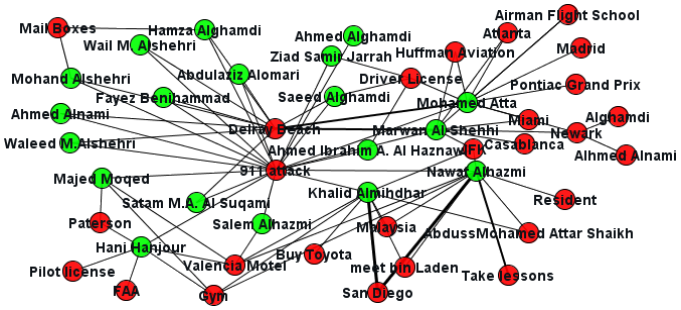

Figure 8. Links between persons and other data entities without time attribute in September 11 terrorist attack association visualization

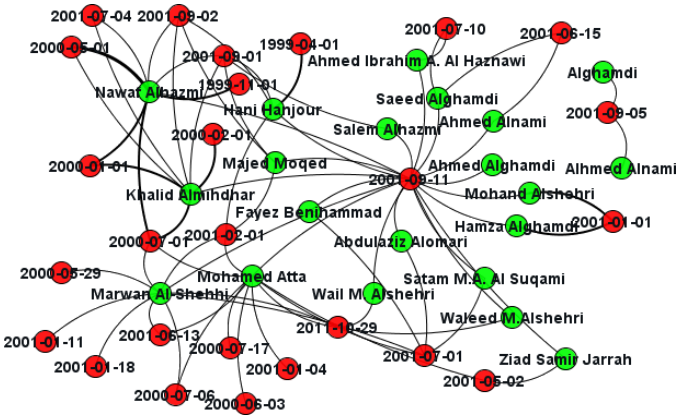

Figure 9. Links between person and time without other attributesin September 11 terrorist attack association visualization

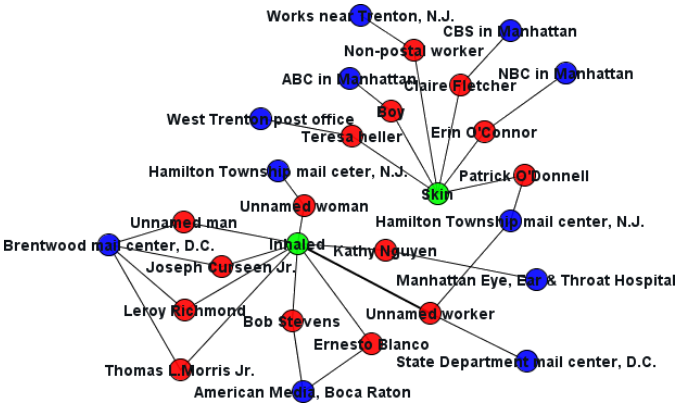

Figure 10. Visualization of linksin Antrax attack in Washington

VI. Related Wrok

A number of event-based systems have been reported in literature. An event is a significant occurrence to present activities at a single point in space-time [5]. Events stored in event-based database usually contain attributes such as occurred time and location related to the activities. For a criminal case, the time of the event and location are considered as the primary attributes [6].

Association analysis tools are often used in criminal investigation to identify the links connecting various events from large datasets. NetMap Analytics [1] is widely used by intelligence agencies around the world for detecting fraudulent activities. NetMap Analytics is capable of linking and revealing invisible connections among entities. NetMap Analytics was used to solve "Backpacker Murders" and the "Mystery TNT Options Trader" cases in Australia. Another well-known analysis and visualization tool for analyzing criminal data and fraudulent activities is Analyst's Notebook [2]. The Analyst's Notebook is capable of acquiring heterogeneous data from different sources. Analyst's Notebook also provides functions for investigators to identify connections, patterns and trends among criminal networks. It also provides functions for easy manipulation of activity data. For instance, users can generate a chart by dragging and dropping entities into the forms. COPLINK project [3] is designed to detect associations among crime entities by integrating multiple data sources from local, regional, and national police departments. The COPLINK prototype has been designed with techniques such as named-entity extraction, deceptive-identity detection, and criminal-network analysis. All the three tools [4] provide functions for visualization of patterns.

VII. Conclusion

In this paper, we describe a layered approach to link analysis and visualization of events. The proposed approach provides different levels of abstraction for aggregating and analyzing events from heterogeneous data sources by using lists and customizable functions. Such capability allows investigators and domain experts to experiment their hypotheses via link analysis and event mining. For future work, we are planning to incorporate name entity extraction functions for mining texts from different data sources. We are also planning to use spatio-temporal databases instead of the current relational databases in our system. A spatio-temporal database [7] is a system which primarily handles both space and time information [8]. These databases are also used in processing information such as criminal records [9], patients' history, and records of traffic accidents etc.

Acknowledgment

This research is funded by the University of Macau.

References

[1] NetMap Analytics, http://www.netmap.com. last accessed on 23 July 2012.

[2] Analyst's Notebook. http://www.i2.co.uk/. last accessed on 23 July 2012.

[3] COPLINK Analytics. http://ai.arizona.edu/research/coplink/. last accessed on 23 July 2012.

[4] R.V.Hauck, H.Atabakhsh, P.Ongvasith, H. Gupta, and H.Chen, " Using Coplink to analyze criminal-justice data," IEEE Computer, 35(3), pp. 30-37, March 2002.

[5] R. Jain, "Experiential computing," Commun. ACM 46(7), pp. 48-55, July 2003.

[6] R.Jain, "Events in heterogeneous environments," In Proceedings of the International Conference on Data Engineering, Bangalore, India, IEEE Computer Society Press, pp. 8-21, 2003.

[7] J.Chen and J.Jiang,"Event-based Spatio-temporal Database Design," International Journal of Geographical. Information Systems, 32(4), pp. 105-109, 1998.

[8] R. Sadri,C.Zaniolo, A.M.Zarkesh, and J.Adibi, "A Sequential Pattern Query Language for Supporting Instant Data Mining for e-Services," In Proceedings of the 27th International Conference on Very Large Data Bases, pp. 653-656, 2001.

[9] S. Schwenke, "Cross-Sector Analysis of Corruption: Summary Report, Sectoral Perspectives on Corruption",November 2002, http://pdf.usaid.gov/pdf_docs/PNACX009.pdf last accessed 23July 2012.

[10] Washington PostNewsweek Interactive, Suspected Hijackers, URL: http://www.washingtonpost.com/wp-srv/nation/graphics/attack/investigation_24.html, last accessed on 23 July 2012.

[11] Washington PostNewsweek Interactive, Confirmed Antrax Cases, URL: http://www.washingtonpost.com/wp-srv/nation/graphics/attack/investigation_43.html, last accessed on 23 July 2012.

[12] JUNG, Java Universal Network/Graph Framework, http://jung.sourceforge.net/, last accessed on 23 July 2012.