JYVÄSKYLÄ STUDIES IN COMPUTING
168

Alexander Semenov

Principles of Social Media Monitoring and Analysis Software

JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 168

Alexander Semenov

Principles of Social Media Monitoring and Analysis Software

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen auditoriossa 3
toukokuun 31. päivänä 2013 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, auditorium 3, on May 31, 2013 at 12 o'clock noon.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2013

Principles of Social Media Monitoring and Analysis Software

JYVÄSKYLÄ STUDIES IN COMPUTING 168

Alexander Semenov

Principles of Social Media Monitoring and Analysis Software

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2013

Editors
Seppo Puuronen
Department. of Computer Science and Information Systems
Pekka Olsbo, Ville Korkiakangas
Publishing Unit, University Library of Jyväskylä

URN:ISBN:978-951-39-5225-9
ISBN 978-951-39-5225-9 (PDF)

ISBN 978-951-39-5224-2 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2013

ABSTRACT

Semenov, Alexander
Principles of Social Media Monitoring and Analysis Software
Jyväskylä: University of Jyväskylä, 2013, 92 p. (+ included articles)
(Jyväskylä Studies in Computing
ISSN 1456-5390; 168)
ISBN 978-951-39-5224-2 (nid.)
ISBN 978-951-39-5225-9 (PDF)

This thesis studies how automatic tools can be used in the monitoring and analysis of the digitally-encoded information that exists on the Internet and, more specifically, on social media sites. Social media sites already have over a billion users. In our view, the content existing within social media sites is modelling parts of the social reality, with certain limitations. For instance, not all people have access to the Internet or are represented in social media. The perception of relationships on social media sites differs from the perception of those in the immediate social reality. Also, fake virtual identities are an important aspect. Yet, there are many interesting phenomena in social reality, information about which can be gathered from social media much faster than in real life. The main objective of this thesis is to find out how automatic tools can be used to collect and analyse the data from social media sites and what their limits are. Conceptual-analytical and constructive research approaches are used to solve these problems. This thesis proposes a general three-layer modelling framework for modelling a social media environment. The first level is the social reality, as modelled by social media sites. It is captured by site ontologies. The second level is a multirelational graph structure that can be used to model any social network and its development over time at any known social media site. The third level is the persistent repository for the instances of this graph structure. The thesis presents the requirements and architecture for the social media monitoring software system that implements this framework. The software gathers data from social media sites relying on the site ontologies. These heterogeneous 1st level models are stored at the monitoring site and transformed to the 2nd level homogeneous multirelational graph structure. The 2nd level is implemented as a repository, the schema of which depends on the technology in use; but the interface is homogeneous, relying on the above graph. The thesis also establishes fundamental limits for the remote monitoring and analysis activities that are novel in the context of social media monitoring. The developed software has been used to collect real data from school-shooter networks, the entire contents of the LiveJournal.com site, and from sites in the Finnish segment of the Tor network. The software system can benefit a number of users, such as cyber police or business intelligence companies.

Keywords: social media analysis, social network analysis, temporal database, targeted crawler, multirelational graph

Author's address
Alexander Semenov
Department of Computer Science and Information Systems
University of Jyväskylä
P.O. Box 35
40014 Jyväskylä, Finland
alexander.v.semenov@jyu.fi

Supervisor
Professor, Dr.-Ing., Jari Veijalainen
Department of Computer Science and Information Systems
University of Jyväskylä
P.O. Box35
40014 Jyväskylä, Finland

Reviewers
Professor, Ph.D., Tore Risch
Computing Science Division,
Department of Information Technology,
Uppsala University, P.O.B. 337
SE-751 05 Uppsala
Sweden

Professor, Ph.D., Markku Oivo
Tietojenkäsittelytieteiden laitos,
PL 3000;
90014 Oulun yliopisto
Finland

Opponent
Assistant Professor, Ph.D., Mykola Pechenizkiy
Department of Computer Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven
The Netherlands

ACKNOWLEDGEMENTS

I want to express special gratitude to my supervisor, Professor Jari Veijalainen, for his invaluable support during the research and preparation of this thesis.

Also, I would like to thank the COMAS graduate school for financial support of my research. I extend my thanks to the coordinator of COMAS graduate school, Sami Kollanus.

I would like to thank Professor Andrei Rybin from NRU ITMO, Saint Petersburg, Russia for his help at the initial stages of this research. In addition, I would like to thank Professor Alexander Boukhanovsky from NRU ITMO for his supervision of my work in the beginning. I express my gratitude to Jorma Kyppö for collaborating and coauthoring one paper. Also, I would like to thank Professor Seppo Puuronen for his comments during preparation of the thesis.

Moreover, I express my gratitude to the reviewers, Professor Markku Oivo, and Professor Tore Risch for their constructive feedback. I wish to express my gratitude to the opponent, Assistant Professor Mykola Pechenizkiy.

Also, I am very thankful to the Department of Computer Science and Information Systems, and Department of Mathematical Information Technology and wonderful people who work there. I give special thanks to Jouko Nieminen, and Tuukka Puranen. Also, I am very grateful to Professor Pasi Tyrväinen, and Dr. Oleksiy Mazhelis for their advices. In addition, I would like to acknowledge my colleagues, doctoral students Sergey Chernov, Pavel Gonchukov, Alexandr Maslov, Oskari Miettinen, Mazhar Sadiq, and Almamy Touray.

I wish to thank my family and friends for their support.

Jyväskylä, 14.05.2013
Alexander Semenov

FIGURES

TABLES

CONTENTS

LIST OF ORIGINAL ARTICLES

I. Veijalainen, J., Semenov, A., Kyppö, J., 2010. Tracing potential school shooters in the digital sphere. In Bandyopadhyay, S.K., Adi, W., Kim, T., Xiao, Y. (Eds.), 4th International Conference, ISA 2010, Miyazaki, Japan, June 2010, Proceedings, Communications in Computer and Information Science. Berlin Heidelberg: Springer, pp. 163–178. DOI: 10.1007/978-3-642-13365-7_16

II. Semenov, A., Veijalainen, J., Kyppö, J (2010). Analysing the presence of school-shooting related communities at social media sites. International Journal of Multimedia Intelligence and Security (IJMIS), 1 (3), 232-268

III. Semenov, A., Veijalainen, J., Boukhanovsky, A., 2011. A Generic Architecture for a Social Network Monitoring and Analysis System. In Barolli, L., Xhafa, F., Takizawa, M. (Eds.), The 14th International Conference on Network-Based Information Systems. Los Alamitos, CA, USA: IEEE Computer Society, pp. 178–185., DOI: 10.1109/NBiS.2011.52

IV. Semenov, A., Veijalainen, J., 2012. Ontology-guided social media analysis System architecture. In A. Maciaszek, L., Cuzzocrea, A., Cordeiro, J. (Eds.), ICEIS 2012 - Proceedings of the 14th International Conference on Enterprise Information Systems, Volume 2. Portugal: SciTePress, pp. 335–341., DOI: 10.5220/0004157303350341

V. Semenov, A., Veijalainen, J., 2012. A Repository for Multirelational Dynamic Networks. In Karampelas, P., Rokne, J. (Eds.), 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 1002–1005., DOI 10.1109/ASONAM.2012.174

VI. Semenov A., Veijalainen J. (2012). A modeling framework for social media monitoring, accepted for publication at International Journal of Web Engineering and Technology (IJWET), 36 p.

VII. Semenov A. (2013). Analysis of services in Tor network. Accepted to 12th European Conference on Information Warfare and Security ECIW-2013, Jyväskylä, Finland. 11 pages.

LIST OF TERMS WITH DEFINITIONS

Conceptualization is “an abstract, simplified view of the world that we wish to represent for some purpose” (Gruber, 1993, p.199)

Crawler is the software program that traverses the World Wide Web (WWW or web) information space by following hypertext links and retrieving web documents by standard HTTP.

Crawling ontology is the ontology, having a description of the current crawling task along with the specification of relations, which should be considered sources for new entities for traversal.

Focused crawler is a crawler that seeks, acquires, indexes, and maintains pages on a specific set of topics that represent a narrow segment of the web.

Graph database is a database used to store graph data. Graph structures with nodes, edges and attributes are used to represent the data.

Immediate social reality is the part of social reality which excludes virtual communities.

Monitoring and analysis software is piece of software that is deployed at the monitoring site; it collects data from the monitored site over the Internet, stores it at the repository, and analyses the collected data. Monitoring software observes the site in the same manner as users do.

Online social network is the digital representation of the virtual community at a social media site, adhering to the site ontology. A virtual community is a group of real individuals and an online social network is a group of users’ accounts, profiles, etc. and their relationships on the site.

Ontology is an explicit specification of a conceptualization (Gruber, 2009). A conceptualization should express a shared view between several parties (Staab and Studer, 2011). Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly.

Privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others.

Private data are data that may be accessed by the owner only (or based on special access rights granted by the owner).

Public data are data that may be accessed by everyone who desires.

Relational database is a database created using a relational model (a set of tables corresponding to a relational scheme). A relational database is managed by a relational database management system (RDBMS).

Repository a persistent data collection consisting of heterogeneous multimedia data that are fetched from diverse social media sites for the purpose of analysis.

School shooting is an event in which: (a) a student or a former student brings a gun, sword or similar weapon, or explosive/flammable liquid to school with the intent to kill somebody; (b) the gun is discharged or the weapon, liquid, or explosive is employed and at least one person is injured; and (c) the perpetrator attempts to shoot or otherwise kill more than one person, at least one of whom is not specifically targeted.

Semipublic data are data that may be accessed by small groups of individuals who must be authenticated and authorized.

Site ontology is a model that captures the essential relationships observable by the users at a social media site from monitoring and analysis point of view.

Social media is a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0 and that allow the creation and exchange of user-generated content (Kaplan and Haenlein, 2010).

Social media analysis is the analysis of information stored at social media sites.

Social media site is a concrete instance of social media, i.e., a web site with a certain URL and features typical of social media. Examples include facebook.com, vkontakte.ru, and twitter.com.

Social network is a community of people in real life (real people and their social relationships). Wasserman and Faust (1994, p. 20) write, "A social network consists of a finite set or sets of actors and the relation or relations defined on them. The presence of relational information is a critical and defining feature of a social network". There, actors are social entities. Social networks are part of the immediate social reality.

Social reality is the reality created through social interaction within communities of people, thus consisting of accepted social norms and things known through people's experiences at a given time.

Software architecture is a set of structures needed to reason about the software system. *Software architecture* comprises software elements, the relations between them, and the properties of both elements and relations (Clements et al., 2010).

Spiral method is a software development method that combines the elements of both the design and prototype phases of software development (Boehm, 1986).

Temporal database is a database with built-in time aspects.

Temporal multirelational graph is graph, consisting of labelled nodes and edges, in which two nodes can be connected by more than one edge and in which nodes and edges can have attributes.

Three-layer modelling framework for social media is a hierarchy of models in which an immediate social reality is modelled by the 1st level models at social media sites; all known social media sites are modelled by a 2nd level model, a temporal multirelational graph; and the temporal multirelational graph is modelled by 3rd level models: database schema, and an instance implementing the temporal multirelational graph.

Universe of discourse is the part of the real world that is modelled by an information system.

User-generated content is the content of web sites produced by users of these sites within boundaries set by the site.

Virtual community is a social network of individuals who interact through specific social media site, potentially crossing geographical and political boundaries in order to pursue mutual interests or goals. The site maintains a digital representation of the community, adhering to the site ontology.

Wrapper is a program that extracts the content of a particular information source and translates it into a relational form or graph instances.

1 INTRODUCTION

The Internet, a global computer network, has emerged during the last 40 years. Its predecessor, Arpanet, was mainly aimed at various scientific and military institutions. The Internet was born in 1982 when the TCP/IP protocol suite was standardised by the US Department of Defense. The Internet is a global open network of networks relying on Internet protocol (IPv4 and now IPv6). To present, it has evolved a lot; and currently, many people use the Internet only for entertainment.

One of the latest trends in the modern Internet is social media—a group of Internet-based applications built on the ideological and technological foundations of Web 2.0 and that allow the creation and exchange of user-generated content (UGC) (Kaplan et al., 2010). Social media act as new types of communication media.

The virtual identity of an individual and the relations between these identities are the main building blocks for social media (Kietzmann, Hermkens, McCarthy, and Silvestre, 2011). Real identities are represented in social media by virtual identities (e.g., by profiles or avatars). There are many popular social media sites, including Facebook, which, in October 2012, announced that it had one billon users (Facebook Newsroom, 2012); Twitter, with about 500M users (Semiocast — Twitter reaches half a billion accounts, 2013); and YouTube, having a number of users in the same range.

Social media sites provide their users the platform, but the majority of the content is generated by the users themselves. Thus, the key feature of social media sites is UGC. For example, users upload various videos to YouTube, comment on them, like or dislike them, and create the channels and subscribe to them. Users post "tweets" expressing their opinions in Twitter. In Facebook, users make friends and upload pictures and various profile details about their lives.

Social media facilitate virtual communities. The term *virtual community* was coined by Howard Rheingold. "Virtual communities are social aggregations that emerge from the loosely connected computer network when enough people carry on those public discussions long enough, with sufficient human feeling, to form webs of personal relationships in cyberspace" (Rheingold, 1993, p.5). Rheingold's view was tied to the technology of his time 20 years ago. In the current situation the cyberspace manifests itself as social media and many currently encountered virtual communities are *online social networks* at various social media sites.

In addition to the general public, social media sites also attract political leaders who try to influence public opinion through them, or various commercial brands that try to expand their market shares using social media.

Also, various types of criminals use social media to spread their ideas. Examples include Anders Behring Breivik, who shot and killed 69 people in Utoya island in Norway, July 2011. He uploaded his 1500-page manifesto to the Internet and posted a YouTube video with the outline of his ideas shortly before his crimes. School shooter Pekka-Eric Auvinen, who shot and killed eight people in Jokela, Finland on November 7, 2007, had a YouTube account with which he had been posting various hate videos. Shortly before the attack, he had uploaded his detailed plan to the World Wide Web (WWW or web) site rapidshare.de and put a link to the plan on his YouTube page. Another shooter, Matti Saari, killed 10 people on September 23, 2008, and had a similar pattern of online behaviour. He had a YouTube account with which he was posting videos of him posing with a gun and various hate content. After the shooting sprees, the above-mentioned perpetrators committed suicide. Later, parts of the information they published online were removed by administrators of the social media sites, but some parts still were retained in various forms on the Internet (Semenov, Veijalainen, and Kyppö, 2010).

Social media also were used extensively to organize protest groups in the so-called Arab Spring (Gilad Lotan, Erhardt Graeff, Mike Ananny, Devin Gaffney, Ian Pearce, and danah boyd, 2011). Similarly, radical postings on Internet forums facilitated disruptive mob behaviour during riots in Northwest China in July 2009 (Yang and Chen, 2012).

The goal of this thesis is to understand the possibilities and limits of a software system that monitors and analyses the immediate and modelled social realities as they are remotely observable at various social media sites.

1.1 Motivation for the research

Collecting analog data from large sets of real people is technically difficult and could require some sorts of surveying or focus groups, which can be very time consuming and expensive since such methods involve large amounts of communications with various individuals. Social media sites allow people to carry out discussions and maintain relations in the digital sphere. They may contain various people's opinions, rumours, or leaks. Data existing on social media sites already are digitally encoded, and so gathering such data would require fewer resources than gathering analogue data. Also, analysing the data generated by large sets of users of social media sites is becoming easier. However, gathering and analysing such data requires special software tools.

Nowadays, there is much interest in monitoring and analysing data from social media sites (Federal Business Opportunities, 2012, IARPA, 2011; Helbing and Balietti, 2011). As such, there is a related concept called *open source intelligence*, which is defined as data "produced from publicly available information that is collected, exploited, and disseminated in a timely manner to an appropriate audience for the purpose of addressing a specific intelligence requirement" (National Defense Authorization Act, 2006). There are several commercial systems to analyse social media (Social media monitoring tools comparison guide, 2013); but there is a lack of similar products developed in academic settings.

Herewith, collection of longitudinal data is important because content of social media evolves, e.g., some parts of it may be removed by users or administrators of the sites. There are ontology evolution and content evolution observable. The former happens when new concepts are added to or removed from the site's social world model. E.g., new relationships or attributes for the user profile may be added or removed over time. This also may be presented as "versioning". For example, in December 2011, Facebook asked its users to move to a new kind of profile named "timeline". We refer to this as *ontology evolution*; this aspect is not handled in this thesis. Usually, only the latest version of the content is presented to the ordinary user through a web browser. For this reason, if one wants to monitor the evolution of the networks, it is necessary to continuously monitor social media sites.

One may argue that content presented on social media sites models the immediate social reality. Yet, nowadays, the world population is approximately 7 billion; and the population is constantly changing since new people are being born and old people are dying every second. Currently, the birth rate is about 134 million per year and the death rate is 56 million per year. Thus, about 4.25 children are born per second (367 000 children per day) and 1.78 persons die per second (153 000 persons per day) (World Population Prospects, the 2010 Revision, 2010). With such a large and varied population, and according to the International Telecommunication Union (Estimated internet users, 2012), there are only an average of 30.23 Internet users per 100 persons. Moreover, not every Internet user is represented in social media sites. Therefore, social media do not capture all of humankind, and that should be considered during the monitoring of the social media.

Next, virtually modelled relationships usually do not correspond to real relationships, although they frequently have the same names. For instance, friends on social media sites may be perceived differently than friends in immediate reality. Thus, sites may allow a much greater number of virtual friends than a person may have in real life. For example, Facebook allows 5000 friends; however, it is not possible to have so many real friends. Another important aspect is the possibility to maintain rather easily a fake virtual identity or steal the virtual identity of another person (identity theft). There are a number of forms of fake identities: an identity of non-existing person, identity of a dead person, or fake identity of a living person like a celebrity. Identity theft is a serious problem and should be analysed properly. Facebook estimates that approximately 4.8% of accounts are duplicates (Facebook - Quarterly Report, 2013). These issues must be considered when the relevance of the data gathered from the social media sites for the immediate social reality are estimated.

Each social media site maintains its own boundaries for UGC. Data existing there have structures based on concepts presented there. Structure is reflected by the design of the sites and data fields that can be filled in with UGC through provided interfaces. Later, entered content can be modified. Relations and other actions at the sites also take place within a defined structure. For instance, Facebook articulates dozens of concepts, including the profile, friend, timeline, photo album, photo, status update, comment, group, like, and public page. In Twitter, there are tweets, user profiles, and followers; and in YouTube, there are channels, subscribers, and videos.

Thus, sites maintain certain social world models, and aspects of reality are captured according to these models. Social world models vary from site to site. An important aspect of social media monitoring and analysis software is the possibility to collect, store, and analyse the data from different social media sites. Since a monitoring system is not integrated with the monitored social media site, the system must remotely access the site to be monitored. Therefore, it must contain a crawler, which accesses sites and transfers data from the site to the repository. For wide monitoring, the crawler should be capable of accessing all known social media sites.

There are differences among the social world models, represented by the sites, and representing them in HyperText Markup Language (HTML) or in other markup languages. This problem can be addressed using ontologies, which would be instantiated by the crawler with parsing technologies such as XML Path Language (XPath) or regular expressions. For monitoring and analysis purposes, we assume that each social media site is captured by the *site ontology* of a particular site. The site ontology is a model that captures the essential relationships from a monitoring and analysis point of view; it is not necessarily a complete representation of the relationships and contents formed by the monitoring site administrator and stored on the monitoring site. For instance, Facebook currently offers dozens of concepts; but for a certain modelling purpose only, (part of the) profile, friend, status updates, and comments might be sufficient.

Instantiation of site ontology with data from the site is the online social network, which is the digital representation of the virtual community existing at a social media site. In this work, the site ontology is primarily based on the concepts offered to an ordinary user who uses browser to access the site. The site application programming interface (API) might offer a subset of these concepts.

Thus, construction of a generalised model of social media sites could facilitate generalisation of the software and, possibility, of its use for many different sites. Naturally, social media sites contain only a finite number of concepts, which can be characterized by a finite number of attributes. By extension, there can be only a finite set of atomic entities, or binary relations.

All social media sites support establishing relationships between individuals. Then, sites having different site ontologies can be modelled by graphs. Under these circumstances, a homogenous model that generalises all possible heterogeneous graphs modelling online social networks would allow a convenient management of the data, gathered from different social media sites. Modelling online social networks represented at social media sites with graphs would allow the convenient application of graph analysis algorithms to their data for calculation of some new properties.

Establishing a common model for all social media sites would allow the application of common analysis methods and the construction of a data repository with a common interface. The repository might be implemented in different ways. Such differences can facilitate the construction of a three-layer modelling environment with which the immediate social reality is modelled by the contents of social media sites. Then, social media sites are modelled by a special type of graph. The next level of the model is the implementation of this graph in a database management system (DBMS), which is discussed in section 3.4.

The modelling framework presented in this thesis provides the basis for the development and evaluation of a generic social media monitoring software. Such software system may be used in a number of contexts.

1.2 Objectives

The main objective of this thesis has been to develop a social media monitoring and analysis software system. Therefore, the requirements should be elicited. For this, a general framework that could act as the theoretical underpinning for the software should be developed. The above framework would help in understanding how the modelling must be done and which parts the software must handle. This would lead to elaboration of software architecture and its parts. The system should make use of ontologies for translating the data from social media sites to the repository.

The framework would facilitate the construction of a repository for homogenously managing heterogeneous longitudinal data, ideally, from all known social media sites. The framework also would help analyse operations that can be used to manipulate model instances and, together with the architecture of the system, address theoretical limits of the longitudinal social media monitoring.

Since social media sites continuously evolve, in order to capture all the changes it is necessary to collect data faster than it is changing on the sites. Thus, a data transfer channel should have enough capacity to make this possible. Also, the monitoring software should have enough processing capacity and fast enough repository access to store the data retrieved. Finally, the social media site should allow its polling with a high enough rate, or a streaming API should push the new data with a high enough rate.

The described performance issues can be addressed by calculation parameters of the channel existing between social media sites and the monitoring software. Thus, data transfer channel capacity and its latency should be analysed. Also, the data repository and performance of the used analysis algorithms can be addressed.

In addition, one of the objectives is to study the online behaviour of school shooters and elaborate a probabilistic model that attempts to capture their behaviour and that shows the limits of the monitoring.

A further objective is to use the developed software to collect and qualitatively and quantitatively analyse the data from several social media sites.

In short, there were following objectives:

1. Develop a social media monitoring and analysis software system.
    a. Develop the requirements.
    b. Develop the architecture.
2. Develop theoretical framework which could act as the theoretical underpinning for the monitoring and analysis software.
3. Assess the performance limits of the monitoring and analysis software and evaluate its effectiveness.
4. Validate the software via using it for the collection and analysis of the data from several social media sites.
5. Study the online behaviour of school shooters.

1.3 Summary of the results

To sum up, the following results were achieved:

1. A general three-layer modelling framework that acts as a theoretical basis for social media monitoring and analysis systems.
2. Architecture of a generic social media monitoring system capable of capturing longitudinal development of online social networks.
3. Analytically established transmission capacity and processing-rate limits for any remote monitoring activity over the Internet.
4. An early version of the system was used to collect real school-shooter communities, a collection of nine million nodes (user profiles) from Livejounal.com. A later version was used to collect and analyse hidden services in the Finnish segment of the Tor network. Services such as the anonymous imageboard Thorlauta and the forum Suojeluskunta also were collected and quantitatively analysed.
5. Qualitative analysis, which suggests that 7 out of 11 analysed perpetrators exposed their intentions on the web; thus, the attacks

could have been prevented. This indicates that exposure of digital information in the public sphere is a necessary condition for certain kinds of monitoring activities, which could have been carried out using the software. Also, this supports the claim that monitoring and analysis software potentially has the necessary public data to monitor possible perpetrators if other conditions are met.

6. A probabilistic model that attempts to estimate how probable it is to detain a potential school shooter before he or she performs an attack. Based on this, one can evaluate the effectiveness of any automatic monitoring system if one wants to capture a perpetrator before he or she performs an attack. One also can evaluate the time frame during which the attack can still be averted.

2 RESEARCH PROBLEM AND METHODOLOGY

This chapter describes the research problem, research methods, and methodologies.

2.1 Research problem

One of the latest trends in the Internet is the development of social media. Nowadays, most popular social media sites have hundreds of millions of users. Social media sites allow people to create virtual identities and maintain virtual relationships with other virtual identities. Social media sites allow generation of content adhering to the sites' social world models.

To some extent, social media sites represent the immediate social reality. They provide the interfaces that facilitate the generation of UGC adhering to the site world model. The site social world model usually adopts some concepts from the immediate social reality. However, virtual relationships within social media may not accurately correspond to relationships from real life, although they frequently have the same names. Also, not all important social relationships are presented on social media sites, such as when A hates B. Next, not all people are users of the Internet and, obviously, not all of them are represented in social media. Further, the creation of fake virtual identities, duplicate identities, and identity theft should be considered. All this implies that social media sites represent the immediate social reality with certain limitations, which should be taken into consideration before developing software that aims to analyse the immediate social reality. Still, social media contain a lot of UGC.

Also, content existing within social media sites is continuously evolving. Users join and leave the sites, establish new connections, and upload, modify, and delete content. Usually only the latest version of the content is presented at the site. However, the availability of historical data could help in many analysis tasks. Such data may be collected by continuous longitudinal monitoring of the social media sites. Then, **research question I** arises:

Under which conditions is it possible to construct an effective and efficient social media monitoring and analysis system for longitudinal monitoring, and what kind of architecture it should have?

Ontologies may be used for capturing social world models of the social media sites. Thus, content at the social media sites is bounded by site ontologies that are heterogeneous and vary from site to site. Examples of entities from the site ontology of Facebook are user profiles and the friendship relations between them. In Twitter, there are user profiles and follower relationships; and there are channels and subscribers on YouTube.

The presence of a homogenous model, which could generalise these evolving heterogeneous site ontologies, would allow the application of the same software system to monitor and analyse different social media sites. This system would implement the developed model. Also, the design of a data repository with a generic interface could follow from the model. Beneficial for the analysis of the data would be to apply known graph algorithms to the collected data. Therefore, it should be possible to represent the collected data as a graph. Then, **research question II** arises:

Is there a homogeneous graph model that could be used to model all or a majority of online social networks at known social media sites?

A positive answer to this question would require the specification of a graph model that has a high enough expressive power. This model would form a homogeneous layer above the semantically heterogeneous site ontologies.

This immediately raises the question, how would the instances of such a homogeneous model (i.e., concrete graphs) be stored in a data repository. The second level of the model is essential as a homogeneous “pivot” model. There is an abstract schema that supports all its instances. The repository would have a homogenous interface based on graph structure.

Research question III would be:

What are the software requirements and the software architecture that implement the above three-level framework in an efficient way? What modules are there? What kind of relationships should they have?

The architecture of the system is modular. The main modules of the system are the *crawler module*, *repository module*, and *analyser module*. The crawler would be able to fetch the data from the sites. The procedures to extract information from HTML content or from the API of social media sites would be described in ontologies, which will guide the collecting process. Subsequently, the collected data can be put to the repository with a homogenous interface that supports the multirelational graph interface.

It is important to notice that social media sites constantly evolve. Thus, in order to capture the entire site evolution process, monitoring software should be able to gather data at least as fast as changes occur. Then, **research question IV** arises:

What are the fundamental limits of remote monitoring and analysis activity, and which properties of the environment affect them?

The answer to this question requires an analysis of the maximal rates with which a social media site allows polling of its state or sends streams through a streaming API, inspection of the communication channel capacity between the monitoring and monitored sites, and monitoring site performance characteristics.

Apart from the above issues, an inherent limiting factor for public monitoring is that not all data can be monitored, for privacy and other reasons. This requires the categorization of data into private, semipublic, and public.

Next, from a practical point of view, it is important to answer **research question V**:

When does the information indicating that some event in physical reality is about to happen appear in the public sphere on social media sites, and how long does it persist? Could this information have been captured by monitoring software and the events prevented?

This question can be answered by analysing earlier information that existed on social media before the occurrence of events that were not prevented. Examples of such events are some school shootings and Breivik's attack on July 22, 2011. The main idea was that, if major perpetrators behave like earlier ones did, then the monitoring software could be used to prevent attacks. To answer this question, analysing online traces left by major school shooters during 2005–2011 was conducted. Seven out of 11 major perpetrators left clear traces that were still partially accessible in 2011 when we studied the issue.

2.2 Research methods

Methods of conducting scientific research have been studied quite extensively. There are a number of research approaches that can address different research questions and suit different research domains. Some research domains are very broad and can use many methods, while others allow only specific methods.

The research questions presented above are related to the construction of the software and related models as well as to the analysis of the properties of the environment. The construction of the software and related models can be answered with help of constructional methods. The analysis of the properties of the environment can be answered with methods that are aimed at detailed studying of the phenomena. Below, descriptions of the used research methods are presented.

Nunamaker, Chen, and Purdin (1991) define *research methodology* as the combination of the process, methods, and tools that are used in conducting research in a research domain. Nunamaker et al. (1991, p. 93) suggest that a major question in computer research is "what can be automated and how can it be done efficiently and effectively"; they suggest system development as a research method. According to Nunamaker et al. (1991), system development consists of five stages: concept design, constructing the architecture of the system, prototyping, product development, and technology transfer.

Hevner, March, Park, and Ram (2004) describe two paradigms of the research conducted in information systems: behavioural science and design science. The former seeks to develop and verify the theories that could verify or predict human or organizational behaviour, and the latter aims to create new artefacts. Hevner et al. (2004, p. 3) define such artefacts as constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems).

According to Järvinen (2004), the motivation behind building a new innovation could be either the low quality of similar existing innovations or the lack of that innovation at all. Building a new innovation involves constructing the functioning artefact, which changes from the initial state to a goal state during the construction process.

Järvinen (2004) notices that the goal state in a construction task may not necessarily be known. In this case, two alternatives are proposed: the first is to specify the goal state and implement measures to achieve this state; the second is to realize implementation and goal-seeking in parallel. Specification of the goal state process might involve communication between several parties of stakeholders (e.g., users and researchers) who should, in the end, come to the description of the desired solution space; or a researcher might develop the first prototype herself and show it to the users in order to facilitate the negotiation process.

However, in cases when the artefact never existed, sequential specification and implementation might be difficult since it can be hard to imagine a thing that never existed. In this case, a parallel specification and implementation process take place. Some outlines of the goal state are presented in the beginning, then an artefact achieving this state is being implemented, and then an updated goal state appears. In software engineering, research specification of a goal state may be referred to as *requirements specification*.

There are many software engineering models: e.g., the waterfall model, where development flows "downwards" as a sequential process from specification to implementation and deployment (Benington, 1983); the iterative model, where development progresses through steps of iterations consisting of prototyping, testing, and analysing. Boehm (1986) introduced the spiral development approach, which combines both elements of the design and prototyping phases (see Figure 1). Spiral approach is based on looped refinement of the goal. In spiral approach, elements of the product may be added when they become known. Floyd, Reisin, and Schmidt (1989) propose a STEPS software engineering approach (software technology for evolutionary and participative system development): they recommend building a version of the software, giving it to users for experimental use, during which the software would be maintained, and then a new version is realized. The agile development method, a key idea of which is adaptive planning, was introduced in 2001 (Manifesto for Agile Software Development, 2013).

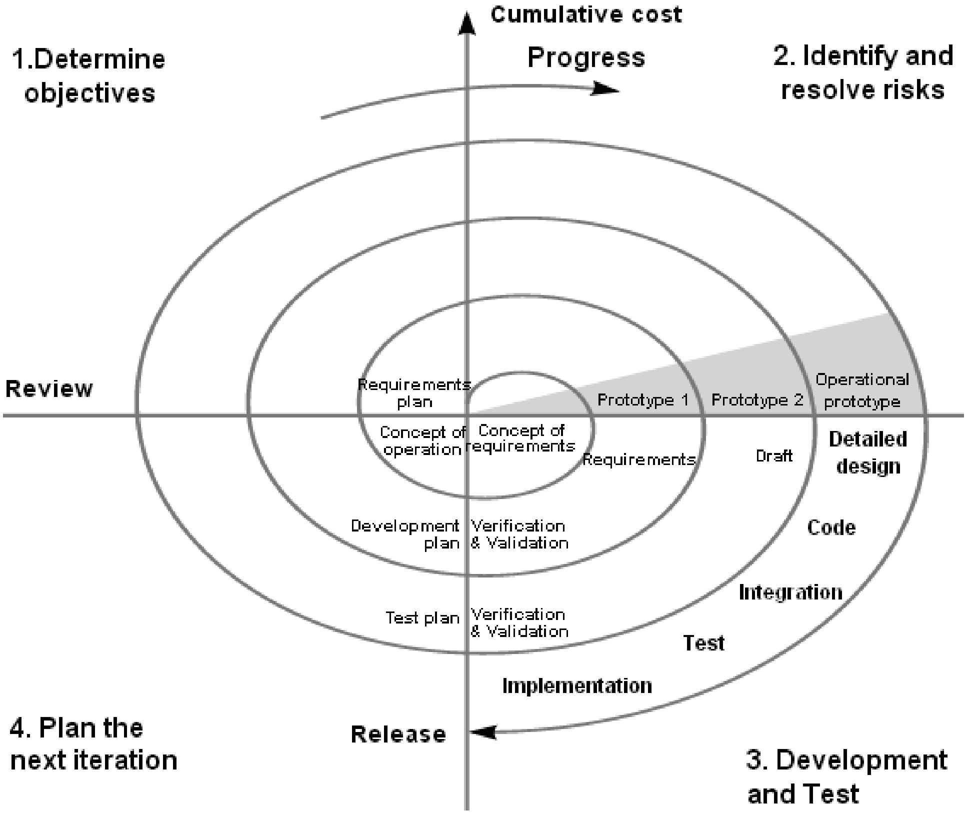


FIGURE 1 Spiral model (Boehm, 1986, p.25)

As noticed above, when a problem is new, an exact description of the goal state might be difficult. Paré (2004) describes the case-study research method and advises its use to study phenomena, which are broad and complex, (1) when the existing body of knowledge is insufficient to permit posing casual questions, (2) when in-depth investigation is needed, or (3) when a phenomenon cannot be studied outside the context in which it occurs. According to Paré (2004), a case study should consist of a following phases:

Design of a case study;
Conduct of the case study;

Analysis of the case-study evidence; and
Writing the case-study report.

Design of a case study consists of such phases:

Initial definition of research questions;
A priori specification of constructs or theory;
Definition of the unit of analysis;
Selection and number of cases; and
Use of a case-study protocol.

During the case study, data about the phenomenon should be collected. There are the following sources of data (Yin, 2002): documentation, archival records, interviews, direct observation, participant observation, and physical artefacts. Ameripour, Nicholson, and Newman (2010) describe online ethnography, or nethnography, as one method to study communities and cultures created through computer-mediated communication.

In the present thesis, case studies are used to analyse the school shooters data.

2.3 Research design

The constructive research approach is the main approach used through the thesis. Software solutions are used to solve particular problems. Problems exist within some *world* (Lamsweerde, 2009). Software solutions are aimed at improving the world by developing some *machine*. Figure 2 depicts world phenomena and machine phenomena.

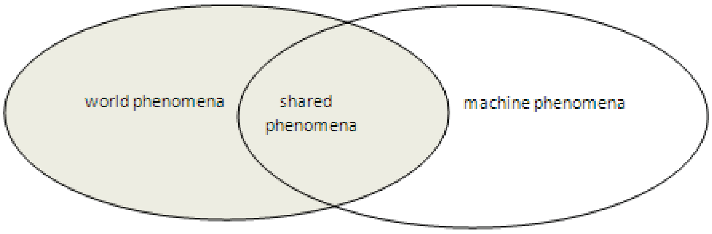


FIGURE 2 World and machine phenomena

In a machine-building project, two versions of a system may be considered: *system-as-is*, the system as it exists before the machine is built into it, and *system-to-be*, the system as it should be when the machine is built and integrated (Lamsweerde, 2009). Then, requirements engineering is a coordinated set of activities for exploring, evaluating, documenting, consolidating, revising, and adapting the objectives, capabilities, qualities, constraints, and assumptions that the system-to-be should meet, based on problems raised by the system-as-is and opportunities provided by new technologies.

In this thesis, it is assumed that the world is the immediate social reality, as modelled by social media sites, and the machine is the developed social media monitoring and analysis system. Since architectures of generic social media monitoring and analysis systems did not exist prior to the research, it was decided to specify the goal state and implement the artefact in parallel (spiral method). Such methods as conceptual analysis and case studies using an online ethnography data collection method were used to study the world phenomena more deeply and identify the system-as-is, its context, and problems existing there. Then requirements for the system-to-be were created and gradually enhanced.

From a software engineering point of view, the research design may be viewed as a spiral (Figure 3). At first, the problem world was studied, then the requirements were elicited, then the prototype was developed and enhanced. Below is a detailed description of the research stages.

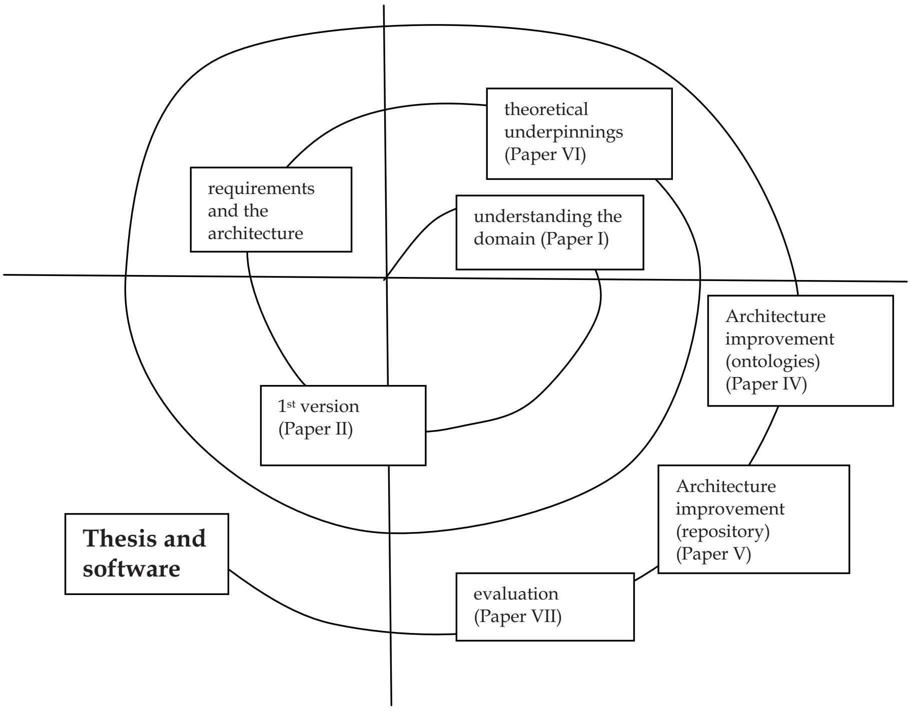

theoretical underpinnings (Paper VI)
requirements and the architecture
understanding the domain (Paper I)
Architecture improvement (ontologies) (Paper IV)
1st version (Paper II)
Architecture improvement (repository) (Paper V)
Thesis and software
evaluation (Paper VII)

FIGURE 3 Spiral of research phases

2.3.1 Theory creating research

Conceptual-analytical research methods were used in the beginning of the research to study the problem world and understand the system-as-is. Thus, **research question V** was answered. This part of the research aims to study school-shooting phenomena more deeply. Also, it studies how long the data persist and whether the major perpetrators could have been detected by monitoring software.

The case study was selected as the research method. Analysis of the information, leaked online from 11 of the most prominent school shooters from 2005 to 2010, was conducted. Although original information emitted by school shooters frequently was removed from social media sites, it still exists in the caches of search engines or is re-uploaded by people to other servers (in the form of saved HTML pages or screenshots of all or part of pages). All found data were analysed.

The result of this phase was the conclusion that 8 out of 11 school shooters left traces in the digital sphere, and 7 of them were active on social media sites. Information was studied and qualitatively analysed—ontology, characterizing online behaviour of potential school shooters, and signs emitted by them were constructed. The analysis is conducted in Article II.

Also analysed were the probability of preventing the school shooting and its dependence on leakages of information from the potential school shooter.

The result of this phase is the probabilistic model presented in Article I, which describes the probability of preventing school shootings. The model tries to measure the probability with which school shooters could be captured before they commit their crimes, using the monitoring and analysis software. In addition, false positive and false negative probabilities were defined. Also, the development path model of a person ending up in a school shooting was proposed.

The answer to the above questions identified the system-as-is and the problems existing there; namely, the possibility of preventing school shootings by analysing gathered longitudinal data. The above can be understood as requirement elicitation.

2.3.2 Conceptual analytical research

Conceptual-analytical research methods were used further to analyse the possibilities of constructing a social media monitoring and analysis system. This part of the research answered **research question I**.

This phase of research analyses the immediate social reality and its relation to social media content. In addition, this phase introduces the system of definitions used throughout the research. Definitions of terms such as *social media*, *social network*, and *online social network* were adopted. Also, a three-layer modelling framework is introduced as the theoretical underpinning for the system-to-be, a generic social media monitoring and analysis system. Implementation of the software supporting the framework would allow monitoring and analysis of different social media sites with one software system.

In this phase, a set of social media sites was conceptually analysed. The result was the claim that content in social media models some part of the real world, adhering to site ontologies adopted by the sites. Site ontologies vary between the sites and describe how the sites model reality. Next, four site ontologies were reviewed, and commonalities were found and analysed. As a result, a temporal multirelational graph (a graph consisting of labelled nodes and edges, in which two nodes can be connected by more than one edge and in which nodes and edges can have attributes) was proposed as a model. This model captures properties from all social media sites.

There is the following argument:

1. Site ontologies have a finite number of different concepts.
2. Binary relations exist on the set of the concepts.
3. Concepts can be characterized by a finite number of attributes.

Therefore, such data can be modelled by a special type of graph. This graph would consist of labelled nodes and edges, in which two nodes can be connected by more than one edge and in which nodes and edges can have attributes. So, a three-layer modelling framework was introduced. The universe of discourse of the 1st level model is the immediate social reality, part of which is modelled by the contents from the social media sites. They are modelled by a temporal multirelational graph, which is modelled by DBMS implementation. Thus, **research question II** was answered.

Also, **research question IV** was answered. The answer to this question led to the description of fundamental limits of the monitoring systems, which are described in Article VI. The answer was found analytically. Properties of the environment were established with conceptual-analytical research.

2.3.3 Constructive research

This part of the research aims to construct the artefact, a generic social media monitoring system. The constructive method with parallel specification of a goal state and implementation were conducted. The spiral method of system development was applied. That is the answer to **research question III**. There were the following iterations:

1. Development of the requirements for the first version of generic social media monitoring software, which aims to monitor and analyse the data from different problem domains. Described in Article III.
2. Development of the data repository that implements the temporal multirelational graph. The repository is able to store heterogeneous data from social media sites and provide a homogenous interface for its population and querying. Described in Article VI.

3. Development of a crawler able to crawl data from social media sites, the structure of which is described by the ontologies, which contains the procedures to extract the data from sites with XPath or regular expressions. The crawler establishes a relation between 1st and 2nd levels of the three-layer model. Described in Article IV.
4. Implementation of the user interface and data model of the application, in which many users can use the system at the same time and arrange tasks for crawling and analysis, which are run in a certain interval of time.

The following results were obtained:

1. Requirements specification for the generic social media monitoring system.
2. Multi-module architecture of the system, which has the following modules: a crawler module, repository module, and analyser module.
3. Architecture of the crawler, which is able to crawl the data from social media sites, where data descriptions are given by ontologies.
4. An interface for the repository implementing a temporal multirelational graph. Current implementation is done with a temporal database on PostgreSQL.
5. Architecture of the analyser module.

Finally, the software was used to collect real data. During the experimentation, the following data were collected:

1. A set of communities around real school-shooter profiles in the social media site LiveJournal.com;
2. Nine million nodes of social network from LiveJournal.com; and
3. Content of hidden services from Tor networks: an imageboard named “Thorlauta” and a forum named “Suojeluskunta”.

Analysis of the size of the sites was carried out, and popular topics were detected. Articles III and VII contain the results.

3 RESULTS

The main goal of this thesis was to develop the monitoring and analysis software. The spiral software development model was used as a method (Boehm, 1986) to achieve this goal. Also, case studies were used to study phenomena more deeply.

3.1 Probabilistic model concerning school shooters

Historically, the first phase of the research included conceptual-analytical research aimed at analysing the *world* and *system-as-is* (Lamsweerde, 2009). During this phase, analysis of the environment was conducted. The result of this part was the probabilistic model, which allows an estimation of the probability of success or failure of the detection of the perpetrators. Table 1 contains the probabilities, represented in the model.

Expression (1) shows the probability of averting the attack. Expression (2) is the overall measuring function and shows the probability that the person is not harmful. Expressions (3) and (4) describe false positives and false negatives, respectively. The probabilistic model acts as a starting point in requirements elicitation, where it describes the world for which a machine solution should be developed.

$$P_{i,r,a}^{avert} = P_i^{avert}(t)|\, P_{i,r,a}(t < a - r) \tag{1}$$

$$\begin{aligned} P_i^{no\ harm} = {} & \left(P_i^{dana} \times \left(P_i^{iana} \middle| P_i^{dana}\right) + P_i^{ddig} \times \left(P_i^{idig} \middle| P_i^{ddig}\right)\right) \times P_i^{redirect} \\ & \times \left(P_i^{iattack} \times (P_i^{avert}(t)|P_i^{iattack}) + P_i^{dattack} \times \left(P_i^{avert}(t) \middle| P_i^{dattack}\right)\right) \end{aligned} \tag{2}$$

$$P_i^{false\ positive} = \left(P_i^{wnana} \middle| P_i^{dnana}\right) + \left(P_i^{wndig} \middle| P_i^{dndig}\right) \tag{3}$$

$$P_i^{false\ neg} = P_i^{dana} \times \left(1 - P_i^{iana} | P_i^{dana}\right) + P_i^{ddig} \times \left(1 - P_i^{idig} | P_i^{ddig}\right) \tag{4}$$

TABLE 1 A probabilistic model for perpetrator monitoring

Symbol	Description
$P_{i,r,a}^{avert}$	The probability of averting a planned attack
$P_i^{no\ harm}$	Overall measuring function
$P_i^{false\ positive}$	Probability of a false positive detection
$P_i^{false\ neg}$	Probability of a false negative detection
P_i^{dana}	Probability that a person i discloses information in analogue form
P_i^{ddig}	Probability that a person i discloses information in digital form
P_i^{iana}	Probability that another person or group of people correctly interprets the analogue information
P_i^{idig}	Probability that another person or group of people correctly interprets the digital information
$P_i^{iattack}$	Probability that the potential perpetrator i releases the attack plan in analogue form
$P_i^{dattack}$	Probability that the potential perpetrator i releases the attack plan in digital form
[r,a]	Time interval between the release of the attack plan information or exposure that the attack will occur, and the planned start of the attack
$P_i^{avert}(t)$	Probability that someone can avert i's attack t seconds after he or she gets the information that it is going to happen
$P_i^{redirect}$	Probability that the potential perpetrator i is moved to a "normal path" in his life before attack
P_i^{dnana}	Probability that a person i discloses information that he is on normal path in analogue form
P_i^{dndig}	Probability that a person i discloses information that he is on normal path in digital form
P_i^{wnana}	Probability that another person j or group of people wrongly interprets the analogue information and decides that i is on dangerous path
P_i^{wndig}	Probability that another person j or group of people wrongly interprets the digital information and decides that i is on dangerous path

The probabilistic model capturing the behaviour of school shooters was elaborated in Article I. The probabilistic model addresses fundamental limitations of detecting school shooters based on monitoring social media sites. In addition, that article contains a list of new technologies that can be used to detect school shooters.

The probabilistic model shows that factors which affect the probability of no harm include digital disclosure of the information about the attack. Then, an automatic detection system could affect this and raise the probability of detecting disclosed information. Probabilities can be assessed further based on the data about past school shootings.

3.2 Qualitative and quantitative analysis

Next, an analysis of the traces left by real school shooters was conducted. The analysis addressed the question whether it is possible to find traces in the web that would make finding the perpetrators possible in theory. Eleven major cases between 2005 and 2011 were analysed. The year 2005 was selected as the starting year because of the emergence that year of various social media sites, together with the growth of the number of shootings (Böckler, Seeger, and Heitmeyer, 2011). Information was gathered from various sources, mainly web and social media sites. Information stored by other users (in the form of saved HTML pages, screenshots, etc.) also was considered, along with articles and reports documenting the cases. Prior to the analysis, information was verified.

It was found that 7 out of 11 perpetrators left various traces online; thus, if these traces would have been interpreted correctly, shootings could have been prevented. The results are discussed in detail in Article II.

Targeting social media sites and crawling them with certain keys might reveal communities with earlier perpetrators. And evidently, social media sites would attract additional perpetrators who could be detected with the social media monitoring and analysis system to be developed.

In terms of requirements engineering, the present analysis describes the world and system-as-is.

3.3 Requirements and architecture of the generic social media monitoring system

After that, the basic requirements and architecture (Figure 4) of the system-to-be were elaborated. The requirements were elaborated based on an analysis of the literature and the domain knowledge about social media and their monitoring.

The following requirements were elicited (from Semenov, Veijalainen, and Boukhanovsky, (2011)).

Functional requirements:

F1 Ability to let a human user describe which sites will be the target of data collection, when this will take place, what data are to be collected, and how they can be accessed (a metadata description of the structure, e.g., ontology).

F2 Ability to access various social media sites as well as other web sources and retrieve any accessible raw data stored at them, including profile data, “friends” or similar relationships, multimedia content attached to the profile, and comments attached to the content or profile by other users.

F3 Ability to retrieve profiles or content from social media sites based on keywords or a larger ontology description given by a human user.

F4 Ability to repeatedly retrieve a social network around a particular person (profile) on a particular site based on the “friends” or similar relationships recursively to a given distance (“friends”, “friends” of “friends”).

F5 Ability to retrieve the entire social network at a particular social media site; i.e., all profiles, relationships, and multimedia contents.

F6 Ability to create automatically a set of (graph) models of the social networks at a social media site based on the profiles and other data retrieved from them.

F7 Ability to store the raw multimedia data as well as the above models persistently and in such a way that different instances of the social network model (e.g., around a certain person) and data can be distinguished and placed on a timeline.

F8 Ability to automatically or semi-automatically analyse various properties of the stored models (and raw data) and visualise the results to a human user or store them persistently for later use.

Non-functional requirements:

NF1 The system must run in parallel so that data collection (crawling) can happen simultaneously with the analysis.

NF2 The crawling performance should be such that all changes at social media sites around the targeted persons can be captured and stored.

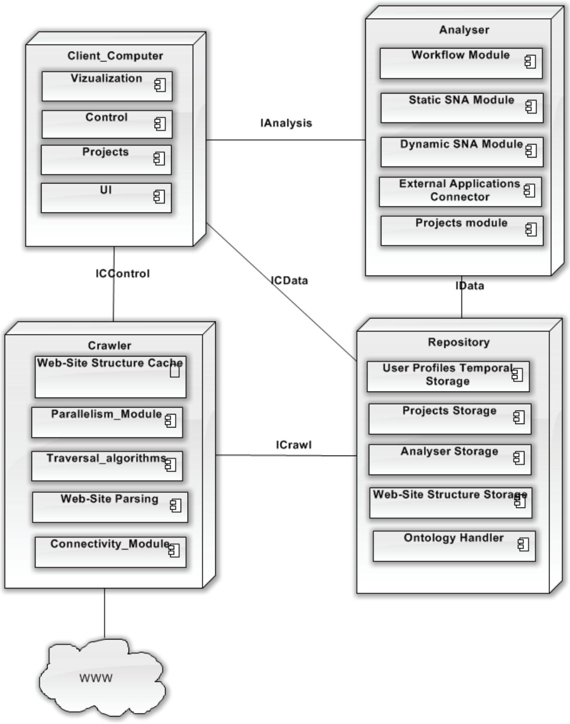


FIGURE 4 Software architecture

The software consists of the crawler module, repository, and analyser. There were several iterations, which included elicitation of the requirements, elaboration of the architecture, and implementation of the prototypes. Requirements elicitation is described in Article III. Figure 4 depicts the architecture.

3.4 The three layer modelling framework

Then, the three-layer modelling framework was elaborated (Figure 5). The framework acts as the theoretical underpinning for social media monitoring and analysis software. In this framework, the universe of discourse (UoD) is the immediate social reality, some part of which is modelled by the 1st level model, the content of the social media sites, which adheres to the site's social world model and captures some aspects of the immediate social reality. The social world models are modelled as site ontologies, which are modelled by a temporal multirelational graph.

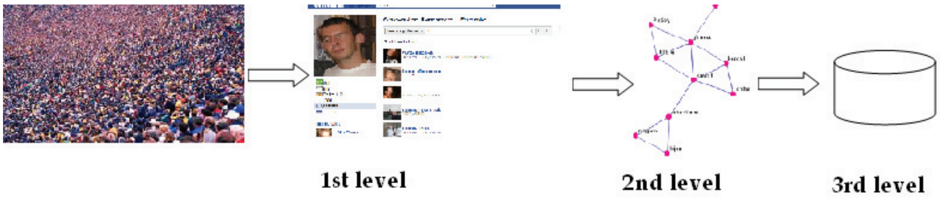


FIGURE 5 Three-layer modelling framework

Thus, the 1st level model acts as the UoD for the 2nd level model, which is a temporal multirelational graph capturing the site ontologies and evolution of their instances (5)

$$G = \{N, E, NT, ET, TN, TE, ValN, ValE\} \tag{5}$$

Where:

N is a countable infinite set of *nodes*.

$\boldsymbol{E_{k,t}} \subset \boldsymbol{N_{k,t}} \times \boldsymbol{N_{k,t}}$ is a multiset of *edges* {$E_{k,1,t}$, . . . , $E_{k,r,t}$} representing different relationships between nodes at site S_k at the moment *t*.

NT = {$NT_1(A_{11}, . . . ,A_{1k1}), . . . ,NT_n(A_{n1}, . . . ,A_{nkn})$} is the finite set of *node types*.

ET = {$ET_1(A_{11}, . . . ,A_{1k1}), . . . ,ET_n(A_{n1}, . . . ,A_{nkn})$} is the finite set of *edge types*.

TN: {TN_{it}: $N_{i,t}$ -> **NT**, 0 < i < j + 1, t >= 0} is an infinite family of functions TN_{it}, each of which maps a node to a node type.

TE: {TE_{it}: E_{it} -> **ET**, 0 < i < j + 1, t => 0} is an infinite family of functions TE_{it}. An individual function TE_{it} attaches a type to each edge in E_{it}.

ValN(t,k,TN$_i$), **0 < i < n**, and **ValE(t,k,TE$_i$)** are the valuation functions.

This thesis argues that such a graph can capture evolving data from all known social media sites. The presence of the 2nd level model would allow application of the same analysis methods to heterogeneous social media sites. Also, all social media sites could be monitored with the same software system.

The 3rd level model is the implementation of the repository, which brings in performance aspects.

Article VI contains the details of the three-layer framework and a discussion about the monitoring limits.

3.5 Repository interface

The second layer of the model allowed the building of a data repository with a homogeneous interface. A subsequent part of the thesis included the development of the more advanced architecture of the software and its implementation. Following the spiral method, new requirements were elicited and the architecture of the software was enhanced: ontology support was added to the crawler module and a more advanced repository interface was developed in order to allow the use of different data management systems.

Table 2 presents the repository interface. The repository interface is built laying on the temporal multirelational graph, and the repository is able to store different site ontologies.

Presently, the repository is mainly implemented as a PostgreSQL database (DB) with the interface software that implements the above operations on top of it. There are possibilities to use two schemas. Details are available in Article V. The first possibility is to store each snapshot individually:

```
NODES(i_id, node_id, node_type, T);
EDGES(i_id, node_from, node_to, edge_type, T);
N_ATTRS(i_id,node_id, attr_id, attr_value, T);
E_ATTRS(i_id,node_id, attr_id, attr_value, T);
MDATA(i_id,name);
S_GRAPHS(s_id, i_id, node_id);
S_GRAPH_META(s_id,i_id, name);
```

The second possibility to endow each entity (node, edge, and attribute) with a time interval (temporal database):

```
NODES(i_id, node_id, n_type, t_st, t_end);
EDGES(i_id, node_from, node_to, e_type, t_st, t_end);
N_ATTRS(i_id, n_id, a_id, a_value, t_st, t_end);
```

E_ATTRS(i_id, n_id, a_id, a_value, t_st, t_end);
MDATA(i_id,name);
S_GRAPHS(s_id, i_id, node_id);
S_GRAPH_META(s_id, i_id, name);

TABLE 2 The repository interface

	Description	Name	Return
1	Add instance X to the repository	addGraph()	ID of the instance X

2	Add or remove node N of type C to or from graph X at time T	addNode(X,N)
		rmNode(X,T,N)
			Node

3	Add or remove edge E connecting nodes N1 and N2 to or from graph X at time T	addEdge(X,N1,N2,type)
		rmEdge(X,T,N1,N2,type)
			Edge

4	Add attribute A to node N or edge E of graph X at time T	addnAttr(X,A,N)
		rmnAttr(X,T,A,N)
		addeAttr(X,A,E)
		rmeAttr(X,T,A,E)	

5	Extract graph G (slice of the instance X) at time T, in a predefined format	getGraph(X,T,Format)	Graph
6	Extract the validity interval [t1, t2] for instance X	getInt(X)	Time interval
7	Extract the sequence of the graphs {G} during time interval [t1, t2] from instance X	getSeq(X,t1,t2)	List of graphs G

8	Extract inserted or deleted nodes or edges in X during [t1,t2]	getNewNodes(X,t1,t2)
		getNewEdges(X,t1,t2)
		getDelNodes(X,t1,t2)
		getDelEdges(X,t1,t2)
			List of nodes/edges

9	Extract the graph, induced by certain types of nodes or edges from the graph during [t1,t2]	getGraph(types)	Graph
10	Get the attribute value of node N or edge E of X at time T	getVal(X,A,T)	Value
11	Get the node by its attributes	getNode(attr A,X)	Node
12	Get the edge by two nodes and type	getEdge(N1,N2,X,type)	Edge

3.6 Crawler architecture

Figure 6 depicts the architecture of the crawler, making use of site ontologies. The Object Oriented OO diagram of the crawler is depicted in Figure 7. The crawler was developed using Twisted, a library for asynchronous programming for Python (Twisted, 2012).

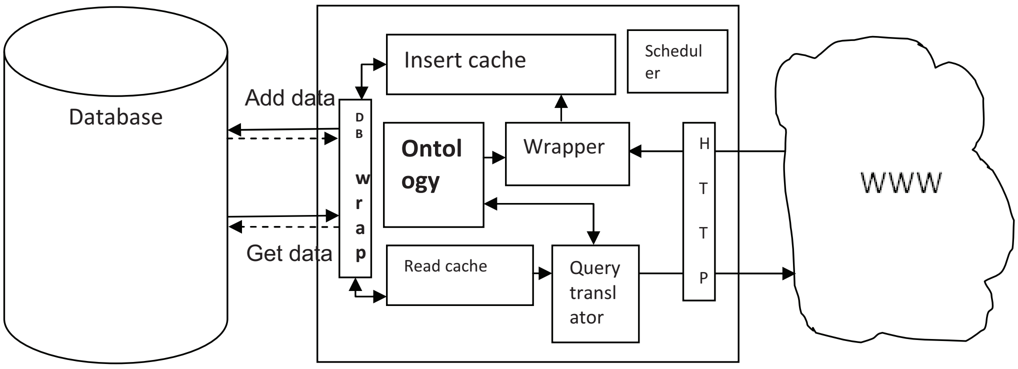


FIGURE 6 Crawler architecture (Semenov and Veijalainen, 2012a)

Algorithm 1 Crawler main loop

```
1: initialize seeds, crawling ontology
2: output: multirelational directed graph
3: while frontier not empty do
4:    fetch entity from frontier
5:    extract outgoing edges and their parse rules
6:    extract URL of entity
7:    connect to URL, fetch data from it
8:    while parse rules not empty do
9:       apply parse rule (regular expression) to data
10:        put resulting nodes into repository
11:        add edge to repository
12:    end while
13: end while
```

There are the following modules:

Database: The database that stores the gathered data. In the current implementation, the PostgreSQL DBMS is used.

DB Wrap: The interface for the repository (see Table 2). The purpose of the module is to allow the use of different DBMSs, such as a relational DBMS (e.g., PostgreSQL) or a graph database (e.g., Neo4j). The interface implements the access to the database. It exports functions that allow selection of the stored entities from the DB and insertion of the parts of the crawled graph.

Insert cache: Stores the list of the elements that are to be put to the data repository.

Read cache: Stores the queue of the elements to be traversed by the crawler (in order to minimize the number of queries to the DB).

Currently it is organized as additional attribute “color”, which is added to graph node. If color = 0, then node is not traversed yet. If color = 1, node is in cache. If color = 2, node is parsed.

Algorithm 2 Read from cache

```
if cache is not empty do
    return one node
end if
else
    extract from repository nodes having color = 0
    set color of extracted nodes = 1
end if
```

Wrapper: Contains procedures of data extraction for the current site ontology. Makes use of XPath and regular expressions to translate concepts from HTML to the ontology form.

Query translator: The module that translates the semantic entity description to request which should be handled by the HTTP connection module, dependent on the description from the ontology.

HTTP: The connection module (handles HTTP protocol, requests, and redirects).

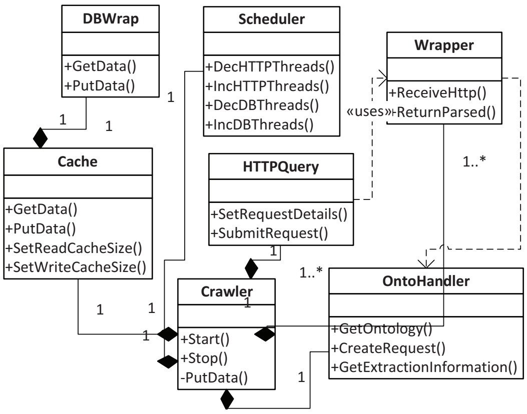


FIGURE 7 OO diagram of the crawler

Connections are carried out through array of deferreds:

```
……………
        self.deferreds = [Deferred() for i in range(num_connections)]
……………

        agent = Agent(reactor)
        logging.critical('URL is ' + self._getURL(node))
        self.deferreds[ind] = agent.request('GET', self._getURL(node), Headers(headers), None)
        self.deferreds[ind].addCallback(cbRequest)
        self.deferreds[ind].addCallback(dataProcess)
        neighbor_edge_types = self._getOutgoingEdgesTypes(node)
        for i in neighbor_edge_types:
```

```
    func_tmp = self._getParse(i)
    self.deferreds[ind].addCallback(func_tmp)
    self.deferreds[ind].addCallback(self._saveParsed, node, i)
self.deferreds[ind].addCallback(self._getSeeds, 1)
self.deferreds[ind].addCallback(self._printTmp, node, ind)
```

Ontology: The ontology, having a description of the current crawling task along with the specification of relations, which should be considered sources for new entities for traversal.

Scheduler: Contains control modules for the Twisted library. The crawler is implemented using an asynchronous event-driven networking engine for the Python programming language, Twisted. Currently scheduling is represented as possibility to change number of deferreds.

3.6.1 Ontology handling

Two types of ontologies are used in the crawler. The first one is the site ontology, which represents the concepts and their relationships at the monitored site; the second one is the crawling ontology, which is subset of the site ontology. An example of site ontology is depicted in Figure 8. An example of the crawling ontology for this site ontology is depicted in Figure 9.

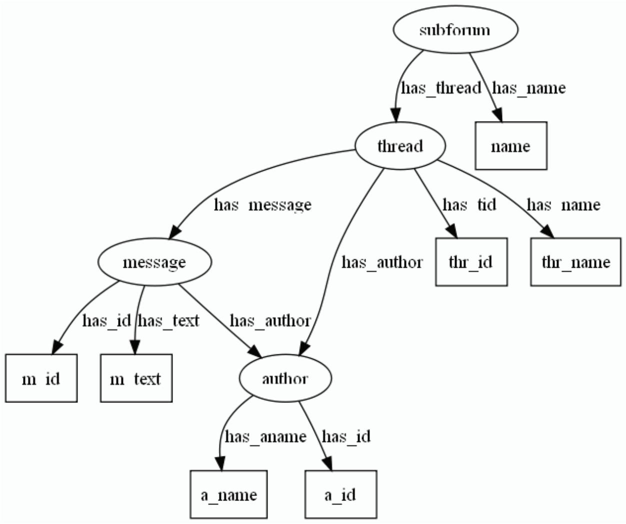


FIGURE 8 A site ontology of Suojeluskunta

The following code would generate the site ontology (internally, it is represented as a multidigraph, in which each node and edge may have a type and attributes).

```
thread_node = Node()
thread_node.setAttr("type", "subforum")
thread_node.setAttr("name","", "has_name")
thread_node.setAttr("fid","", "has_fid")

topic_node = Node()
topic_node.setAttr("type", "thread")
topic_node.setAttr("thr_name", "", "has_name")
topic_node.setAttr("thr_id", "", "has_tid")

msg_node = Node()
msg _node.setAttr("type", "message")
msg _node.setAttr("m_id", "", "has_id")
msg _node.setAttr("m_text", "" "has_text")
crawlontology = CrawlOntology()
crawlontology.addNode(thread_node)
crawlontology.addNode(topic_node)
crawlontology.addNode(user_node)
crawlontology.addEdge(thread_node, topic_node, "has_thread", "link")
crawlontology.addEdge(topic_node, msg _node, "has_message", "link")
```

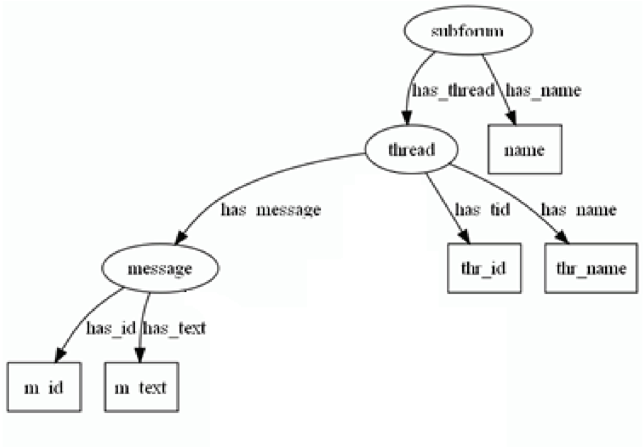


FIGURE 9 A crawling ontology for the above site ontology

Then, each node is endowed with URL generation function:

```
crawlontology.addURLFunc("thread", lambda x: "http://
v7ovl2hciwt72lqi.onion/forum/showthread.php?tid=" + str(x["thr_id"]))
```

and each edge is endowed with a parsing function that takes raw data of the previous node as the input and produces the nodes as output. Parsing may be carried out using regular expressions:

```
    def threadParse(data):
        a =
re.finditer('href=\"showthread\.php\?tid=(?P<tid>[\d]+?)\" class=\"
subject_[newold]+\"
id=\"tid_[\d]+\">(?P<name>[\s\S]+?)</a>(?P<sp>[\s]*</span>)?(?(sp)|[\
s]+<span
class=\"smalltext\">\(Pages:[\s]*(?P<pages>[\s\S]+?)[\s]*\)</span>[\s]+<
/span>)[\s]+<div class=\"author smalltext\">(?P<tmp><a
)?(?(tmp)href=\"http:\/\/v7ovl2hciwt72lqi\.onion\/forum\/member\.p
hp\?action=profile&amp;uid=(?P<uid>[\d]+)">(?P<unamehref>[\s\S]+?)<
/a>|(?P<unamewohref>[\s\S]+?))</div>[\s]+</div>[\s]+</td>[\s]+<td
align=\"center\" class=\"trow[\d] forumdisplay_regular\"><a
href=\"javascript:MyBB\.whoPosted\([\d]+\);\">(?P<posts>[\d]+)</a><
/td>[\s]+<td align=\"center\" class=\"trow[\d]
forumdisplay_regular\">(?P<views>[,\d]+)</td>',data)
        res = []
        for i in a:
            k = i.groupdict()
            tmp_node = Node()
            tmp_node["type"] = "thread"
            tmp_node["thr_id"] = k['tid']
            tmp_node["thr_name"] = k['name']

            res.append(tmp_node)
        return res

    crawlontology.addParseFunc("has_thread", threadParse)
```

The crawler carries out the graph traversal process, in which, while following the edge, it executes parsing functions that parse raw data and generate the next nodes. During the processing of the node, the crawler checks whether the node can be crawled and calls the URL generation function, which downloads the data. Then the crawler traverses further, according to the crawling ontology.

3.7 Fundamental limits of monitoring

The fundamental limits of monitoring were analytically estimated. The monitoring software accesses the monitored site remotely, through the Internet, or more abstractly, through a channel characterized by typical parameters. Also, the contents, i.e., instances of social media site ontology, change with some speed. Basically, for monitoring, the rate of data collection should be faster than the speed of the changes of the site.

The following theorem and two corollaries show the limits of monitoring.

Theorem 1. Let us assume that the monitoring site uses the polling method to capture the changes in VC_0 and that it can only use an exhaustive scan. The monitoring site is able to capture all the changes in VC_0 during a fixed period of time, lag > 0 (in s), if it holds:

$$lag \, * \, R_{pol} \, > \, |VC_0| \text{ and } lag \, * \, R_{pol} \, > \, R_{snch} \tag{6}$$

$$\text{and } Size(VC_0) * 8/C_{ch} \, < lag \, < \, L_{snc} \tag{7}$$

Where:

R_{snch} (changes/second) is the average rate of changes at the social media site (or in a certain part of it) to be monitored.

L_{snc} (seconds) is the shortest living time of the change at the social media site.

C_{ch} (bits/second) is the average communication (downlink) channel capacity between the site to be monitored and the monitoring site.

R_{pol} (request/second) is the average polling rate that the monitoring site is able or allowed to generate towards the site to be monitored.

lag (seconds) is the fixed period of time during which collection is carried out.

VC_0 is the virtual community.

Corollary 1. If the site to be monitored grows monotonically and the downlink channel capacity is non-zero, then all individual changes can be captured. If $R_{pol} \geq R_{snch}$, then there is an upper time limit for capturing all the changes. If $R_{pol} < R_{snch}$, then all changes eventually can be captured, but the difference between the real state of VC_0 and the corresponding state at the monitoring site can differ arbitrarily.

Corollary 2. If the monitored site pushes the modified entities to the monitoring site with fixed lag l and the monitoring site can store and analyse the data at least as fast as the data arrives, then the monitoring site is able to capture all the changes if the size (changed entities/second) * 8 < C_{ch}.

Proofs of the theorem and corollaries are presented in Article VI.

3.8 Real data collection

Next, as a proof of the concept, software was used to collect data from social media. Software was deployed at a 16-core server, having 193 GB of RAM, and 4 TB of hard disk space running on Linux. Then, the entire social network of site LiveJournal.com, which consisted of 9M nodes, was collected. Also, the contents from the Finnish segment of the Tor network were collected and analysed. The analysis of LiveJournal is described in Articles III and V. A description of the

Tor contents is presented in Article VII. During spring 2013 software was used to collect data from Twitter.

4 RELATED WORK

The present section contains research background from the fields relevant to the thesis.

4.1 Social media

The term *virtual community* was coined by Rheingold (1993, p. 5): “Virtual communities are social aggregations that emerge from the loosely connected computer network when enough people carry on those public discussions long enough, with sufficient human feeling, to form webs of personal relationships in cyberspace”. Boyd and Ellison (2007) define social media sites as web-based services that allow individuals to: (1) construct a public or semipublic profile within a bounded system; (2) articulate a list of other users with whom they share a connection; and (3) view and traverse their list of connections and those made by others within the system. Kaplan et al. (2010) list categories of social media such as blogs, online content communities, social networking sites, virtual game worlds, virtual social worlds, and collaborative communities. Kietzmann et al. (2011) introduce seven building blocks for social media: identity, conversations, sharing, presence, relationships, reputation, and groups. Bertot, Jaeger, and Hansen (2012) notice that *social media* is a relatively new term, but the idea of applying online tools to facilitate social interaction was elaborated long ago. Predecessors of social media are email lists, Usenet, and bulletin boards.

Examples of the most popular social media sites are Facebook, which announced one billion users in October 2012 (Facebook Newsroom, 2012); Twitter, with more than 500M users (Semiocast — Twitter reaches half a billion accounts, 2013); and YouTube, with number of users in the same range. Social media sites exist within the web, and many of them have various desktop and mobile clients.

Data from social media sites can be gathered in a number of ways. The most straightforward way is to use web-crawling software, accessing the web interface of the site. However, social media sites frequently provide a special application programming interface (API), which allows more convenient data collection. Usually, an API requires authentication and gives some subsets of data from the site. Generally, there are two types of APIs. The first is request-response type and provides interfaces, which allow querying of the site; the second type of API sends filtered data, when it appears on the site, to the requester (streaming API).

Digitally-encoded data may be classified as public, semipublic, or private (Semenov and Veijalainen, 2012b). Public information may be accessed by everyone who desires it. Semipublic information may be accessed by small groups of individuals, who must be authenticated and authorized. Private information may be accessed by the owner only (or based on special access rights granted by the owner).

Digitally-encoded information in the web also may be classified as surface web and hidden web (deep web). Surface web information, or publicly indexable web information, may be indexed by search engines. Hidden web information is not easily accessible for standard crawling since it may be stored in databases, and pages would be created dynamically as a result of a special search or query. Also, hidden web information may require authentication. Bergholz and Childlovskii (2003) note that, in 2000, the size of the hidden web was about 400 to 500 times larger than the surface web. According to data classification used in this thesis, public data are located in the surface web, and the hidden web may contain semipublic and private data.

There is a lot of research devoted to analysing the data presented in social media. Prochaska, Pechmann, Kim, and Leonhardt (2012) discuss their analysis of quit-smoking social networks on Twitter. Erik Tjong Kim Sang, Bos, Inkpen, and Farzindar (2012) discuss the possibilities of predicting the 2011 Dutch senate elections by analysing Twitter data. Larsson and Moe (2012) discuss political microblogging in Sweden. Skoric, Poor, Achananuparp, Lim, and Jiang (2012) study the use of Twitter during elections in Singapore. Doan, Ohno-Machado, and Collier (2012) discuss tracking influenza-like illnesses by analysing messages from Twitter. An analysis of earthquakes based on messages from Twitter is discussed by Earle, Bowden, and Guy (2011).

Aliprandi and Marchetti (2011) describe CAPER, a Collaborative Platform for Open and Closed Information Acquisition, Processing and Linking. The goal of CAPER is to prevent organized crime by exploiting and sharing open and closed information sources.

IARPA (2011) runs a program to continuously and automatically monitor public data and predict events. This activity also may be referred to as *open source intelligence* (Schaurer and Störger, 2010).

4.2 Crawling

This section describes research in the field of architecture of web crawlers' architectures and methods of extraction of structured data from the web.

4.2.1 Crawler architectures

There are a number of research papers that discuss the construction of web crawlers. A crawler is the software that crawls over the WWW and saves data. A crawler extracts URL addresses of web pages by parsing already stored content and then crawls further. Cheong (1996, p.96) defines *crawlers* as "software programs that traverse the WWW information space by following hypertext links and retrieving web documents by standard HTTP protocol". Crawlers are also called "web robots" (Heinonen, Hätönen, and Klemettinen, 1996) and "web spiders" (Thelwall, 2001). Crawlers are used by web search engines like Google or Yahoo (Brin and Page, 1998) and by various business intelligence software tools (Menczer, Pant, and Srinivasan, 2004). Crawlers also can be used to harvest information for spam and phishing purposes (Jagatic, Johnson, Jakobsson, and Menczer, 2007).

Typically, crawlers consider the web as a graph, nodes of which are web pages, and edges of which are links between the web pages. Then, crawlers traverse this graph using some graph traversal algorithm, such as breadth-first traversal (BFS), depth-first traversal (DFS), or another algorithm. Liu and Menczer (2011) provide a flow-chart diagram for the basic sequential crawler (Figure 10).

First, a crawler initializes list of the URLs to visit called the frontier with seed URLs. Then, the crawler dequeues the URLs from the frontier and fetches the corresponding resource from the web (by accessing it via HTTP). Next, the crawler extracts the URLs from a fetched page, adds them to the frontier, and stores the contents of the page in the repository. The crawling process stops when the frontier becomes empty.

In the case of a BFS, the frontier may be implemented as a first-in-first-out (FIFO) queue; in the case of a DFS, it may be implemented as a last-in-first-out data structure. There are also preferential crawlers that assign a priority to each URL in the frontier and fetch the URLs according to priority.

Focused crawler "seeks, acquires, index, and maintains pages on a specific set of topics that represent a narrow segment of the web" (Chakrabarti, van den Berg, and Dom, 1999, p. 546). Focused crawling is based on the topical locality of information on the web (Davison, 2000), meaning that pages are usually linked to pages with similar topics. Peng and Wen-Da (2010) describe the application of a focused crawler to crawl information in a financial field. Yang and Hsu (2009) describe focused crawling of sites with calls for papers.

Since the web is evolving, and thus collected information is becoming outdated, it may become necessary to update the data. There are crawlers that conduct periodic updates, which means recollecting all the pages after some interval of time and incremental updates when newly created pages are identified and crawled. Yih, Chang, and Kim (2004) describe incremental crawling of the forum. Timestamps attached to forum messages, indicating the time at which the messages were sent, were used to detect whether the message is new and should be crawled or old and thus already crawled.

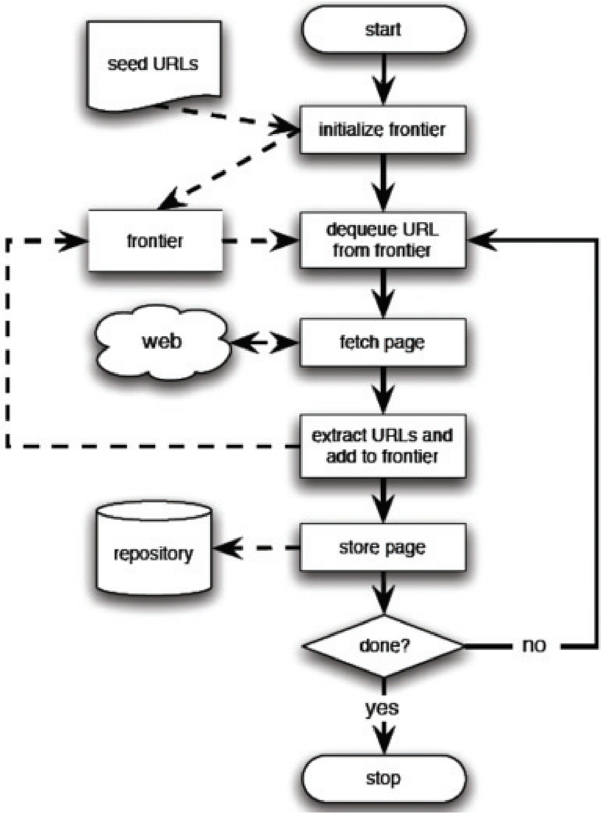


FIGURE 10 Diagram for a sequential web crawler, (B. Liu et al., 2011, p.313)

Batsakis, Petrakis, and Milios (2009) evaluate the approaches to focused crawling and describe the application of hidden Markov models (HMM) for focused crawling. HMMs are used to learn the paths leading to web pages with relevant content.

Dong, Hussain, and Chang (2008) discuss the application of semantic web technologies to focused crawlers. Ontologies are used there to link the fetched documents with the concepts from ontologies. A crawler using a similar approach is presented in (Yuvarani, Iyengar, and Kannan, 2006) and (Yang and Hsu, 2010).

Duda, Frey, Kossmann, Matter, and Zhou (2009) describe a crawler that deals with asynchronous JavaScript and XML (AJAX). The crawler recognizes AJAX requests and runs them, thus imitating a browser. Crawling AJAX-enabled sites also is discussed in (Mesbah, Bozdag, and Deursen, 2008).

Fu, Abbasi, and Chen (2010) describe the crawler for dark-web forums, which comprise a "problematic facet of the Internet" and contain various extremist groups' discussions. The human-assisted accessibility approach is used to get access to the forums; in our view, that is access to semipublic data. The main goal of their research was to collect hate and extremist group content from hidden web forums. Fu et al. (2010) argue that there are two techniques to access the hidden web. The first one is to apply automatic form fillers; and the second one is the human-assisted approach. The extent of human involvement depends on the complexity of the site. Fu et al. (2010) discuss the registration process at the forums and remind that many dark-web forums do not allow anonymous access. In some cases, a personal approval from the webmaster of the forum is needed. Appropriate **s**pidering parameters also are discussed, such as the number of connections, download intervals, and timeouts (since intensive crawling may trigger various blocking mechanisms or exhaust the network capacity). Using proxy servers also is considered. In addition, the paper includes a topical analysis of the messages from the eight forums studied and an analysis of the participant interaction. Topics were clustered into 'national socialist movement party news', 'hate crime news', 'politics and religion discussion', 'general discussion', 'Aryan books', and 'Persian content'.

Liu, Liu, and Dang (2011) discuss automatic discovery of hidden web entry points. Madhavan, Ko, Kot, Ganapathy, Rasmussen, and Halevy (2008) describe the surfacing of the hidden web, which is the pre-computation of results of submissions to HTML forms.

Bergholz et al. (2003) describe a hidden web crawler. The crawler contains a starter, which prepares starting points; a local crawler, which discovers interesting pages; a form analyser, which analyses found forms; and a query prober, which tries to make a query to the form. The crawler searches for the forms, fills them in, and sends a generated HTTP request to the server. Bergholz et al. (2003) describe an experiment in which a crawler is used to detect hidden web resources from Google's Directory. Zhang, Dong, Peng, and Yan (2011) also describe the framework for deep web crawling.

Google uses sitemaps (sitemaps.org - Home, 2012) to access the hidden web. A sitemap is an "XML file that lists URLs for a site along with additional metadata about each URL (e.g., when it was last updated, how often it usually changes, and how important it is relative to other URLs in the site) so that search engines can more intelligently crawl the site." (sitemaps.org - Home, 2012). The webmaster of the site that is to be crawled generates the sitemap and submits it to Google. Site map protocol is supported by the major search engines.

Glance, Hurst, and Tomokiyo (2004) describe BlogPulse, a blog-analysis portal that contains the analysis of key trends from approximately 100 000 crawled blogs. Glance, Hurst, Nigam, Siegler, Stockton, and Tomokiyo (2005) describe a system for collecting data from online message forums and weblogs. The goal of their system was to analyse collected data and provide an interactive analysis for marketing intelligence.

Limanto, Giang, Trung, Zhang, He, and Huy (2005) describe the information extraction engine from web forums. Their system contains the wrapper. The wrapper is generated by locating the repeating entities in content of the HTML page and building a non-deterministic finite-state automaton to extract the data.

Li, Meng, Wang, and Li (2006) describe RecipeCrawler, a focused crawler to collect recipes from web forums. The crawler classifies web pages into two categories: recipe web pages and category web pages. Recipe web pages contain recipe details, and category pages contain links to recipe pages. Automatic extraction of data from both types of pages is based on partial tree alignment (Zhai and Liu, 2005).

Crawling also involves a human operator who annotates the data from the pages for wrapper generation. Aggarwal, Al-Garawi, and Yu (2001) describe an intelligent crawler that relies on learning to further prioritise the candidate URLs during crawling, depending on user defined predicates such as groups of the keywords on a page. During traversal of the pages, the crawler builds a statistical model depending on the following features: content of the page that links to candidate pages; URL tokens from candidate URLs; the nature of the pages; and the nature of sibling pages of the candidate. The crawler estimates the probability of satisfaction of the predicate, depending on these features, for the URLs from the frontier. Jalilian and Khotanlou (2011) describe the architecture of the focused crawler, which uses ontologies to detect the topic of the page.

Leung, Lin, Ng, and Szeto (2009) describe the implementation of a focused crawler for Facebook. Extraction of the user information from the HTML profile page and URL is discussed. Also, they discuss such countermeasures as CAPTCHA, shown by Facebook.

Henrique, Ziviani, Cristo, de Moura, da Silva, and Carvalho (2011) describe a fast algorithm to verify the URL uniqueness for crawlers in large-scale settings. To verify the uniqueness, the crawler checks whether the URL is presented in the repository.

Mukhopadhyay, Mukherjee, Ghosh, Kar, and Kim (2011) describe the architecture of a scalable parallel web crawler. They identify the following challenges: prevention of crawling the same pages by crawlers running in parallel, minimizing the communication overhead between parallel-run crawlers, and maintaining a high quality of downloaded documents where the back-link count is used as a quality measure. Yadav (2010) describes the implementation of an incremental parallel web crawler. Yadav (2010) discusses avoiding the overlap among crawled URLs, ranking URLs in the frontier, and web documents that change frequencies.

Yang (2010) describes the crawler with ontology-supported web site models. There, ontologies contain metadata about the sites and pages (the URL, statistical data, and information about the topic of the page, which is filled by a keyword search). Ontologies are constructed in advance and instantiated by the crawler.

In addition, there are frameworks that allow the building of web crawlers. Examples include Scrapy (An open source web scraping framework for Python, 2012). Scrapy is the framework for the Python programming language. It provides convenient functions and classes for extraction of data from web pages, based on regular expressions and XPath; it also isolates low-level functions, such as HTTP connections.

Zhang and Nasraoui (2009) discuss crawling of social media sites. Classification methods are used to identify list pages, detail pages, and profile pages at the crawled sites. Architecture of a Twitter crawler working in the cloud is discussed by Noordhuis, Heijkoop, and Lazovik (2010).

It is important to consider crawling ethics. Thelwall and Stuart (2006) describe that a crawler may cause a denial of service attack for the site, incur costs due to using bandwidth, crawl copyrighted or private content. There is robot exclusion protocol, which provides a way for administrators of web sites to disallow crawler access to some pages. That is done by adding URLs to the file robots.txt, which is put into the root directory of a site. Description of the robot exclusion protocol may be found at (The Web Robots Pages, 2013).

4.2.2 Wrapper induction

Crawlers deal with the extraction of data from web pages. Structured data in the web is represented by HTML. Nowadays, the latest version of it is HTML5, which contains many new features compared to HTML4 (HTML5 Introduction, 2013). In addition, web pages frequently contain code interpreted by the client machine, such as JavaScript or Adobe Flash; however, HTML5 may be used as an alternative to Adobe Flash (Amazon to Introduce Web-Based Book Previews, 2013).

A wrapper is a program that extracts the content of a particular information source and translates it into a relational form (Kushmerick, 1997). Wrappers were studied in (Blanco, Bronzi, Crescenzi, Merialdo, and Papotti, 2010; Bronzi, Crescenzi, Merialdo, and Papotti, 2011; Liu, 2011a; Xia, Zhang, and Yu, 2010; Zhao, Meng, Wu, Raghavan, and Yu, 2005). There are three approaches to generate wrappers (Liu, 2011a): the manual approach, in which a human programmer writes the wrapper; wrapper induction, which is semi-automatic generation using supervised learning (Cohen, Hurst, and Jensen, 2002; Irmak and Suel, 2006; Muslea, Minton, and Knoblock, 1999); and fully automatic wrapper generation (Liu, 2011a).

During wrapper induction, data extraction rules are learned from the training examples. Training examples are manually labelled by a human user and point to the data that should be extracted from the HTML page.

Muslea et al. (1999) describe embedded catalogue (EC) formalism. An EC is a tree-like structure, the leaves of which are terms of interest for the user. Internal nodes of an EC represent lists of k-tuples. Elements of a k-tuple may be either another list or a leaf. Two rules are generated for each node of the tree: the start rule and the end rule. Extraction rules are based on a sequence of tokens (in HTML) that are used to identify the beginning or end of a target entity. After labelling by the user, a wrapper is generated by applying to training examples a sequential covering machine-learning algorithm (Liu, 2011a). Extraction rules consisting of prefix and suffix tokens are generated for each node of an EC tree, where tokens are sequences of HTML elements or are wildcards. Prefix tokens uniquely identify the beginning of the node, and suffix tokens uniquely identify the end of the node. After generation, the wrapper might be applied to other pages for data extraction.

Wrapper verification is the evaluation of whether the wrapper correctly extracts the data from other examples that were not presented in the training set. Wrapper maintenance is a process of regenerating the wrapper, in case the structure of the page changed (Liu, 2011a).

Zhai and Liu (2007) describe instance-based wrapper learning. The algorithm consists of three steps. First, a random example is selected from a set of unlabelled examples. Then, a user labels the items (parts of the web page) that should be extracted and the system generates a prefix and suffix string for the labelled items as sequences of characters before and after the items. Then, the algorithm starts to apply extraction rules to other pages in the training set by searching the items between the prefix and suffix strings. If not all the items can be discovered from the example, it is given to the user for re-labelling. The presented mechanism is an example of active learning.

The WWW contains a large number of pages, and its structure is constantly changing. According to (Domain Counts & Internet Statistics | Whois Source, 2013), there were approximately 143 million registered top-level domains as of January 2013. Thus, for large-scale crawlers, manual or semi-automatic wrapper generation would be very resource consuming and, due to that, automatic wrapper generation methods were studied. (Liu, 2011a) discuss automatic methods by focusing on two problems. The first problem is the extraction of the items from a single web page. In this case, the input for the algorithm is a full web page and the outputs are the items existing on that page. The second problem is automatic extraction of items from the set of pages with the same template.

An important concept from the area of information extraction is the regular expression, which provides a means to extract patterns of particular strings from text. A formal definition of *regular expressions* may be found in Hopcroft, Motwani, and Ullman (2000).

Regular expressions are implemented in many programming languages, like PHP (PHP: PCRE - Manual, 2012) and Python (re — Regular expression operations, 2012). Regular expressions can be implemented as deterministic finite automatons (DFAs) since, for any regular expression, there is a DFA that accepts the same language as the regular expression denotes and vice versa. In addition, there are algorithms that generate one from the other (Hopcroft et al., 2000).

Another relevant technology is XPath (XML Path Language (XPath) 2.0, 2012). XPath is a query language that allows the processing of values conforming to a data model and provides a tree representation of XML documents as well as atomic values. There are a number of libraries that support XPath and allow extraction of entities from HTML and XML documents; e.g., (The XML C parser and toolkit of Gnome, 2013).

The WWW Consortium (W3C) defines *document object model* (DOM) (W3C Document Object Model, 2013) as a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure, and style of documents (HTML or XML). Liu, Grossman, and Zhai (2003), and Zhai et al. (2005) describe the algorithms applying DOM trees for extraction of data from web pages. Zhai et al. (2005) provide the following method of data extraction. At first, a tag tree of the web page is built. Then, an algorithm mines for data regions, which are the areas that contain lists of similar entities. Then, data records are identified from each data region by analysing HTML tags.

4.3 Data repositories

A database is a collection of data managed by database management system (DBMS) (Ullman and Widom, 2007, p. 1). DBMSs allow users to create new databases with specified schemas and query the data with query language. Data stored in a database are closely related and describe a concrete or abstract part of the world (miniworld).

The first generation of databases included old network and hierarchical database systems. In the 1960s, the CODASYL committee developed a network database standard and concepts such as schemata, data manipulation language, and query language (Bachman et al., 1969).

Codd (1970) introduced the relational model. Generally, this proposed mathematical relations and relational calculus/relational algebra, along with table organization for the data, and operations for the tables. Tables are the implementation concept for the abstract relations. Structured English Query Language (SEQUEL) was developed by IBM researchers as an implementation of relational algebra operating on tables. It then evolved into SQL (Chamberlin et al., 1981). In 1986, it became standardised by ANSI and in 1987 by ISO; SQL is very widely used now.

Later relational DBMSs were referred to as second-generation database systems. Bancilhon (1996) proposed third-generation database systems, object databases having a rich-type system, inheritance, encapsulation of functions, etc. Later, XML databases appeared (Chaudhri, Rashid, and Zicari, 2003). An atomic manifesto was proposed by Jones et al. (2005) due to the need to support concurrency and proliferation of systems built from the components. The key idea of this manifesto was the atomicity of transactions, which affects distributed and parallel computations.

After that, many data management systems appeared that were referred to by the umbrella term NoSQL. Examples are Membase (Couchbase | Simple, Fast, Elastic NoSQL Database, 2013), CouchDB (Apache CouchDB, 2013), MongoDB (MongoDB, 2013), Neo4j (neo4j: World’s Leading Graph Database, 2012). Meijer and Bierman (2011) discuss the data model for NoSQL databases. Key features of NoSQL databases are non-adherence to the relational data model and horizontal scalability (NOSQL Databases, 2013).

Nowadays, there is a lot of research devoted to different fields of data management systems; also there is a lot of implemented software. Examples are the NoSQL graph database Neo4j (neo4j: World's Leading Graph Database, 2012) and InfiniteGraph (Home | Objectivity, 2013).

Jensen and Snodgrass (1999) discuss temporal database management systems, which are DBMSs with temporal support. Temporal databases represent not single snapshot of the state of affairs like traditional databases do; but they show database changes in intervals of time. There are such temporal data models as Temporally Oriented Data Model (Ariav, 1986). Query languages for temporal databases include TempSQL (Gadia, 1992), TSQL2 (Snodgrass et al., 1994), and others.

In temporal databases, there are two types of time attributes: validity time and transaction time. Validity time denotes the period when a fact is true with respect to the state of the UoD. Transaction time is the period when a fact is presented in the database. When both times, valid and transactional, are added to the table, it is called a bitemporal table. In a non-temporal relational database validity and transactional time intervals can be implemented as additional columns. Temporal database research discusses only determined periods of time; in cases in which the exact time of the event is not known, undetermined periods of time are necessary. Undetermined or fuzzy time intervals (when an exact date is not known) are discussed in the literature (Dyreson and Snodgrass, 1998; Gadia, Nair, and Poon, 1992; Nagypál and Motik, 2003; Schockaert and De Cock, 2008).

Databases used to store and query data related to objects in space are called *spatial databases*. Usually, spatial DBMSs offer special type systems capable of representing spatial data (Güting, 1994). Spatial databases may be used for storing geographical, geological, or geometrical data. There are raster and vector spatial data models.

Spatial databases with temporal support are called spatio-temporal databases (Erwig et al., 1999). Spatio-temporal databases may support queries about spatial properties or relationships, queries about temporal properties and relationships, and queries about spatio-temporal properties and relationships. Spatio-temporal databases may be used, e.g., in land information systems, databases containing data on migration of animals, databases of moving objects, or other contexts (Pelekis, Theodoulidis, Kopanakis, and Theodoridis, 2004).

Another type of database is the graph database, a database used for storage of network or graph data. There are different basic data structures that may be used to represent a graph as a data structure. These include an adjacency list, incidence list, adjacency matrix, and incidence matrix. Graph database may implement these data structures in a more efficient way, provide attributes for nodes or edges, and include various operations. Graph databases may be used in social networks or bioinformatics. There are graph database management systems such as Neo4j (neo4j: World's Leading Graph Database, 2012) and InfiniteGraph (Home | Objectivity, 2013).

The graph data model is also supported by Oracle (Network Data Model Overview, 2012). There are graph query languages such as GraphQL. Aggarwal and Wang, (2010, p.127) provide examples of graph queries:

- Find all heterocyclic chemical compounds that contain a given aromatic ring and a side chain.
- Find all co-authors from the DBLP dataset in a specified set of conference proceedings.
- Find all instances from an RDF [resource description framework] graph where two departments of a company share the same shipping company . . . .

Another example of a query is a reachability query, which shows whether one node is reachable from another node. Reachability queries and their efficient implementation are discussed in Yu and Cheng (2010).

He and Singh (2008) describe a formal language for graphs. The language consists of basic graph structures, called motifs, and operations on them such as concatenation, disjunction, and repetition.

Yan and Han (2010) describe graph indexing technologies. These are feature-based graph indexing methods, substructure searches, and approximate substructure searches.

4.4 Ontologies

Ontologies were first discussed in the context of Artificial Intelligence (AI). Gruber (1995, p. 908) tells that, "For AI systems, what 'exists' is that which can be represented". The structure of a system may be modelled formally using computational ontologies. Studer, Benjamins, and Fensel (1998, p. 25) adopt the definition from Gruber (1993) and say that "An ontology is a formal, explicit specification of a shared conceptualization", where *conceptualization* is defined by Gruber (1993, p.199) as "an abstract, simplified view of the world that we wish to represent for some purpose." "Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly" (Gruber, 1993, p.199). Ontology provides a shared vocabulary, including properties, concepts, and their definitions (Staab et al., 2011).

Pan (2009) describes a resource description framework (RDF). An RDF provides a data model for semantic annotation of the semantic web and is recommended by W3C (RDF - Semantic Web Standards, 2013). An RDF statement (triple) is of the form *subject property object*. An RDF graph is a set of RDF statements. RDF resources are represented by unified resource identifiers

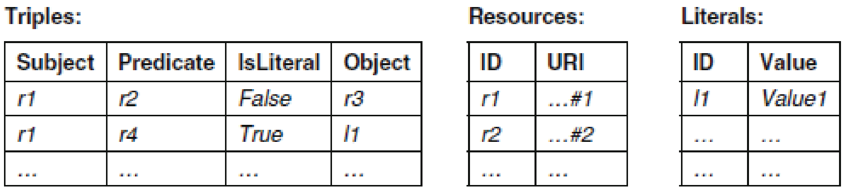
(URIs); and for representation of an RDF in computers, there is a special XML syntax. An RDF statement can provide meaning for the resources, where meaning may be taken from ontologies (Pan, 2009).

An RDF Schema (RDFS) is the means to express ontologies using RDF syntax. RDFS statements are RDF triples. An RDFS allows one to define types, classes, properties, their domains and ranges, and so on (RDF Vocabulary Description Language 1.0: RDF Schema, 2013).

Antoniou and Harmelen (2009) notice that the expression power of an RDF and RDFS is very limited—specifically, the scope of properties can be local, thus the range restrictions cannot be applied to certain classes only; classes cannot be declared as disjoint; Boolean definitions of classes are not allowed; cardinality restrictions cannot be set. In addition, there are several further limitations.

These limitations are overcome in web ontology language OWL. There are three versions: OWL-Full, OWL-DL, and OWL Lite. The latter versions are proper subsets of former versions. This means that all ontologies and valid conclusions expressible in OWL Lite are ontologies and valid conclusions in OWL-DL. The same holds for ontologies and valid conclusions expressible in OWL-DL and OWL-Full. On the other hand, OWL-Full allows phrases that cannot be expressed in OWL-DL or OWL-Lite and OWL-DL allows phrases that cannot be expressed in OWL Lite. OWL-Full is undecidable language. These languages are based on first-order description logic (Baader, Horrocks, and Sattler, 2009). Similarly, there are languages such as DAML (About DAML, 2013), OIL (Fensel, van Harmelen, Horrocks, McGuinness, and Patel-Schneider, 2001), and DAML+OIL (DAML+OIL, 2013). OWL is a successor of DAML+OIL. Successor language of OWL is OWL 2. OWL 2 adds new features such as keys, property chains, richer data types (OWL 2 Web Ontology Language, 2013).

There are several ways to store RDF data in relational databases. The first one is to create an individual DB schema for each ontology. Such schema would change each time concepts or their relationships in the ontology change. The next one is a generic store schema, which does not require restructuring when concepts or relationships change. Figure 11 shows generic schema for storage of RDF data, a normalized triple store. It may be used for any RDF data and described ontologies.



Triples:

Subject	Predicate	IsLiteral	Object
r1	*r2*	*False*	*r3*
r1	*r4*	*True*	*l1*
...	...	...	...

Resources:

ID	URI
r1	*...#1*
r2	*...#2*
...	...

Literals:

ID	Value
l1	*Value1*
...	...
...	...

FIGURE 11 Normalized triple store schema, Hertel, Broekstra, and Stuckenschmidt (2009, p. 492)

For modelling RDFSs, table triples (Figure 11) may be split into several tables such that each triple would model a separate RDFS property (SubConcept, SubProperty, PropertyDomain, etc. (Hertel et al., 2009)).

The W3C candidate recommended for querying an RDF is the SPARQL language (SPARQL Query Language for RDF, 2013). San Martín and Gutierrez (2009) describe the possibilities of representing social networks—consisting of actors, relationships between actors, and actors' attributes—with an RDF and its querying with SPARQL. According to San Martín et al. (2009), formally, a social network is a triple S = (V, E, L), where (V, E) is a directed graph and L is a set of labels (and labelling functions), specified as follows:

- $V = A \cup R \cup C$; V is the set of nodes where A is the set of actors, R is the set of relations, and C is the set of attributes. Each set is partitioned into families, actors are partitioned into families of actors A_i, and relations are partitioned into families of relations R_i.
- The set of edges $E = E_{AR} \cup E_{AC} \cup E_{RC}$ is a disjoint union of the following types of edges:
    - E_{AR} is a multiset of elements of $A \times R$ and
    - $E_{AC} \subseteq A \times C$ and $E_{AC} \subseteq A \times C$ are the sets of meaning edges.
- L is the union of sets of labels for the different types of edges, with corresponding labelling functions.

Then, San Martín et al. (2009) show that social networks can be modelled with an RDF. Next, an entire social network can be instantiated as RDF tuples and queried with SPARQL. Erétéo, Buffa, Gandon, and Corby (2009) show SPARQL queries that may be used to analyse social networking data stored in an RDF. Ereteo et al. (2011) discuss the possibilities of analysing social networks using ontologies such as Friend of a Friend (FOAF) and Semantically Interlinked Online Communities (SIOC). Modelling social networks with ontologies also is discussed by Mika (2005).

Gutierrez, Hurtado, and Vaisman (2007) describe the introduction of time labels into RDF statements, thus yielding temporal RDF graphs.

Borgo and Masolo (2009) describe foundational ontologies. These are ontologies that have a large scope, can be highly reusable in different modelling scenarios, and are philosophically well-founded and semantically transparent (Borgo et al., 2009, p. 361). Philosophical background is the theory of properties, which is discussed in (Armstrong, 1989; Mellor and Oliver, 1997).

Borgo et al. (2009) describe the foundational ontology DOLCE-CORE, which is an extension of Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE), described by Masolo, Borgo, Gangemi, Guarino, and Oltramari (2001). DOLCE aims at capturing the intuitive and cognitive bias underlying common-sense, while recognizing standard considerations and examples of linguistic nature (Borgo et al., 2009, p. 372).

Another example of foundational ontology is the general formal ontology (Herre, 2010). Core software ontology is described by Oberle, Grimm, and Staab

(2009). It reuses DOLCE ontology and introduces fundamental concepts of the software domain. Core ontology for media annotation (COMM) is described by Arndt, Troncy, Staab, and Hardman (2009). This is foundational ontology for media objects. Also, there are foundational ontologies for biological systems (Shah and Musen, 2009) and for cultural heritage (Doerr, 2009). Finnish general upper ontology (YSO) may be found at YSO - Semantic Computing Research Group (SeCo) (2013).

4.5 Social network analysis

Since a social network consists of actors and binary relations between them, it may be modelled as a complex network, or graph, in which actors of the social network would correspond to nodes in the graph and relations would correspond to the edges. Then, applying the methods from graph theory would become possible in social network data.

There is a lot of research on networks constructed from different data. For example, Jeong, Mason, Barabási, and Oltvai (2001), and Podani, Oltvai, Jeong, Tombor, Barabási, and Szathmáry (2001) analyse biological networks, Bascompte, Jordano, Melián, and Olesen (2003) analyse networks built from ecological data, and Glänzel and Schubert (2004) analyse networks of coauthorship of scientific papers. Goyal (2009) describes application of social network analysis to economics.

The first major developments in the field of social network analysis took place in the 1930s (Scott, 2000) in sociology (Parsons, 1949), psychology, and anthropology (Radcliffe-Brown, 1931). Milgram (1967) describes an experiment that led to the concept of "small world". It shows that networks, in which people represent the nodes and acquaintances between them represent the edges, at that time had an average shortest path length between random people equal to roughly six in the United States. Here, the path is the sequence of edges connecting two nodes in the network, and path length is measured as the number of edges of the path. The distance between two nodes is measured by the length of the shortest path. Leskovec and Horvitz (2008) show that the average distance between users of MSN Messenger is 6.6, and Réka Albert, Jeong, and Barabási (1999) show that the diameter of the WWW was equal to 19 at that time (diameter is the maximal shortest distance between any two nodes in the network).

Before the era of the Internet, analysis of social networks was time and resource intensive; however, nowadays, it has become much more convenient, especially after the emergence of social media sites. Backstrom, Boldi, Rosa, Ugander, and Vigna (2011) show that the average distance between two arbitrary users in Facebook was 4.74 ± 0.02 in May 2011. Also, networks formed from Italian, Swedish, and US users were studied. The average distance for the Italian subgraph was 3.89 ± 0.02, 3.90 ± 0.04 for Swedish subgraph, and 4.32 for the US subgraph.

Evolving random graphs are studied by Erdős and Rényi (1960). The authors present a model for generating random graphs in which edges between each pair of nodes may be added with equal probability. Barabási and Albert (1999) describe the preferential attachment property of the growing networks; i.e., the attachment of new nodes preferentially to the nodes having higher numbers of connections. An important finding is the power-law distribution of nodes' degrees (number of edges incident to the node) in complex networks; i.e., the variation of the fraction of the nodes having the same degree as the power of the degree. Networks with a degree distribution that follows the power law are referred to as *scale-free networks* (Barabási et al., 1999; Faloutsos, Faloutsos, and Faloutsos, 1999; Kleinberg, Kumar, Raghavan, Rajagopalan, and Tomkins, 1999; Newman, 2005).

Evolving social networks, which are typically studied as a series of snapshots of the network at different moments of time, have the property of shrinking over time the distance between the nodes (shrinking diameter). Leskovec, Kleinberg, and Faloutsos (2005) study the diameter and out-degree of large networks such as citation networks, affiliation networks, and networks built from the router communication networks. The authors provide a Forest Fire model, which captures heavy tailed in- and out-degrees, the densification power law, and shrinking diameter. The authors also show that networks are becoming denser over time; i.e., the average degree of nodes is increasing.

Girvan and Newman (2002) show that there is a high concentration of edges between the nodes within the groups and a low concentration between the groups. That suggests that some networks contain a community structure in which communities are groups of nodes in the network, tightly connected with each other. Also, the authors describe a community detection algorithm and apply it to computer generated networks and real data such as the Zachary karate club, a network built from plays of football clubs, scientific collaboration networks, and networks built from ecological data. Their algorithm detects the known communities with a high degree of success.

Liu (2011b, p.295), gives the definition for the community detection:

> given finite set of entities S = {s_1, . . . , s_n} of the same type, a community is a pair C = (T, G), where T is the community theme, and $G \subseteq S$ is the set of entities in S that shares theme T. Then, $s_i \in G$ is member of community C. Here entities are the nodes of the network.

There are a number of methods for community detection. In this case, the algorithm labels nodes of the graph corresponding to communities to which the nodes belong. Fortunato (2010, p. 8) stresses that communities can be identified only in sparse graphs; i.e., in graphs, where the number of edges is at most of the order of the number of nodes. Otherwise, the distribution of edges would be too homogeneous.

Figure 12 shows an example of a graph with community structure (there are three communities).

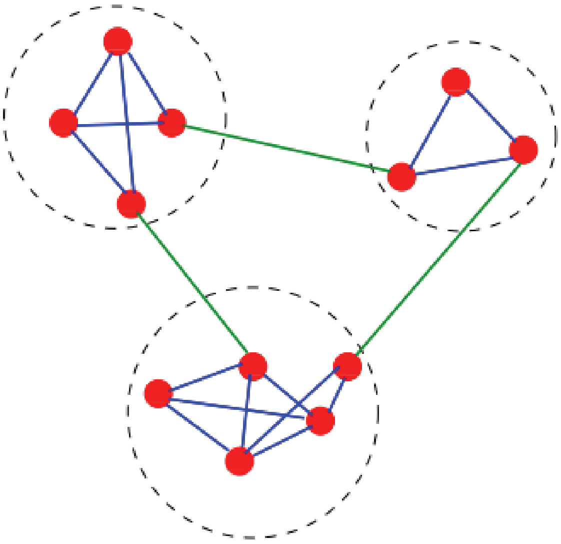


FIGURE 12 Communities (Fortunato and Castellano, 2007, p.2)

One of the quality functions for communities is the normalized cut (Shi and Malik, 2000):

$$Ncut(S) = \frac{\sum_{i\in S, j\in \bar{S}} A(i,j)}{\sum_{i\in S} degree(i)} + \frac{\sum_{i\in S, j\in \bar{S}} A(i,j)}{\sum_{j\in \bar{S}} degree(j)}$$

where S is a subgroup of N, A is the adjacency matrix, and A(i, j) is the weight of the edge between nodes i and j. Then, a normalized cut of a group of nodes S is the sum of weights of the edges that connect them to the rest of the graph, normalized by the total edge weight of S and the edge weight of the rest of the graph.

A related measure is conductance (Kannan, Vempala, and Veta, 2000). Conductance of a division of the graph into clusters is the sum of the normalized cuts of each of the clusters.

Newman and Girvan (2003) introduce the modularity concept:

$$Q = \sum_{c=1}^{k} \left[\frac{A(V_i, V_i)}{m} - \left(\frac{degree(V_i)}{2m} \right)^2 \right]$$

where V_i are the clusters, $A(V_i,V_i)$ is the sum of weights of the edges in V_i (Parthasarathy, Ruan, and Satuluri, 2011), m is the number of edges in the graph, $degree(V_i)$ is the total degree of cluster V_i ; i.e., the sum of the degree of nodes from V_i.

Maximization of the modularity and normalized cut is NP-hard (Brandes et al., 2007; Garey, Johnson, and Stockmeyer, 1976) and thus cannot be calculated quickly for large networks, if at all in reasonable time.

Algorithms for community detection include the Kernighan-Lin algorithm (Kernighan and Lin, 1970), divisive algorithm by Newman (2004), and spectral methods (Luxburg, 2007). Also, there are methods to detect overlapping communities (Palla, Derényi, Farkas, and Vicsek, 2005) where a single node can belong simultaneously to several communities. A survey of the performance characteristics of community detection algorithms may be found in Papadopoulos, Kompatsiaris, Vakali, and Spyridonos (2011).

In addition, there are algorithms for community detection in dynamic networks. Hopcroft, Khan, Kulis, and Selman (2004), analyse communities in several snapshots of a database of scientific papers. Communities in dynamic networks also were studied by Palla, Barabási, Vicsek, and Hungary (2007).

Leskovec, Lang, Dasgupta, and Mahoney (2009) apply community detection algorithms to over a hundred large datasets collected from real networks. They claim that the size of good communities is not more than approximately 100 nodes. Cai, Shao, He, Yan, and Han (2005) also studied community detection in multirelational networks; i.e., networks having several types of relations.

Kazienko, Musial, Kukla, Kajdanowicz, and Bródka (2011) discuss the analysis of multidimensional networks. Multidimensional networks are evolving networks having different types of relationships (however, attributes are not considered).

There also is a body of research about social influence. Sun and Tang, (2011) define *social influence* as the behavioural change of a person because of his or her perceived relationship with other people, organizations, and society in general. There are many different measures related to influence in social networks.

The tie strength between two nodes depends on the overlap of their neighbourhoods. A tie is strong when two nodes have more common neighbours. If the overlap between neighbourhoods is small, then the edge between two nodes is considered weaker (Sun et al., 2011). In case neighbourhoods to not overlap, the edge is called a local bridge (Granovetter, 1973).

There are several centrality measures. The simplest concept is *degree centrality*, which is equal to the degree of a node. It can be interpreted as a count of the number of paths that start from the node or end at the node. *Closeness centrality* measures the average shortest distance to all other nodes in the network. *Betweenness centrality* of a node x measures how many shortest paths between two arbitrary nodes a and b in the graph pass through node x. In addition, there is the *Katz centrality* measure, which counts all the nodes accessible from a given node, penalizing the distant nodes with an attenuation factor (Parthasarathy et al., 2011).

Many software packages implement computations of graph measures. These include NetworkX (Overview — NetworkX, 2013), a library written using Python; Pajek (start [Pajek Wiki], 2013) software, which has a number of input formats; and the powerful graph analysis tool Gephi (Gephi, an open source graph visualization and manipulation software, 2013), which contains a number of implemented graph algorithms and visualisation tools.

4.6 School shooting analysis

School shooting is defined by Semenov et al., (2010) as an event in which: (a) a student or a former student brings a gun, sword or similar weapon, or explosive/flammable liquid to school with the intent to kill somebody; (b) the gun is discharged or the weapon, liquid, or explosive is employed and at least one person is injured; and (c) the perpetrator attempts to shoot or otherwise kill more than one person, at least one of whom is not specifically targeted. That definition was adopted from Larkin (2009).

Böckler et al. (2011) provide an analysis of the increase in school-shooting incidents since 1956 (Figure 13).

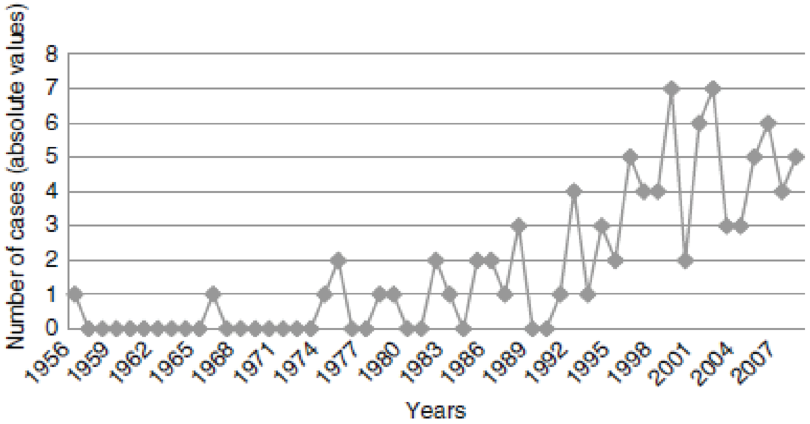


FIGURE 13 Frequency of incidents (Böckler et al., 2011, p.264)

Bondü and Scheithauer (2011) provide a hypothesis to explain the phenomenon. According to their hypothesis (see Figure 14), there are several risk factors that affect the behaviour of an individual during his developmental course. They also hypothesise that leaks of information occur, the correct interpretation of which may indicate that a person is on dangerous developmental path. A similar theory of the developmental path towards school shooting is described in Article I.

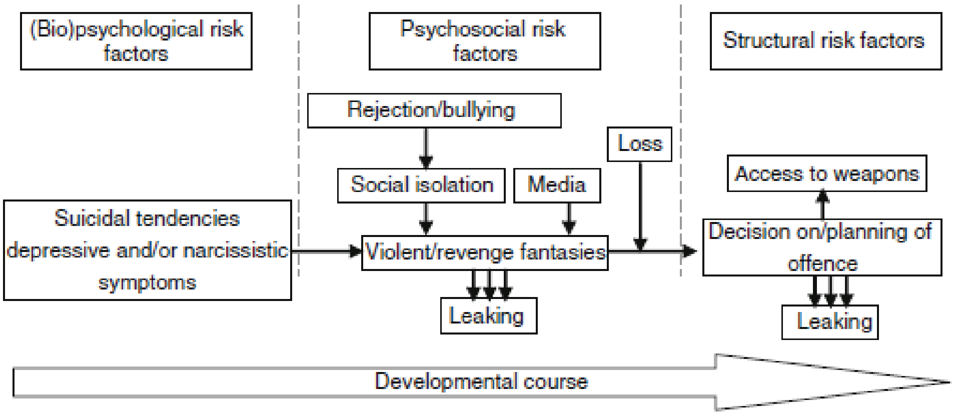

(Bio)psychological risk factors
Psychosocial risk factors
Structural risk factors
Rejection/bullying
Loss
Social isolation
Media
Access to weapons
Suicidal tendencies depressive and/or narcissistic symptoms
Violent/revenge fantasies
Decision on/planning of offence
Leaking
Leaking
Developmental course

FIGURE 14 Development path and risks (Bondü et al., 2011, p.304)

5 SUMMARY OF THE ORIGINAL ARTICLES

Seven papers total are included in the thesis; this chapter provides the summary of each.

5.1 Article I: ″Tracing potential school shooters in the digital sphere″

Veijalainen, J., Semenov, A., Kyppö, J., 2010. Tracing potential school shooters in the digital sphere. In Bandyopadhyay, S.K., Adi, W., Kim, T., Xiao, Y. (Eds.), 4th International Conference, ISA 2010, Miyazaki, Japan, June 2010, Proceedings, Communications in Computer and Information Science. Berlin Heidelberg: Springer, pp. 163–178. DOI: 10.1007/978-3-642-13365-7_16

5.1.1 Research problems

We describe the dependency between real school shootings and their traces in the Internet. There are more than 300 known school shootings, and frequently the perpetrators leave a suicide message or detailed plan on a forum or blog. The paper discusses the possibilities of following the traces of potential school shooters to avert an attack before it happens. It presents a probabilistic model of averting the attack, depending on environmental factors. The results of the research are used for requirements elicitation of the online monitoring systems.

5.1.2 Results

We analysed the history of school shootings and built a conceptual model of the developmental path of the offender. The model implies that there are developmental paths for all people and that each person has a specific path. The path is influenced by many factors, such as society or genes of the person. Some paths may lead people to criminal lives. During the developmental path, the person emits information that can be interpreted by other people and indicate which path person is on. We present a probabilistic model of carrying out a school-shooting attack based on the developmental path model. Paper also discusses false positives and false negatives. In addition, we presented high-level software architecture to analyse online information.

5.2 Article II: "Analysing the presence of school-shooting related communities at social media sites"

Semenov, A., Veijalainen, J., Kyppö, J (2010). Analysing the presence of school-shooting related communities at social media sites. International Journal of Multimedia Intelligence and Security (IJMIS), 1 (3), 232-268

5.2.1 Research problem

Paper hypothesises that all the school shooters since 2005 were active in the digital sphere and have similarities in their behaviour. We present case studies of 11 of the most prominent school shooters since 2005. Information emitted by the shooters is studied. The year 2005 was selected as a starting point because of the emergence of social media sites and growth of the number of school shooters.

5.2.2 Results

Paper suggests that, out of 11 school shooters, 8 were active in the digital sphere and 7 of those were active in social media sites. We present an ontology of the common behaviour that could characterize a potential school shooter. In addition, we analyse the definition of school shooters and distinguish them from other classes of criminals. In addition, we observe that the two major school shooters in Finland were connected to the YouTube social media site.

5.3 Article III: "A generic architecture for a social network monitoring and analysis system"

Semenov, A., Veijalainen, J., Boukhanovsky, A., 2011. A Generic Architecture for a Social Network Monitoring and Analysis System. In Barolli, L., Xhafa, F., Takizawa, M. (Eds.), The 14th International Conference on Network-Based Information Systems. Los Alamitos, CA, USA: IEEE Computer Society, pp. 178–185., DOI: 10.1109/NBiS.2011.52

5.3.1 Research problem

Paper describes the requirements and partial implementation for a generic social media monitoring system.

5.3.2 Results

We present requirements and the architecture of the system, which is able to carry out long-term monitoring of social media sites. Architecture of the system was implemented and, for evaluation purposes, a sub-network of the LiveJournal.com social media site was collected. The first collection took place in April 2011, when 890 000 profiles as well as their "friend" relations and interests information were collected. The networks consisting of nodes having school-shooting-related keywords were visualised by the external plugged-in software "Pajek".

5.4 Article IV: "Ontology-guided social media analysis, system architecture"

Semenov, A., Veijalainen, J., 2012. Ontology-guided social media analysis System architecture. In A. Maciaszek, L., Cuzzocrea, A., Cordeiro, J. (Eds.), ICEIS 2012 - Proceedings of the 14th International Conference on Enterprise Information Systems, Volume 2. Portugal: SciTePress, pp. 335–341., DOI: 10.5220/0004157303350341

5.4.1 Research problem

The authors take the advantage of the similarities in social media sites and discuss the architecture of a crawler that crawls social media sites based on their site ontologies. Collected data was put in a repository, the interface of which is derived from the temporal multirelational graph.

5.4.2 Results

Architecture of the crawler is presented. Ontologies are stored in internal storage. The crawler uses ontologies to parse web pages and extract data using XPath and regular expressions. Extracted data instantiate the ontology. A prototype of the crawler is developed using the Python programming language with a Twisted framework.

5.5 Article V: "A repository for multirelational dynamic networks"

Semenov, A., Veijalainen, J., 2012. A Repository for Multirelational Dynamic Networks. In Karampelas, P., Rokne, J. (Eds.), 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 1002–1005. DOI 10.1109/ASONAM.2012.174

5.5.1 Research problem

Paper describes the implementation of the interface for the repository, which implements the 2nd layer of the model—a temporal multirelational graph. The scheme of the relational database, which is used as data management system, is discussed.

5.5.2 Results

Paper presents an interface for the repository and architecture of the DB scheme, implemented as a temporal database. We compare the snapshot approach and the temporal approach, showing that the temporal approach requires less space to store the evolving multirelational graph. An evaluation is done using a simulation of the data-insertion software.

5.6 Article VI: "A modeling framework for social media monitoring"

Semenov A., Veijalainen J. (2012). A modeling framework for social media monitoring, accepted for publication at International Journal of Web Engineering and Technology, 32 pp.

5.6.1 Research problem

Currently, there are about 7 billion people in the world. The Internet penetration rate varies by country; thus, citizens of some countries are represented online, and specifically in social media, more frequently than citizens of other countries. In addition to that, social media sites have a lot of relationships that do not necessarily correspond to any relationships in immediate social life (e.g., whether friends in real life are friends in Facebook or followers in Twitter can only occur after the platforms were put in operation). Also, fake profiles exist in social media sites that could not exist in the immediate social reality. Information about people and their relationships in the real world evolves. This also happens at the social media sites.

There are many different social media sites. The paper studies how the content of social media sites models the immediate social reality and whether it is possible to model all social media sites with a single model. A homogenous model for the social media sites would allow application of the same software for monitoring and analysis different sites.

5.6.2 Results

The authors introduce a three-level model in which the immediate social reality is modelled by the social media, which is modelled by a temporal multirelational graph, which is modelled by its representation in the DB.

The researchers describe in detail the process of elaboration of the model: at first, differences between the immediate social reality and social media are analysed; then site ontologies of the popular social media sites are compared and generalised to a temporal multirelational graph; finally, a DB schema is presented to store the graph. The site ontology must be presented at the monitoring site so that it can be crawled.

The paper also contains an analytical estimation of the limits of the monitoring systems and a description of a database schema for the implementation of the repository. In addition, the writers present a system of the definitions of the terms used through the thesis. They study the possibilities of relating the results of the monitoring with the data from the immediate social reality.

5.7 Article VII: “Analysis of services in Tor network”, Finnish segment

Semenov A. (2013). Analysis of services in Tor network. Accepted to 12th European Conference on Information Warfare and Security ECIW-2013, Jyväskylä, Finland. 11 pages.

5.7.1 Research problem

The Tor network was built to allow stronger anonymity in the Internet and primarily targeted the US military. Tor uses onion routing. Messages sent there travel from source to destination through a random number of nodes and are rerouted at each node. In addition, messages are re-encrypted at each node using an asymmetric encryption scheme. All in all, this allows secure data transfer and strongly protects the privacy of the users.

A Tor network provides the possibility of maintaining *hidden services*, which are Tor clients, running server software. Hidden services are accessed through the .onion pseudo top-level domain zone, and their names are generated automatically. Because of strong anonymity, some hidden services store content, which is illegal in some jurisdictions, and provide platforms for various illicit activities; e.g., trading controlled substances. A quantitative analysis of hidden services, maintained in the Finnish language, was carried out for this paper.

5.7.2 Results

In the present paper, URL addresses of hidden services were extracted from the well-known hidden service “Hidden Wiki”. There are seven services; however, only two of them act as active discussion boards: Thorlauta imageboard and Suojeluskunta forum.

First, the author describes the possibilities of crawling Tor hidden services. Then, the whole contents of the two mentioned services are crawled. During the collection of Thorlauta, 792 MB of data were downloaded. This lasted 5 hours and 15 minutes (with an average download rate of 343.28 Kbit/s). Suojeluskunta contained 40 MB of data, which were downloaded in 96 minutes (with an average speed of 56.89 Kbit/s).

It was found that most popular topics on Thorlauta are related to purchasing controlled substances with BitCoin currency. The top Suojeluskunta threads contained various hate speeches with white supremacy implications (e.g., discussion of “ethnic cleanings” in Oulu). In addition, the topic of purchasing arms in Thorlauta was among the top-ten most popular topics.

Moreover, the writer discusses the performance of the repository used in the system. The researcher shows the performance of 1-Hop, 2-Hop, and 3-Hop queries run on small datasets from the mentioned forums (their execution time is around 2ms); however, for the earlier collected dataset from LiveJournal.com, the 3-hop query completes in 304 seconds.

5.8 About joint publications

The thesis introduction was written solely by the author. Articles I–VI were written in coauthorship.

In Article I, the author of this thesis was the second author. His contribution was in analysing the technologies that can be applied to detect school shooters by analysing data that they leaked online. His contribution also was in developing and describing the preliminary architecture of the monitoring and analysis system.

The author contributed to Article II by analysing school-shooting cases, gathering and analysing online data about school shooters from various online sources, performing computational analysis of school-shooting communities, and developing an ontology with keywords used in those communities. These parts of the Article II were written by the author.

The author contributed to Article III by developing requirements and the architecture as well as by implementing the system’s prototype. These aspects are the main issues of this paper and were written by the author.

The author was the first author of Article IV; his contribution was in elaboration of the architecture of the crawler, implementing the prototype, and describing this in the paper. In Article V the author developed database schemas, interface, and estimated the growth of the size of the databases. He was describing these aspects in the article.

The author's role in Article VI was to analyse the ontologies of social media sites. Also, the author contributed to the development of the 2nd and 3rd levels of the models.

Article VII was written solely by the author.

6 DISCUSSION

The present section contains discussion of overall results of the thesis. Chapter presents discussion of the issues related with privacy while using software for social media monitoring and analysis. Also, chapter contains description of similar software made outside of Academia.

6.1 Privacy

A major issue in the field of social media monitoring is privacy. People put a lot of data about themselves in social media, and individuals can perceive losing privacy as a threatening experience (Walther, 2011). Alan Westin (1967, p.7) defined privacy as "the claim of individuals, groups or institutions to determine for themselves when, how, and to what extent information about them is communicated to others." Altman (1975, p. 24) describes privacy as "the selective control of access to the self". Alan Westin (1967) describes four functions of privacy: personal autonomy, emotional release, self-evaluation, and limited and protected communication. A survey of research about online privacy may be found in Walther (2011).

The software described in this thesis to monitor and analyse social media was used to collect public data only. Semenov et al., (2012b) describe that digital data can be classified into public, semipublic, and private.

However, issues with privacy are related not only to ethics of research but also to various laws and terms of use of the web sites, which may prohibit data collection. The developed software was used after detailed consultations with lawyers and in accordance with the resulting statement of person registry, ("Rekisteriseloste"), which states that collected data may be used for research purposes only and may not be handed anywhere. The statement may be found at Rekisteriseloste (2012). The Personal Data Act of Finland, which states that personal data may be used for research purposes (§12, §14), may be found at

(FINLEX ® - Ajantasainen lainsäädäntö: 22.4.1999/523, 1999). European Union data protection directive may be found at (EUR-Lex - 31995L0046 - EN, 2013).

6.2 Existing software

There is a number of commercial business intelligence tools; a comparison of them may be found at, e.g., (Social media monitoring tools comparison guide, 2013). Also, during the preparation of the present thesis, one of the commercial solutions, RecordedFuture (Recorded Future - Unlock The Predictive Power Of The Web, 2012), was examined by the author. RecordedFuture provides service aiming at monitoring news sites and social media on specific topics (given as keywords). It concentrates on monitoring large number of sources, extracts temporal information such as dates of the events (e.g. mentioning of certain topic in the news), and visualizes the information.

Information from the social media sites may also be collected by using such software as iMacros (iMacros, 2013); which allows automation of existing browsers with scripting language. Scripting language contains commands for extracting and saving the information from the web-pages. Since it uses existing browser, it has possibilities for extraction of Javascript enabled pages, however scripting language is not very powerful. Similar to iMacros software which was found but not tested is Browser Automation Studio (Browser Automation Studio, 2013).

Also, there are crawler generation frameworks, such as Scrapy (An open source web scraping framework for Python, 2012). However, working with it requires knowledge of Python programming language. Also, it does not provide any possibilities for analysis of data, since it is only crawler.

The approach taken in this thesis is different in number of ways from that existing software. First, except for the provision of the business intelligence software as a service, the goal of the thesis was to develop a theoretical framework. Second, the software developed for this thesis allows monitoring of just certain sites, a description of which may be given by the user. An example is the hidden services in Tor, but not the entire world of social media. In the future, the developed software shall be scaled and enhanced with automatic wrapper generation. Then it could become similar to a commercial product.

7 CONCLUSIONS

The present chapter contains contributions of the thesis, its limitations, and further research.

7.1 Contributions

The present thesis discusses the principles of social media monitoring and analysis software.

The first contribution is a novel three-layer modelling framework for social media monitoring. The present framework establishes the relations among (1) the immediate social reality, its 1st level model, the content of social media sites; (2) a temporal multirelational graph, which models the content of social media sites and is the 2nd level of the model; and (3) a repository that comprises the 3rd level.

The parts of this model were, to some extent, developed by the researchers, as shown in the state of the art; however, the entire modelling framework is new. The thesis describes the novel use of ontologies to model online social networks along the contents of social media sites. The framework addresses the modelling issues that must be solved by any remote monitoring and analysis software. It also guides the construction of a single software system that allows monitoring and analysis of all social media sites.

The second contribution is the analysis of the requirements and development of an architecture for a generic social media monitoring and analysis system. The system consists of a crawler, repository, and analysis module. It allows the monitoring of social media sites, the description of which is represented in ontologies. Detailed architectures for the crawler and repository are presented in the thesis. The thesis also provides performance measures for the software, including analytically estimated theoretical limits for the monitoring activity in general.

The third contribution is a probabilistic model, which models the probability of averting school-shooting attacks, depending on a number of factors. False positives and false negatives are also considered.

The fourth contribution was the result of the analysis of real online data. A journal article analyses online traces left by 11 major school shooters at social media sites between 2005 and 2011. It was found out that, in 7 cases, traces were found and their correct interpretation could have facilitated the prevention of the shootings.

The thesis also analyses the Finnish segment of Tor network. The collection targets of the developed software are especially Thorlauta and Suojeluskunta. A quantitative analysis of the message intensity and some content analysis is presented. Results of the analysis are presented in the **article VII**; that is the fifth contribution.

7.2 Limitations

The research in this thesis contains a number of limitations. The performance of the monitoring and analysis software has not been systematically tested. Especially, the access rate to the repository during analysis tasks is unknown and the reachable crawling and processing rates are not systematically collected. For the above reasons, a comparison of the performance of this system with other similar systems such as (San Martín et al., 2009) has not been done.

During the software development, it was discovered that a large number of social media sites are implemented using various pieces of client-side code, such as JavaScript relying on AJAX. So far, how to crawl such sites (including Facebook) was not satisfactorily solved.

Another limitation is the use of (site and crawling) ontologies inside the wrapper. Currently, a manual approach is used, although there are many automatic and semi-automatic methods.

The next limitation is lack of calibration of the probabilistic models which estimate the probability of detention of potential school shooters based on leaked data.

7.3 Further research

Further, the repository building approach developed for this thesis may be combined with a temporal RDF, as described by Gutierrez et al. (2007). Also, foundational ontologies may be used. Then, existing RDF-based software may be used. Also, repository performance should be systematically tested. After that, efficient storage methods, such as graph indexing, may be applied. The next direction of this research is to apply large data processing tools like Hadoop to analyse the gathered data.

Next direction of further research is crawling JavaScript enabled social media sites. Perhaps the best approach for handling the code would be to integrate a JavaScript interpreter into the wrapper of the crawler, or integrate an entire browser code into it. This should correctly process JavaScript and work properly with DOM.

In the future, it would be necessary to compare methods of wrapper generation, and implement the most suitable of them. This enhancement would make the software a convenient tool for social media analysis for people who are not experts in regular expressions or other similar technologies. One of the possibilities is to consider template languages for generation of HTML pages, used in web frameworks, like Chameleon ZPT (Chameleon, 2013), or Mako (Mako, 2013). Similar technologies may be used for extraction of the data.

Another direction of further research is in the field of user interface (UI). Currently, the software has a prototypical UI, developed in Python, using Pyramid web framework, and Twitter Bootstrap. A proper research is, however, needed to figure out the ordinary users' needs. That will lead to extensive testing of the system by number of users, which may be seen as one form of validation of the product.

Further, the probabilistic model assessing school shooter detection probability presented in the thesis should be calibrated based on real data concerning school shooters.

YHTEENVETO (FINNISH SUMMARY)

Sosiaalisen median seuranta- ja analysointiohjelmiston suunnittelu- ja toimintaperiaatteet

Miljardit ihmiset käyttävät nykyään Internettiä päivittäin työtehtävissään ja vapaa-aikoinaan. Hyvin nopeasti Internetissä ovat lisääntyneet erityisesti sosiaalisen median sivustot, kuten Facebook, Twitter, Vkontakte, jne. Useilla niistäkin on kymmeniä tai satoja miljoonia käyttäjiä ja Facebook on päässyt jo miljardin käyttäjän rajan yli. Nämä sivustot tukevat erilaisten virtuaaliyhteisöjen syntyä ja toimintaa. Virtuaaliyhteisö terminä on peräisin Howard Rheingoldin julkaisuista 1990-luvun alusta. Virtuaaliyhteisö on ryhmä ihmisiä, joka käyttää tieto- ja viestintäteknologiaa aluksi sanomien, myöhemmin multimediasisältöjen tuottamiseen ja jakamiseen verkossa. Nykyisellään pääosa virtuaaliyhteisöistä syntyy ja toimii sosiaalisen median sivustoilla, joista jotkut tallentavat käyttäjien tuottaman sisällön vuosiksi, jotkut vain päiviksi. Sosiaalisen median sivustot tukevat multimediasisällön ohella erityisesti ihmisten välisten suhteiden esittämistä ja suljettujen tai avointen ryhmien muodostamista. Digitaalinen sisältö ja ihmisten välisten suhteiden esittäminen bittijonoina mahdollistavat sivustojen automaattisen seurannan ja analysoinnin. Sikäli kun sivustoilla on julkisesti saatavissa totuudenmukaista informaatiota ihmisten elämästä, toiveista, pyrkimyksistä, inhon kohteista ja niin edelleen, sivustojen seuranta ja analysointi voi paljastaa monenlaisia seikkoja. Keskeinen havainto työssä on ollut, että sosiaalisen median sivustoja pitäisi seurata jatkuvasti ja pitkän aikaa, jopa vuosia, ja näin muodostaa kuva virtuaaliyhteisöjen ja niissä toimivien yksilöiden ajattelun ja toiminnan kehityksestä, jos halutaan ennakoida erilaisia kehityskulkuja. Informaation tallentaminen monitorointijärjestelmään on hyödyllistä myös siinä mielessä, että useat sosiaalisen median sivustot eivät säilytä sisältöjä ja virtuaaliyhteisöjen vanhoja rakenteita lainkaan tai vain lyhyen aikaa ja näin virtuaaliyhteisöjen kehitysvaiheiden ja lainalaisuuksien analysointi ei ilman informaation tallennusta sivustojen ulkopuolelle ole lainkaan mahdollista. Lisäksi informaation tallentaminen eri sivustoilta samaan tietovarastoon on välttämätöntä, jos halutaan tutkia eri sivustoilla toimivien yksilöiden ja virtuaaliyhteisöjen rinnakkaista kehitystä ja suhteita.

Työn haasteet ja kontribuutiot liittyvät edellisiin tavoitteisiin. Väitöskirjassa kehitetään ja analysoidaan sosiaalisen median seuranta- ja analyysiohjelmistoon liittyviä vaatimuksia, rajoitteita, yleisiä mallinnusperiaatteita ja arkkitehtuuria. Työssä on myös toteutettu prototyyppijärjestelmä, joka automaattisesti seuraa ja analysoi sosiaalisen median sivustoja.

Ohjelmistotuotannon näkökulmasta työssä on seurattu Boehmin vuonna 1986 esittämää spiraalimallia, jossa ohjelmiston kehitys etenee prototyyppiversiosta toiseen evaluoinnin ja uusien vaatimusten määrittäessä seuraavan prototyyppiversion toiminnallisuuden. Ensimmäinen versio laadittiin kouluammuskelijoiden verkkoon jättämän informaation löytämiseksi ja analysoimiseksi. Tässä yhteydessä kehitettiin ensimmäinen versio järjestelmäarkkitehtuurista.

Seuraavassa vaiheessa työn kohdetta laajennettiin yleensä sosiaalisen median sivustojen monitorointiin ja analysointiin. Tällöin tultiin siihen tulokseen, että järjestelmän perustana toimii kolmitasoinen mallinnus. Alimpana tasona ovat sosiaalisen median sivustojen ontologiat, eli käsitteet ja niiden väliset suhteet, joita sivusto tarjoaa; esim. "ystävä", "seuraaja", "tviitti", "ryhmä", jne. Seuraava taso on temporaalinen moniverkko, jonka avulla kaikki sivustojen erilaiset ontologiat ja aikaulottuvuus voidaan mallintaa. Kunkin sivuston ajallisesti kehittyvät sisällöt ja virtuaaliyhteisöt ovat tällaisen temporaalisen moniverkon instansseja. Kolmas taso muodostuu em. moniverkon kaavasta tietokannassa. Sivustoilta kerätty data tallennetaan tämä kaavan mukaisesti. Yksi työn tulos on, että tällainen kolmitasoinen mallinnus on aina toteutettava tämän kaltaisessa monitorointi- ja analyysiohjelmistossa.

Lisäksi väitöskirjassa kehitetään hakurobotin arkkitehtuuri, joka perustuu sivustojen ontologioihin. Temporaalisen moniverkon manipulointia tietovarastossa tukeva ohjelmointirajapinta on myös yksi tulos. Tämä voidaan toteuttaa joko relaatiokannan avulla, kuten prototyypissä on tehty, tai esim. graafitietokannan avulla. Toteutus voi myös periaatteessa perustua useampaan kannanhallintajärjestelmään samanaikaisestikin.

Yksi tulos on tällaisen monitorointi- ja analyysiohjelmiston periaatteellisen suorituskyvyn rajat. Ne asettaa yhtäältä monitoroivan palvelimen ja monitoroitavan sivuston välisen kanavan siirtokapasiteetti, monitoroidulla sivustolla tapahtuvien muutosten määrä aikayksikössä niissä virtuaaliyhteisöissä, joita halutaan seurata, ja ohjelmiston suorituskyky tallennettaessa haettua dataa tietovarastoon. Useat sivustot asettavat tarjoamilleen hakurajapinnoille suorituskykyrajoja, jolloin siirtokanavan kapasiteetti määräytyy oleellisesti niiden kautta.

Spiraalimallia seurattaessa syntyneitä prototyyppiversioita on käytetty informaation keräämiseen sosiaalisen median sivustoilta ja näin saatu kokemuksia ja ideoita seuraavaan versioon. Tässä työssä analysoitiin ääriesimerkkinä maailmassa paljon huomiota saaneita kouluammuskelijoita ja heidän jättämiään jälkiä erilaisille Internet-sivustoille. Kohteena olivat vuoden 2005 jälkeen tapahtuneet hyökkäykset. Analyysillä pyrittiin selvittämään, olisiko rikokset ollut mahdollista estää näiden ammuskelijoiden Internetistä vuonna 2011 vielä löytyneiden jälkien perusteella. Tulos oli, että 7 näistä 11 ammuskelijasta oli jättänyt jollekin sivustolle vähän ennen hyökkäystä joko täysin selvän viestin hyökkäyksestä tai ainakin melko selvän. Työssä kehitettiin myös todennäköisyyksiin nojaava kvantitatiivinen malli, jolla yksilön kehitystä ja hänen eri kehitysvaiheissaan Internetiin ja muuhun ympäristöön jättämää informaatiota ajatuksistaan ja mahdollisista ongelmistaan pyritään mallintamaan. Keskeinen kysymys on, voidaanko tällaisen julkisen tai puolijulkisen informaation pohjalta automaattisesti tai edes puoliautomaattisesti päätellä ja jopa estää mahdollisia kouluammuskelun kaltaisia rikoksia. Näistä pohdinnoista johdettiin monitorointi- ja analysointiohjelmistolle vaatimuksia. Tarkemman vastauksen antaminen em. kysymykseen vaatii jatkotutkimusta.

Ohjelmiston viimeistä versiota on käytetty, kun koko Livejournal.com -sivuston julkinen sisältö on kerätty ohjelmiston tietovarastoon. Työssä raportoidaan myös tulokset puolijulkisen TOR-verkon suomenkielisiin sivuihin kohdistetusta informaation keräyksestä ja perusanalyysistä.

Työ koostuu englanninkielisestä johdannosta ja seitsemästä artikkelista, joista kuusi on vertaisarvioitu ja julkaistu ja seitsemäs on arvioitavana konferenssiin.

REFERENCES

About DAML [WWW Document], 2013. URL http://www.daml.org/about.html (accessed 12.2.13).

Aggarwal, C.C., Al-Garawi, F., Yu, P.S., 2001. Intelligent crawling on the World Wide Web with arbitrary predicates. In Y. Shen, V., Saito, N., R. Lyu, M., Zurko, M.E. (Eds.), Proceedings of the 10th International Conference on World Wide Web, WWW '01. New York, NY, USA: ACM, pp. 96–105.

Aggarwal, C.C., Wang, H. (Eds.), 2010. Managing and Mining Graph Data, 1st ed. Berlin, Heidelberg: Springer.

Alan Westin, 1967. Privacy and Freedom. New York, NY, USA: Atheneum.

Aliprandi, C., Marchetti, A., 2011. Introducing CAPER, a Collaborative Platform for Open and Closed Information Acquisition, Processing and Linking. In Stephanidis, C. (Ed.), HCI International 2011 – Posters' Extended Abstracts. Berlin, Heidelberg: Springer, pp. 481–485.

Altman, I., 1975. The Environment and Social Behavior: Privacy, Personal Space, Territory, and Crowding. Monterey, California: Brooks/Cole Publishing Company.

Amazon to Introduce Web-Based Book Previews [WWW Document], 2013. URL http://bits.blogs.nytimes.com/2010/06/30/amazon-to-launch-web-based-book-previews/ (accessed 27.1.13).

Ameripour, A., Nicholson, B., Newman, M., 2010. Conviviality of Internet social networks: An exploratory study of Internet campaigns in Iran. Journal of Information Technology 25(2), 244–257.

An open source web scraping framework for Python [WWW Document], 2012. URL http://scrapy.org/ (accessed 7.5.12).

Antoniou, G., Harmelen, F. van, 2009. Web Ontology Language: OWL. In Staab, S., Studer, R. (Eds.), Handbook on Ontologies, International Handbooks on Information Systems. Berlin Heidelberg: Springer, pp. 91–110.

Apache CouchDB [WWW Document], 2013. URL http://couchdb.apache.org/ (accessed 10.2.13).

Ariav, G., 1986. A temporally oriented data model. ACM Trans. Database Syst. 11(4), 499–527.

Armstrong, D.M., 1989. Universals: An Opinionated Introduction, Second Impression. ed. USA: Westview Press.

Arndt, R., Troncy, R., Staab, S., Hardman, L., 2009. COMM: A Core Ontology for MultimediaAnnotation. In Staab, S., Studer, R. (Eds.), Handbook on Ontologies, International Handbooks on Information Systems. Berlin Heidelberg: Springer, pp. 403–421.

Baader, F., Horrocks, I., Sattler, U., 2009. Description Logics. In Staab, S., Studer, R. (Eds.), Handbook on Ontologies, International Handbooks on Information Systems. Berlin Heidelberg: Springer, pp. 21–43.

Bachman, C.W., Batchelor, R.E., Beriss, I.M., Blose, C.R., Burakreis, T.I., Valle, V.D., Dodd, G.G., Helgeson, W., Lyon, J., Metaxides, A. (Tax), McKinzie, G.E., Siegel, P., Simmons, W.G., Sturgess, L.L., Tellier, H., Weinberg, S.B., Werner, G.T., 1969. Data base task group report to the CODASYL programming language committee, October 1969. New York, NY, USA: ACM.

Backstrom, L., Boldi, P., Rosa, M., Ugander, J., Vigna, S., 2011. Four Degrees of Separation. In S. Contractor, N., Uzzi, B., W. Macy, M., Nejdl, W. (Eds.), Web Science 2012. New York, NY, USA: ACM, pp. 33–42.

Bancilhon, F., 1996. Object databases. ACM Comput. Surv. 28(1), 137–140.

Barabási, A.-L., Albert, R., 1999. Emergence of Scaling in Random Networks. Science 286(5439), 509–512.

Bascompte, J., Jordano, P., Melián, C.J., Olesen, J.M., 2003. The nested assembly of plant–animal mutualistic networks. PNAS 100(16), 9383–9387.

Batsakis, S., Petrakis, E.G.M., Milios, E., 2009. Improving the performance of focused web crawlers. Data & Knowledge Engineering 68(10), 1001–1013.

Benington, H.D., 1983. Production of Large Computer Programs. Annals of the History of Computing 5(4), 350–361.

Bergholz, A., Childlovskii, B., 2003. Crawling for domain-specific hidden Web resources. In Santucci, G., Klas, W., Bertolotto, M., Calero, C., Baresi, L. (Eds.), Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003. Washington, DC, USA: IEEE Computer Society, pp. 125 – 133.

Bertot, J.C., Jaeger, P.T., Hansen, D., 2012. The impact of polices on government social media usage: Issues, challenges, and recommendations. Government Information Quarterly 29(1), 30–40.

Blanco, L., Bronzi, M., Crescenzi, V., Merialdo, P., Papotti, P., 2010. Exploiting information redundancy to wring out structured data from the web. In Rappa, M., Jones, P., Freire, J., Chakrabarti, S. (Eds.), Proceedings of the 19th International Conference on World Wide Web, WWW '10. New York, NY, USA: ACM, pp. 1063–1064.

Boehm, B., 1986. A spiral model of software development and enhancement. SIGSOFT Softw. Eng. Notes 11(4), 14–24.

Bondü, R., Scheithauer, H., 2011. Explaining and Preventing School Shootings: Chances and Difficulties of Control. In Heitmeyer, W., Haupt, H.-G., Malthaner, S., Kirschner, A. (Eds.), Control of Violence. New York, NY, USA: Springer, pp. 295–314.

Borgo, S., Masolo, C., 2009. Foundational Choices in DOLCE. In Staab, S., Studer, R. (Eds.), Handbook on Ontologies, International Handbooks on Information Systems. Berlin Heidelberg: Springer, pp. 361–381.

Boyd, D., Ellison, N., 2007. Social Network Sites: Definition, History, and Scholarship. Journal of Computer-Mediated Communication 13(1), 210–230.

Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., Wagner, D., 2007. On Finding Graph Clusterings with Maximum Modularity Graph-Theoretic Concepts in Computer Science. In Brandstädt, A., Kratsch, D., Müller, H. (Eds.), Graph-Theoretic Concepts in Computer Science. Berlin, Heidelberg: Springer, pp. 121–132.

Brin, S., Page, L., 1998. The anatomy of a large-scale hypertextual Web search engine. Comput. Netw. ISDN Syst. 30(1-7), 107–117.

Bronzi, M., Crescenzi, V., Merialdo, P., Papotti, P., 2011. Wrapper Generation for Overlapping Web Sources. In Boissier, O., Benatallah, B., Papazoglou, M.P., Ras, Z.W., Hacid, M.-S. (Eds.), 2011 IEEE/WIC/ACM International Conference On Web Intelligence and Intelligent Agent Technology (WI-IAT). Washington, DC, USA: IEEE Computer Society, pp. 32 –35.

Browser Automation Studio [WWW Document], 2013. URL http://browserautomationstudio.com/ (accessed 8.5.13).

Böckler, N., Seeger, T., Heitmeyer, W., 2011. School Shooting: A Double Loss of Control. In Heitmeyer, W., Haupt, H.-G., Malthaner, S., Kirschner, A. (Eds.), Control of Violence. New York, NY, USA: Springer, pp. 261–294.

Cai, D., Shao, Z., He, X., Yan, X., Han, J., 2005. Community Mining from Multi-relational Networks. In Jorge, A.M., Torgo, L., Brazdil, P., Camacho, R., Gama, J. (Eds.), Knowledge Discovery in Databases: PKDD 2005. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 445–452.

Chakrabarti, S., van den Berg, M., Dom, B., 1999. Focused crawling: a new approach to topic-specific Web resource discovery. Comput. Netw. 31(11-16), 1623–1640.

Chamberlin, D.D., Astrahan, M.M., Blasgen, M.W., Gray, J.N., King, W.F., Lindsay, B.G., Lorie, R., Mehl, J.W., Price, T.G., Putzolu, F., Selinger, P.G., Schkolnick, M., Slutz, D.R., Traiger, I.L., Wade, B.W., Yost, R.A., 1981. A history and evaluation of System R. Commun. ACM 24(10), 632–646.

Chameleon [WWW Document], 2013. URL http://chameleon.readthedocs.org/en/latest/ (accessed 8.5.13).

Chaudhri, A.B., Rashid, A., Zicari, R. (Eds.), 2003. Xml Data Management: Native Xml and Xml-Enabled Database Systems. Boston, MA: Addison-Wesley Professional.

Cheong, F.-C., 1996. Internet agents: spiders, wanderers, brokers, and bots. Indianapolis, IN, USA: New Riders Publishing.

Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Merson, P., Nord, R., Stafford, J., 2010. Documenting Software Architectures: Views and Beyond, 2nd ed. Boston, MA, USA: Addison-Wesley Professional.

Codd, E.F., 1970. A relational model of data for large shared data banks. Commun. ACM 13(6), 377–387.

Cohen, W.W., Hurst, M., Jensen, L.S., 2002. A Flexible Learning System for Wrapping Tables and Lists in HTML Documents. In Yih-Fam, R.C., Kovacs, L., R. Lyu, M., Lawrence, S. (Eds.), Proceedings of the 11th International Conference on World Wide Web. New York, NY, USA: ACM, pp. 232–241.

Couchbase | Simple, Fast, Elastic NoSQL Database [WWW Document], 2013. URL http://www.couchbase.com/ (accessed 10.2.13).

DAML+OIL [WWW Document], 2013. URL http://www.w3.org/Submission/2001/12/ (accessed 12.2.13).

Davison, B.D., 2000. Topical Locality in the Web. In Belkin, N.J., Ingwersen, P., Leong, M.-K. (Eds.), Proceedings of the 23rd Annual International Conference on Research and Development in Information Retrieval (SIGIR 2000). New York, NY, USA: ACM Press, pp. 272–279.

Doan, S., Ohno-Machado, L., Collier, N., 2012. Enhancing Twitter Data Analysis with Simple Semantic Filtering: Example in Tracking Influenza-Like Illnesses. In Ohno-Machado, Lucila, Jiang, X. (Eds.), 2012 IEEE Second International Conference on Healthcare Informatics, Imaging and Systems Biology (HISB). California, USA: IEEE Computer Society, pp. 62 –71.

Doerr, M., 2009. Ontologies for Cultural Heritage. In Staab, S., Studer, R. (Eds.), Handbook on Ontologies, International Handbooks on Information Systems. Berlin Heidelberg: Springer, pp. 463–486.

Domain Counts & Internet Statistics | Whois Source [WWW Document], 2013. URL http://www.whois.sc/internet-statistics/ (accessed 27.1.13).

Dong, H., Hussain, F.K., Chang, E., 2008. A survey in semantic web technologies-inspired focused crawlers. In Pichappan, P., Abraham, A. (Eds.), Third International Conference on Digital Information Management, 2008. ICDIM 2008. Washington, DC, USA: IEEE Computer Society, pp. 934 –936.

Duda, C., Frey, G., Kossmann, D., Matter, R., Zhou, C., 2009. AJAX Crawl: Making AJAX Applications Searchable. In Ioannidis, Y.E., Lun Lee, D., Ng, R.T. (Eds.), Proceedings of the 2009 IEEE International Conference on Data Engineering, ICDE ’09. Washington, DC, USA: IEEE Computer Society, pp. 78–89.

Dyreson, C.E., Snodgrass, R.T., 1998. Supporting valid-time indeterminacy. ACM Trans. Database Syst. 23(1), 1–57.

Earle, P.S., Bowden, D.C., Guy, M., 2011. Twitter earthquake detection: earthquake monitoring in a social world. Annals of Geophysics 54(6), 708–715.

Erdős, P., Rényi, A., 1960. On the Evolution of Random Graphs. Publication of the mathematical institute of the hungarian academy of sciences 5, 17–61.

Erétéo, G., Buffa, M., Gandon, F., Corby, O., 2009. Analysis of a Real Online Social Network Using Semantic Web Frameworks. In Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (Eds.), The Semantic Web - ISWC 2009, Lecture Notes in Computer Science. Berlin Heidelberg: Springer, pp. 180–195.

Ereteo, G., Limpens, F., Gandon, Fabien, L., Corby, O., Buffa, M., Leitzelman, M., Sander, P., 2011. Semantic Social Network Analysis: A Concrete Case. In Ben Kei, D. (Ed.), Handbook of Research on Methods and Techniques for Studying Virtual Communities: Paradigms and Phenomena. USA: IGI Global, pp. 122–156.

Erik Tjong Kim Sang, Bos, J., Inkpen, D., Farzindar, A., 2012. Predicting the 2011 Dutch Senate Election Results with Twitter. In Farzindar, A., Inkpen, D. (Eds.), Proceedings of the Workshop on Semantic Analysis in Social Media. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 53–60.

Erwig, M., Gu¨ting, R., Schneider, M., Vazirgiannis, M., Erwig, M., Gu¨ting, R., Schneider, M., Vazirgiannis, M., 1999. Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. GeoInformatica 3(3), 269–296.

Estimated internet users [WWW Document], 2012. URL http://www.itu.int/ITU-D/ict/statistics/material/excel/EstimatedInternetUsers00-09.xls (accessed 12.1.12).

EUR-Lex - 31995L0046 - EN [WWW Document], 2013. URL http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML (accessed 12.5.13).

Facebook - Quarterly Report [WWW Document], 2013. URL http://investor.fb.com/secfiling.cfm?filingID=1193125-12-325997&CIK=1326801 (accessed 24.1.13).

Facebook Newsroom [WWW Document], 2012. URL http://newsroom.fb.com/content/default.aspx?NewsAreaId=22 (accessed 18.3.12).

Faloutsos, M., Faloutsos, P., Faloutsos, C., 1999. On power-law relationships of the Internet topology. SIGCOMM Comput. Commun. Rev. 29(4), 251–262.

Federal Business Opportunities [WWW Document], 2012. URL https://www.fbo.gov/index?s=opportunity&mode=form&id=c65777356334dab8685984fa74bfd636&tab=core&_cview=1 (accessed 19.3.12).

Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., 2001. OIL: an ontology infrastructure for the Semantic Web. IEEE Intelligent Systems 16(2), 38 – 45.

FINLEX ® - Ajantasainen lainsäädäntö: 22.4.1999/523 [WWW Document], 1999. URL http://www.finlex.fi/fi/laki/ajantasa/1999/19990523 (accessed 13.2.13).

Floyd, C., Reisin, F.-M., Schmidt, G., 1989. STEPS to software development with users. In Ghezzi, C., McDermid, J.A. (Eds.), ESEC ’89, Lecture Notes in Computer Science. Berlin Heidelberg: Springer, pp. 48–64.

Fortunato, S., 2010. Community detection in graphs. Physics Reports 486(3-5), 75–174.

Fortunato, S., Castellano, C., 2007. Community Structure in Graphs. Encyclopedia of Complexity and Systems Science 1141–1163.

Fu, T., Abbasi, A., Chen, H., 2010. A focused crawler for Dark Web forums. Journal of the American Society for Information Science and Technology 61(6), 1213–1231.

Gadia, S.K., 1992. A Seamless Generic Extension of SQL for Querying Temporal Data. USA: University of Iowa, Department of Computer Science.

Gadia, S.K., Nair, S.S., Poon, Y.-C., 1992. Incomplete Information in Relational Temporal Databases. In Yuan, L.-Y. (Ed.), Proceedings of the 18th International Conference on Very Large Data Bases, VLDB ’92. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 395–406.

Garey, M.R., Johnson, D.S., Stockmeyer, L., 1976. Some simplified NP-complete graph problems. Theoretical Computer Science 1(3), 237–267.

Gephi, an open source graph visualization and manipulation software [WWW Document], 2013. URL https://gephi.org/ (accessed 14.1.13).

Gilad Lotan, Erhardt Graeff, Mike Ananny, Devin Gaffney, Ian Pearce, danah boyd, 2011. The Arab Spring| The Revolutions Were Tweeted: Information Flows during the 2011 Tunisian and Egyptian Revolutions. International Journal of Communication; Vol 5 (2011) 1375–1405.

Girvan, M., Newman, M.E.J., 2002. Community structure in social and biological networks. PNAS 99(12), 7821–7826.

Glance, N., Hurst, M., Nigam, K., Siegler, M., Stockton, R., Tomokiyo, T., 2005. Analyzing online discussion for marketing intelligence. In Ellis, A., Hagino, T. (Eds.), Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, WWW '05. New York, NY, USA: ACM, pp. 1172–1173.

Glance, N.S., Hurst, M., Tomokiyo, T., 2004. BlogPulse: Automated trend discovery for weblogs [WWW Document]. WWW 2004 workshop on the weblogging ecosystem: aggregation, analysis and dynamics. URL http://www.wim.bwl.uni-muenchen.de/courses/businessintelligence/ss07/bi_sose07_blogpulse.pdf

Glänzel, W., Schubert, A., 2004. Analyzing Scientific Networks Through Co-Authorship. In Moed, H.F., Glänzel, W., Schmoch, U. (Eds.), Handbook of Quantitative Science and Technology Research. The Netherlands: Kluwer Academic Publishers, pp. 257–276.

Goyal, S., 2009. Connections: An Introduction to the Economics of Networks. USA: Princeton University Press.

Granovetter, M.S., 1973. The strength of weak ties. The American Journal of Sociology 78(6), 1360–1380.

Gruber, T., 2009. Ontology (Computer Science) - definition in Encyclopedia of Database Systems, Encyclopedia of Database Systems. Berlin, Heidelberg: Springer-Verlag.

Gruber, T.R., 1993. A translation approach to portable ontology specifications. Knowl. Acquis. 5(2), 199–220.

Gruber, T.R., 1995. Toward principles for the design of ontologies used for knowledge sharing. Int. J. Hum.-Comput. Stud. 43(5-6), 907–928.

Gutierrez, C., Hurtado, C.A., Vaisman, A., 2007. Introducing Time into RDF. IEEE Transactions on Knowledge and Data Engineering 19(2), 207–218.

Güting, R., 1994. An introduction to spatial database systems. VLDB Journal 3(4), 357–399.

He, H., Singh, A.K., 2008. Graphs-at-a-time: query language and access methods for graph databases. In Wang, J.T.-L. (Ed.), Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD '08. New York, NY, USA: ACM, pp. 405–418.

Heinonen, O., Hätönen, K., Klemettinen, M., 1996. WWW Robots and Search Engines (Seminar on Mobile Code No. TKO-C79). Helsinki University of Technology, Department of Computer Science.

Helbing, D., Balietti, S., 2011. From social data mining to forecasting socio-economic crises. Eur. Phys. J. Spec. Top. 195(1), 3–68.

Henrique, W., Ziviani, N., Cristo, M., de Moura, E., da Silva, A., Carvalho, C., 2011. A New Approach for Verifying URL Uniqueness in Web Crawlers. In Grossi, R., Sebastiani, F., Silvestri, F. (Eds.), String Processing and Information Retrieval, Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 237–248.

Herre, H., 2010. General Formal Ontology (GFO): A Foundational Ontology for Conceptual Modelling. In Poli, R., Healy, M., Kameas, A. (Eds.), Theory and Applications of Ontology: Computer Applications. Springer Netherlands, pp. 297–345.

Hertel, A., Broekstra, J., Stuckenschmidt, H., 2009. RDF Storage and Retrieval Systems. In Staab, S., Studer, R. (Eds.), Handbook on Ontologies, International Handbooks on Information Systems. Berlin Heidelberg: Springer, pp. 489–508.

Hevner, A.R., March, S.T., Park, J., Ram, S., 2004. Design Science in Information Systems Research. MIS Quarterly 28(1), 75–105.

Home | Objectivity [WWW Document], 2013. URL http://objectivity.com/ (accessed 14.1.13).

Hopcroft, J., Khan, O., Kulis, B., Selman, B., 2004. Tracking evolving communities in large linked networks. Proceedings of the National Academy of Sciences 101(suppl_1), 5249–5253.

Hopcroft, J.E., Motwani, R., Ullman, J.D., 2000. Introduction to Automata Theory, Languages, and Computation, 2nd ed. USA: Addison Wesley.

HTML5 Introduction [WWW Document], 2013. URL http://www.w3schools.com/html/html5_intro.asp (accessed 26.1.13).

IARPA [WWW Document], 2011. URL http://www.iarpa.gov/solicitations_osi.html (accessed 19.3.12).

iMacros [WWW Document], 2013. URL http://www.iopus.com/imacros/ (accessed 8.5.13).

Irmak, U., Suel, T., 2006. Interactive wrapper generation with minimal user effort. In Carr, L., De Roure, D., Iyengar, A., A. Goble, C., Dahlin, M. (Eds.), Proceedings of the 15th International Conference on World Wide Web, WWW ’06. New York, NY, USA: ACM, pp. 553–563.

Jagatic, T.N., Johnson, N.A., Jakobsson, M., Menczer, F., 2007. Social phishing. Commun. ACM 50(10), 94–100.

Jalilian, O., Khotanlou, H., 2011. A new fuzzy-based method to weigh the related concepts in semantic focused web crawlers. In Ting, Z. (Ed.), 2011 3rd International Conference on Computer Research and Development (ICCRD). Washington, DC, USA: IEEE, pp. 23 –27.

Jensen, C.S., Snodgrass, R.T., 1999. Temporal data management. IEEE Transactions on Knowledge and Data Engineering 11(1), 36–44.

Jeong, H., Mason, S.P., Barabási, A.-L., Oltvai, Z.N., 2001. Lethality and centrality in protein networks. Nature 411(6833), 41–42.

Jones, C., Lomet, D., Romanovsky, A., Weikum, G., Fekete, A., Gaudel, M.-C., Korth, H.F., de Lemos, R., Moss, E., Rajwar, R., Ramamritham, K., Randell, B., Rodrigues, L., 2005. The atomic manifesto: a story in four quarks. SIGMOD Rec. 34(1), 63–69.

Järvinen, P., 2004. On research methods. Tampere: Opinpaja.

Kannan, R., Vempala, S., Veta, A., 2000. On clusterings-good, bad and spectral. In Blum, A. (Ed.), 41st Annual Symposium on Foundations of Computer Science, 2000. Proceedings. Los Alamitos, CA, USA: IEEE Computer Society, pp. 367 –377.

Kaplan, A.M., Haenlein, M., 2010. Users of the world, unite! The challenges and opportunities of Social Media. Business Horizons 53(1), 59–68.

Kazienko, P., Musial, K., Kukla, E., Kajdanowicz, T., Bródka, P., 2011. Multidimensional Social Network: Model and Analysis. In Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (Eds.), Computational Collective Intelligence. Technologies and Applications. Berlin, Heidelberg: Springer, pp. 378–387.

Kernighan, B., Lin, S., 1970. An Efficient Heuristic Procedure for Partitioning Graphs. The {B}ell system technical journal 49(1), 291–307.

Kietzmann, J.H., Hermkens, K., McCarthy, I.P., Silvestre, B.S., 2011. Social media? Get serious! Understanding the functional building blocks of social media. Business Horizons 54(3), 241–251.

Kleinberg, J.M., Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A.S., 1999. The Web as a graph: measurements, models, and methods. In Asano, T., Imai, H., Lee, D.T., Nakano, S.-I., Tokuyama, T. (Eds.), Proceedings of the 5th Annual International Conference on Computing and Combinatorics. Berlin, Heidelberg: Springer-Verlag.

Kushmerick, N., 1997. Wrapper Induction for Information Extraction. Washington, DC, USA: University of Washington.

Lamsweerde, A. van, 2009. Requirements Engineering - From System Goals to UML Models to Software Specifications. Glasgow, UK: Wiley.

Larkin, R.W., 2009. The Columbine Legacy Rampage Shootings as Political Acts. American Behavioral Scientist 52(9), 1309–1326.

Larsson, A.O., Moe, H., 2012. Studying political microblogging: Twitter users in the 2010 Swedish election campaign. New Media Society 14(5), 729–747.

Leskovec, J., Horvitz, E., 2008. Planetary-scale views on a large instant-messaging network. In Huai, J., Chen, R., Hon, H.-W., Liu, Y., Ma, W.-Y., Tomkins, A., Zhang, X. (Eds.), Proceedings of the 17th International Conference on World Wide Web, WWW ’08. New York, NY, USA: ACM, pp. 915–924.

Leskovec, J., Kleinberg, J., Faloutsos, C., 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In Özcan, F. (Ed.), Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD ’05. New York, NY, USA: ACM, pp. 177–187.

Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W., 2009. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. Internet Mathematics 6(1), 29–123.

Leung, A., Lin, R., Ng, J., Szeto, P., 2009. Implementation of a Focused Social Networking Crawler [WWW Document]. URL http://courses.ece.ubc.ca/412/term_project/reports/2009/focused_social_net_crawler.pdf

Li, Y., Meng, X., Wang, L., Li, Q., 2006. RecipeCrawler: collecting recipe data from WWW incrementally. In Xu Yu, J., Kitsuregawa, M., Va Leong, H. (Eds.), Proceedings of the 7th International Conference on Advances in Web-Age Information Management, WAIM '06. Berlin, Heidelberg: Springer-Verlag, pp. 263–274.

Limanto, H.Y., Giang, N.N., Trung, V.T., Zhang, J., He, Q., Huy, N.Q., 2005. An information extraction engine for web discussion forums. In Ellis, A., Hagino, T. (Eds.), Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, WWW '05. New York, NY, USA: ACM, pp. 978–979.

Liu, B., 2011a. Structured Data Extraction: Wrapper Generation. In Carey, M.J., Ceri, S. (Eds.), Web Data Mining, Data-Centric Systems and Applications. Berlin Heidelberg: Springer, pp. 363–423.

Liu, B., 2011b. Social Network Analysis. In Carey, M.J., Ceri, S. (Eds.), Web Data Mining, Data-Centric Systems and Applications. Berlin Heidelberg: Springer, pp. 269–309.

Liu, B., Grossman, R., Zhai, Y., 2003. Mining data records in Web pages. In Getoor, L., Senator, T., Domingos, P., Faloutsos, C. (Eds.), Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03. New York, NY, USA: ACM, pp. 601–606.

Liu, B., Menczer, F., 2011. Web Crawling. In Carey, M.J., Ceri, S. (Eds.), Web Data Mining, Data-Centric Systems and Applications. Berlin Heidelberg: Springer, pp. 311–362.

Liu, G., Liu, K., Dang, Y., 2011. Research on discovering Deep Web entries based ontopic crawling and ontology. In Carey, M.J., Ceri, S. (Eds.), 2011 International Conference on Electrical and Control Engineering (ICECE). pp. 2488 –2490.

Luxburg, U., 2007. A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416.

Madhavan, J., Ko, D., Kot, Ł., Ganapathy, V., Rasmussen, A., Halevy, A., 2008. Google's Deep Web crawl. Proceedings of the VLDB Endowment 1(2), 1241–1252.

Mako [WWW Document], 2013. URL http://www.makotemplates.org/ (accessed 8.5.13).

Manifesto for Agile Software Development [WWW Document], 2013. URL http://agilemanifesto.org/ (accessed 31.1.13).

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., 2001. WonderWeb Deliverable D18 [WWW Document]. URL http://www.loa.istc.cnr.it/Papers/D18.pdf

Meijer, E., Bierman, G., 2011. A co-relational model of data for large shared data banks. Communications of the ACM 54(4), 49.

Mellor, D.H., Oliver, A. (Eds.), 1997. Properties. Oxford University Press, USA.

Menczer, F., Pant, G., Srinivasan, P., 2004. Topical web crawlers: Evaluating adaptive algorithms. ACM Trans. Internet Technol. 4(4), 378–419.

Mesbah, A., Bozdag, E., Deursen, A. van, 2008. Crawling AJAX by Inferring User Interface State Changes. In Schwabe, D., Curbera, F., Dantzig, P. (Eds.), Proceedings of the 2008 Eighth International Conference on Web Engineering, ICWE '08. Washington, DC, USA: IEEE Computer Society, pp. 122–134.

Mika, P., 2005. Ontologies are us: A unified model of social networks and semantics. In Gil, Y., Motta, E., Benjamins, V.R., A. Musen, M. (Eds.), International Semantic Web Conference. Berlin, Heidelberg: Springer, pp. 522–536.

Milgram, S., 1967. The Small World Problem. Psychology Today 2, 60–67.

MongoDB [WWW Document], 2013. URL http://www.mongodb.org/ (accessed 10.2.13).

Mukhopadhyay, D., Mukherjee, S., Ghosh, S., Kar, S., Kim, Y.-C., 2011. Architecture of A Scalable Dynamic Parallel WebCrawler with High Speed Downloadable Capability for a Web Search Engine [WWW Document]. arXiv:1102.0676. URL http://arxiv.org/abs/1102.0676 (accessed 4.5.12).

Muslea, I., Minton, S., Knoblock, C., 1999. A hierarchical approach to wrapper induction. In Etzioni, O., Müller, J.P., M. Bradshaw, J. (Eds.), Proceedings of the Third Annual Conference on Autonomous Agents, AGENTS '99. New York, NY, USA: ACM, pp. 190–197.

Nagypál, G., Motik, B., 2003. A Fuzzy Model for Representing Uncertain, Subjective, and Vague Temporal Knowledge in Ontologies. In Meersman, R., Tari, Z., Schmidt, D. (Eds.), On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 906–923.

National Defense Authorization Act [WWW Document], 2006. URL http://www.gpo.gov/fdsys/pkg/PLAW-109publ163/html/PLAW-109publ163.htm (accessed 19.3.12).

neo4j: World's Leading Graph Database [WWW Document], 2012. URL http://neo4j.org/ (accessed 19.3.12).

Network Data Model Overview [WWW Document], 2012. URL http://docs.oracle.com/cd/B28359_01/appdev.111/b28399/sdo_net_concepts.htm (accessed 10.4.12).

Newman, M., 2005. Power laws, Pareto distributions and Zipf's law. Contemporary Physics 46(5), 323–351.

Newman, M., Girvan, M., 2003. Finding and evaluating community structure in networks. Physical Review E 69(2), 1–15.

Newman, M.E.J., 2004. Fast algorithm for detecting community structure in networks. Phys. Rev. E 69(6), 066133.

Noordhuis, P., Heijkoop, M., Lazovik, A., 2010. Mining Twitter in the Cloud: A Case Study. In Bilof, R. (Ed.), IEEE International Conference on Cloud Computing. Los Alamitos, CA, USA: IEEE Computer Society, pp. 107–114.

NOSQL Databases [WWW Document], 2013. URL http://nosql-database.org/ (accessed 17.2.13).

Nunamaker, J.F., Chen, M., Purdin, T.D.M., 1991. Systems development in information systems research. J. Manage. Inf. Syst. 7(3), 89–106.

Oberle, D., Grimm, S., Staab, S., 2009. An Ontology for Software. In Staab, S., Studer, R. (Eds.), Handbook on Ontologies, International Handbooks on Information Systems. Berlin Heidelberg: Springer, pp. 383–402.

Overview — NetworkX [WWW Document], 2013. URL http://networkx.github.com/ (accessed 14.1.13).

OWL 2 Web Ontology Language [WWW Document], 2013. URL http://www.w3.org/TR/owl2-overview/ (accessed 12.5.13).

Palla, G., Barabási, A., Vicsek, T., Hungary, B., 2007. Quantifying social group evolution. Nature 446, 664–667.

Palla, G., Derényi, I., Farkas, I., Vicsek, T., 2005. Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043), 814–818.

Pan, J.Z., 2009. Resource Description Framework. In Staab, S., Studer, R. (Eds.), Handbook on Ontologies, International Handbooks on Information Systems. Berlin Heidelberg: Springer, pp. 71–90.

Papadopoulos, S., Kompatsiaris, Y., Vakali, A., Spyridonos, P., 2011. Community detection in Social Media. Data Mining and Knowledge Discovery 24(3), 515–554.

Paré, G., 2004. Investigating Information Systems with Positivist Case Research. Communications of the Association for Information Systems 13(18), 233–265.

Parsons, T., 1949. The structure of social action; a study in social theory with special reference to a group of recent European writers. New York : Free Press.

Parthasarathy, S., Ruan, Y., Satuluri, V., 2011. Community Discovery in Social Networks: Applications, Methods and Emerging Trends. In Aggarwal, C.C. (Ed.), Social Network Data Analytics. Boston, MA: Springer US, pp. 79–113.

Pelekis, N., Theodoulidis, B., Kopanakis, I., Theodoridis, Y., 2004. Literature review of spatio-temporal database models. The Knowledge Engineering Review 19(03), 235–274.

Peng, L., Wen-Da, T., 2010. A focused web crawler face stock information of financial field. In Zhou, M. (Ed.), 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS). pp. 512 –516.

PHP: PCRE - Manual [WWW Document], 2012. URL http://php.net/manual/en/book.pcre.php (accessed 13.1.13).

Podani, J., Oltvai, Z.N., Jeong, H., Tombor, B., Barabási, A.-L., Szathmáry, E., 2001. Comparable system-level organization of Archaea and Eukaryotes. Nature Genetics 29(1), 54–56.

Prochaska, J.J., Pechmann, C., Kim, R., Leonhardt, J.M., 2012. Twitter=quitter? An analysis of Twitter quit smoking social networks. Tob Control 21(4), 447–449.

Radcliffe-Brown, A.R., 1931. The social organization of Australian tribes. Melbourne : Macmillan & co., limited.

RDF - Semantic Web Standards [WWW Document], 2013. URL http://www.w3.org/RDF/ (accessed 14.1.13).

RDF Vocabulary Description Language 1.0: RDF Schema [WWW Document], 2013. URL http://www.w3.org/TR/rdf-schema/ (accessed 14.1.13).

re — Regular expression operations [WWW Document], 2012. URL http://docs.python.org/2/library/re.html (accessed 13.1.13).

Recorded Future - Unlock The Predictive Power Of The Web [WWW Document], 2012. URL https://www.recordedfuture.com/ (accessed 19.3.12).

Réka Albert, Jeong, H., Barabási, A.-L., 1999. Internet: Diameter of the World-Wide Web. Nature 401(6749), 130–131.

Rekisteriseloste [WWW Document], 2012. URL https://www.jyu.fi/it/laitokset/cs/en/research/socialmediaanalysis/REKISTERISELOSTE

Rheingold, H., 1993. The virtual community: homesteading on the electronic frontier. USA: Addison-Wesley Pub. Co.

San Martín, M., Gutierrez, C., 2009. Representing, Querying and Transforming Social Networks with RDF/SPARQL. In Antoniou, G., Grobelnik, M., Paslaru Bontas Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Z. Pan, J. (Eds.), Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications, ESWC 2009 Heraklion. Berlin, Heidelberg: Springer-Verlag, pp. 293–307.

Schaurer, F., Störger, J., 2010. OSINT Report 3/2010 [WWW Document]. URL http://www.isn.ethz.ch/isn/Digital-Library/Publications/Detail/?id=122008 (accessed 12.5.13).

Schockaert, S., De Cock, M., 2008. Temporal reasoning about fuzzy intervals. Artificial Intelligence 172(8–9), 1158–1193.

Scott, J.P., 2000. Social Network Analysis: A Handbook, 2nd ed. London, UK: SAGE Publications Ltd.

Semenov, A., Veijalainen, J., 2012a. Ontology-guided social media analysis System architecture. In A. Maciaszek, L., Cuzzocrea, A., Cordeiro, J. (Eds.), ICEIS 2012 - Proceedings of the 14th International Conference on Enterprise Information Systems, Volume 2. Portugal: SciTePress, pp. 335–341.

Semenov, A., Veijalainen, J., 2012b. A modeling framework for social media monitoring. accepted to IJWET.

Semenov, A., Veijalainen, J., Boukhanovsky, A., 2011. A Generic Architecture for a Social Network Monitoring and Analysis System. In Barolli, L., Xhafa, F., Takizawa, M. (Eds.), The 14th International Conference on Network-Based Information Systems. Los Alamitos, CA, USA: IEEE Computer Society, pp. 178–185.

Semenov, A., Veijalainen, J., Kyppö, J., 2010. Analysing the presence of school-shooting related communities at social media sites. International Journal of Multimedia Intelligence and Security 1(3), 232 – 268.

Semiocast — Twitter reaches half a billion accounts [WWW Document], 2013. URL http://semiocast.com/publications/2012_07_30_Twitter_reaches_half_a_billion_accounts_140m_in_the_US (accessed 26.1.13).

Shah, N., Musen, M., 2009. Ontologies for Formal Representation of Biological Systems. In Staab, S., Studer, R. (Eds.), Handbook on Ontologies, International Handbooks on Information Systems. Berlin Heidelberg: Springer, pp. 445–461.

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905.

sitemaps.org - Home [WWW Document], 2012. URL http://www.sitemaps.org/ (accessed 11.1.13).

Skoric, M., Poor, N., Achananuparp, P., Lim, E.-P., Jiang, J., 2012. Tweets and Votes: A Study of the 2011 Singapore General Election. In 2012 45th Hawaii International Conference on System Science (HICSS). pp. 2583 – 2591.

Snodgrass, R.T., Ahn, I., Ariav, G., Batory, D., Clifford, J., Dyreson, C.E., Elmasri, R.A., Grandi, F., Jensen, C.S., Kafer, W., Kline, N., Kulkarni, K., Leung, T.Y.C., Lorentzos, N., Roddick, J.F., Segev, A., Soo, M.D., Sripada, S., 1994. TSQL2 Language Specifi_cation [WWW Document]. URL ftp://ftp.cs.arizona.edu/tsql/tsql2/spec.pdf (accessed 12.5.13).

Social media monitoring tools comparison guide [WWW Document], 2013. URL http://files.www.pr2020.com/blog/social-media-monitoring-tools-2012-comparison-guide/PR_SM-Monitoring-Comparison-1_Sheet1.pdf

SPARQL Query Language for RDF [WWW Document], 2013. URL http://www.w3.org/TR/rdf-sparql-query/ (accessed 14.1.13).

Staab, S., Studer, D.R. (Eds.), 2011. Handbook on Ontologies, International Handbooks on Information Systems. Berlin, Heidelberg: Springer.

start [Pajek Wiki] [WWW Document], 2013. URL http://pajek.imfm.si/doku.php (accessed 14.1.13).

Studer, R., Benjamins, V.R., Fensel, D., 1998. Knowledge engineering: Principles and methods. Data & Knowledge Engineering 25(1–2), 161–197.

Sun, J., Tang, J., 2011. A Survey of Models and Algorithms for Social Influence Analysis. In Aggarwal, C.C. (Ed.), Social Network Data Analytics. Boston, MA: Springer US, pp. 177–214.

The Web Robots Pages [WWW Document], 2013. URL http://www.robotstxt.org/robotstxt.html (accessed 12.5.13).

The XML C parser and toolkit of Gnome [WWW Document], 2013. URL http://www.xmlsoft.org/ (accessed 16.2.13).

Thelwall, M., 2001. A web crawler design for data mining. Journal of Information Science 27(5), 319–325.

Thelwall, M., Stuart, D., 2006. Web crawling ethics revisited: Cost, privacy, and denial of service. J. Am. Soc. Inf. Sci. Technol. 57(13), 1771–1779.

Twisted [WWW Document], 2012. URL http://twistedmatrix.com/trac/ (accessed 7.5.12).

Ullman, J.D., Widom, J., 2007. A First Course in Database Systems, 3rd ed. USA: Prentice Hall.

W3C Document Object Model [WWW Document], 2013. URL http://www.w3.org/DOM/ (accessed 13.1.13).

Walther, J.B., 2011. Introduction to Privacy Online. In Trepte, S., Reinecke, L. (Eds.), Privacy Online. Berlin Heidelberg: Springer, pp. 3–8.

Wasserman, S., Faust, K., 1994. Social Network Analysis, Structural Analysis in the Social Sciences. Cambridge, UK: Cambridge University Press.

World Population Prospects, the 2010 Revision [WWW Document], 2010. URL http://esa.un.org/unpd/wpp/Excel-Data/fertility.htm (accessed 18.12.12).

Xia, Y., Zhang, S., Yu, H., 2010. Web wrapper generation using tree alignment and transfer learning. In Software Engineering and Data Mining (SEDM), 2010 2nd International Conference On. pp. 410 –415.

XML Path Language (XPath) 2.0 [WWW Document], 2012. URL http://www.w3.org/TR/xpath20/ (accessed 13.1.13).

Yadav, D., 2010. Design of a novel incremental parallel webcrawler [WWW Document]. URL http://ir.inflibnet.ac.in:8080/handle/10603/2415 (accessed 4.5.12).

Yan, X., Han, J., 2010. Graph Indexing. In Aggarwal, C.C., Wang, H. (Eds.), Managing and Mining Graph Data, Advances in Database Systems. Springer US, pp. 161–180.

Yang, M., Chen, H., 2012. Partially supervised learning for radical opinion identification in hate group web forums. In Rasool Qureshi, P.A. (Ed.), 2012 IEEE International Conference on Intelligence and Security Informatics (ISI). pp. 96 –101.

Yang, S.-Y., 2010. OntoCrawler: A focused crawler with ontology-supported website models for information agents. Expert Systems with Applications 37(7), 5381–5389.

Yang, S.-Y., Hsu, C.-L., 2009. Ontology-supported web crawler for information integration on call for papers. In Chan, P. (Ed.), 2009 International Conference on Machine Learning and Cybernetics. pp. 3354 –3360.

Yang, S.-Y., Hsu, C.-L., 2010. An ontology-supported web focused-crawler for Java programs. In Klamma, R., Lau, R., Chen, S.-C., Li, Q., Ahmad, I., Zhao, J. (Eds.), Ubi-media Computing (U-Media), 2010 3rd IEEE International Conference On. pp. 266 –271.

Yih, W., Chang, P., Kim, W., 2004. Mining Online Deal Forums for Hot Deals. In Zhong, N., Tirri, H., Yao, Y., Zhou, L., Liu, J., Cercone, N. (Eds.), Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, WI ’04. Washington, DC, USA: IEEE Computer Society, pp. 384–390.

Yin, R.K., 2002. Case Study Research: Design and Methods, 3rd ed. London, UK: SAGE Publications, Inc.

YSO - Semantic Computing Research Group (SeCo) [WWW Document], 2013. URL http://www.seco.tkk.fi/ontologies/yso/ (accessed 12.2.13).

Yu, J.X., Cheng, J., 2010. Graph Reachability Queries: A Survey. In Aggarwal, C.C., Wang, H. (Eds.), Managing and Mining Graph Data, Advances in Database Systems. USA: Springer, pp. 181–215.

Yuvarani, M., Iyengar, N.C.S.N., Kannan, A., 2006. LSCrawler: A Framework for an Enhanced Focused Web Crawler Based on Link Semantics. In C.M. Kwan, A., Szczuka, M., Wang, G., Xiong, H. (Eds.), IEEE/WIC/ACM International Conference on Web Intelligence, 2006. WI 2006. Washington, DC, USA: IEEE Computer Society, pp. 794 –800.

Zhai, Y., Liu, B., 2005. Web data extraction based on partial tree alignment. In Ellis, A., Hagino, T. (Eds.), Proceedings of the 14th International Conference on World Wide Web, WWW ’05. New York, NY, USA: ACM, pp. 76–85.

Zhai, Y., Liu, B., 2007. Extracting Web Data Using Instance-Based Learning. World Wide Web 10(2), 113–132.

Zhang, Z., Dong, G., Peng, Z., Yan, Z., 2011. A Framework for Incremental Deep Web Crawler Based on URL Classification. In Gong, Z., Luo, X., Chen, J., Lei, J., Wang, F. (Eds.), Web Information Systems and Mining, Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 302–310.

Zhang, Z., Nasraoui, O., 2009. Profile-based focused crawling for social media-sharing websites. J. Image Video Process. 2009, 2:1–2:13.

Zhao, H., Meng, W., Wu, Z., Raghavan, V., Yu, C., 2005. Fully automatic wrapper generation for search engines. In Ellis, A., Hagino, T. (Eds.), Proceedings of the 14th International Conference on World Wide Web, WWW ’05. New York, NY, USA: ACM, pp. 66–75.

ORIGINAL PAPERS

I

TRACING POTENTIAL SCHOOL SHOOTERS IN THE DIGITAL SPHERE

by

Veijalainen, J., Semenov, A., Kyppö, J., 2010. Tracing potential school shooters in the digital sphere. In Bandyopadhyay, S.K., Adi, W., Kim, T., Xiao, Y. (Eds.), 4th International Conference, ISA 2010, Miyazaki, Japan, June 2010, Proceedings, Communications in Computer and Information Science. Berlin Heidelberg: Springer, pp. 163–178. DOI: 10.1007/978-3-642-13365-7_16

Reproduced with kind permission by Springer.

Tracing Potential School Shooters in the Digital Sphere

Jari Veijalainen, Alexander Semenov, and Jorma Kyppö

Univ. of Jyväskylä, Dept. of Computer Science and Information Systems,
P.O. Box 35 FIN-40014 Univ. of Jyväskylä
{Jari.Veijalainen,Alexander.Semenov,Jorma.kyppo}@jyu.fi

Abstract. There are over 300 known school shooting cases in the world and over ten known cases where the perpetrator(s) have been prohibited to perform the attack at the last moment or earlier. Interesting from our point of view is that in many cases the perpetrators have expressed their views in social media or on their web page well in advance, and often also left suicide messages in blogs and other forums before their attack, along the planned date and place. This has become more common towards the end of this decennium. In some cases this has made it possible to prevent the attack. In this paper we will look at the possibilities to find commonalities of the perpetrators, beyond the fact that they are all males from eleven to roughly 25 years old, and possibilities to follow their traces in the digital sphere in order to cut the dangerous development towards an attack. Should this not be possible, then an attack should be averted before it happens. We are especially interested in the multimedia data mining methods and social network mining and analysis that can be used to detect the possible perpetrators in time. We also present in this paper a probabilistic model that can be used to evaluate the success/failure rate of the detection of the possible perpetrators.

Keywords: School shootings, social media, multimedia data mining.

1 Introduction

Various educational institutions play an important role in the developed countries. Almost every individual spends at least nine years at school and majority of them continue their studies two to ten years more in vocational schools, high schools and universities. Thus, the time spent in these institutions has a big impact for the lives of the individuals both mentally and professionally. Success or failure at various stages in education can have a profound effect for the future of individuals.

Schools have existed several hundred years, but only since the 19th century they have become a mass institution that offers education for the entire population. There have been violent incidents at schools since the 19th century, but they were almost solely performed by adults. Since 1970'ies a new phenomenon can be observed; such attacks that are performed by the students that visit the educational institution at the time of the attack or have recently visited it. The target of the attack can be the institution itself, or the social order among peers. Often it is the goal to kill as many as

S.K. Bandyopadhyay et al. (Eds.): ISA 2010, CCIS 76, pp. 163–178, 2010.

possible people randomly, but in some cases there can be also a list of people that should be at least killed. The attack has been in most cases performed by fire arms, although also explosive devices, Molotov cocktails, and knives and swords have been used alone or in combination with fire arms. This trend started in USA in the 1970'ies and has spread since then to Europe and to some extent to other countries, like Japan and Middle-East.

In this paper we first shortly review the history of attacks that have been performed against or at schools. After that we will present a mathematical model that predicts what the capture rate of the potential school shooters is during the planning phase or just before they are able to launch the attack. We then present an architecture for a system that is meant to crawl the "interesting" multimedia material in Internet and evaluate it from the perspective of the potential dangerous developments. Conclusions close the paper.

2 Historical Notes about School Shootings

The only global public source about school shootings seems to be Wikipedia [1, 2]. In USA there is an authority called National School Safety Center that collects statistics and publishes reports on the issues [5]. In Finland a working group has been set up in 2009 that has produced an intermediary report on school safety issues [7] and has published its final report in February 2010. In Germany, three incidents have happened during 2009 and the German experts have collected a list of 83 actions against such cases [8].

The Wikipedia page [1] reports 18 school shootings or other lethal incidences in or at elementary schools in various parts of the world before the year 1974 (since 18th century), where the perpetrators have been civilian adults, organized military forces or adult terrorists. The first incidence performed by a school kid is from January 17, 1974, when a 14-years old boy named Steven Guy shot the principal and wounded three others with a pistol and revolver at Clara W. Barton Elementary School in Chicago, USA. The reason seems to have been that it was decided that he would be transferred to a social adjustment center. The next similar case is from January 29, 1979, when 16 years old Brenda Ann Spencer killed 2 adults, and wounded eight children and one police officer. The reason for the shootings according to her "I don't like Mondays. This livens up the day". In addition to her, there are currently only two further known female perpetrators, one in USA and one in Germany (the latter from May 2009, see above).

The list of incidents is considerably longer for the secondary schools. The first incident of the current style seems to have happened in Vilnius on May 6, 1925, where two-three students have attacked the board of examiners and other students during final exam with revolvers and hand grenades killing 5-10 people including themselves. Since then there were some cases per each decennium, almost all reported from USA. The frequency of incidents begins to grow during 1960'ies, although during 1958-1966 there are no reported incidents but about six towards the end of the decennium. The first reported cases from Canada are from 1975. In both cases the perpetrators used guns and killed themselves after the shoot out. There are already ten cases during 1970'ies.

During 1980'ies the number of incidents begins to grow, reaching seventeen cases in total. The first incident reported from France is from the year 1984. The student killed his teacher and subsequently himself. The first incident from Finland is from the year 1989, when a 14-year old student killed two fellow students and tried to kill the third one. He used the pistol of his father. Most cases during 1980's are from USA. During 1990'ies the number of listed cases explodes to roughly 70. Most of the cases are from USA, but also two cases in Austria, one in Netherlands, one in UK, and one in Canada are recorded. In addition, there were a few military operations against schools in developing countries.

Since 1999 there have been almost 100 incidents at secondary schools. During the last ten years there have been almost ten incidents in Germany and several in Netherlands and in South-Africa and at least one in Brazil and Finland (in Jokela on Nov. 7, 2007). Perhaps the most known is Columbine High School incident in Colorado, USA, performed by two students of the school, Eric Harris and Dylan Klebold on April 20, 1999. Although they also committed suicide after their rampage, they had left a lot of planning and other literary material, as well as video material behind. Since the release of the material to the public space by the police in 2006, several books have appeared on the case [3, 4]. On March 26, 2001 In Kenya, two students burned down a dormitory and 67 students were killed, 19 wounded. The reported reason for the arson was that the results of the exam were nullified by the university. The pattern seems to be in most cases similar, as concerns the way of performing the attack. In North-America and Europe hand guns are mostly used as weapons. In some rare cases also inflammable liquids like gasoline or explosive devices are used. In Far-East knives have almost solely been used.

The list of incidents on university/college campuses also contains over hundred entries. The first is from the year 1908. The incidents are of private nature (grudge against a particular individual, love affairs within the college faculty etc.) until the case of Charles Whitman, 25, killed 14 people and wounded 32 in 1966,. After this the incidents perpetrated by students usually require one or at most few casualties, until Dec. 6, 1989 incident in Canada, Montreal. During this rampage Marc Lepine, 25, killed 15 and injured 14. He also committed suicide. During nineties the highest death toll in this category was six in 1991 In USA and six in 1997 in Russia.

The last ten years has meant increasing number of incidents in this category. Most of the cases required 1-3 victims. The worst case was the Virginia Tech massacre on April 16, 2007, performed by Seung-Hui Cho, 23. His shoot out left 33 dead and 25 injured. He also committed suicide. The next worst shoot out happened on April 30, 2009 in Baku, Azerbaijan, where Farda Gadirov, 29, shot 13 and injured 10 persons at the Azerbaijan State Oil Academy. He also committed suicide after his shootout. The third worst case has been the Kauhajoki shooting in Finland on September 23, 2008 performed by Matti Saari, 22, who also shot himself after the rampage. The trial against the police officer that had granted the gun license and did not confiscate the weapon of Saari on the previous day, Sept. 22nd, after an interview, has been started on Dec. 3, 2009 [14]. According to media reports the police officer was aware of the two videos that Saari had posted to YouTube where he was shooting with his gun on a shooting range.

There are about ten cases reported in [1] where the perpetrator or perpetrators have been discovered before they been able to launch the attack. Since 2006 all the cases are such that various Internet resources have played a role in foiling the case, because the perpetrators have left there clear enough signals for the planned attack.

3 Background of the Modeling and Related Work

Our basic view is that a baby is not predestined to become a school shooter. Rather, during the millions of interactions with the surrounding world his mental state, ideological and political views develop towards such a potential. Out of those with such a potential some individuals then begin to plan an attack and some of them really perform such an attack, before it is disclosed and prohibited. A school shooter goes through a particular mental process before he begins to plan an attack and finally performs it. During this process he very often engages in certain kind of behavioral patters that should be interpreted in a suitable way while assessing the violence potential.

This result was established in the US report [12] that was based on 37 incidents with 41 perpetrators during years 1974-2000 in USA. Ten detained perpetrators were also interviewed. The same idea is used in Germany [11,13]. According to the findings of [12] 1) school shootings are a special variant among the youth violence that is seldom characterized by excessive alcohol or drug consumption, bad school results, or breaking of common norms. 2) The perpetrators are often introvert and solitaire, 3) the perpetrators were often depressive and had even attempted suicide 4) many of the perpetrators had hinted about their plans in advance 5) just before the attack some negative things happened to the perpetrator or a stabilizing factor was lost from his life.

The German sample in [13] is rather small but covers all the school shooting cases in 1999-2006. In many ways the patterns are similar to those in USA. All the perpetrators came from stable family circumstances, six of them still lived at home. The families were middle class and no family problems were known in the neighborhood. All perpetrators except one had access to fire arms at home. Three had had incidents with police earlier and one was waiting for a trial. All had threatened to bring weapons to school and kill somebody. In some cases the perpetrator had threatened to kill certain teacher or a peer. All seem to have narcissistic treats in their characters and low self-esteem. All had had recently problem in their life with school or a grandparent had just died (in two cases).

From our point of view interestingly, in all seven cases the perpetrators had leaked their intentions to attack to the environment. The last one in 2006 used Internet to do it, others communicated orally with their peers. The authors think, though, that Internet will gain fast more importance in this respect, i.e. leakage might happens more through Internet. This seems to be the case, but it would require more analysis to be more exact about this.

It has been observed in majority of the cases [15] that all school shooters have at least some kind of a plan before they start it. All phases take time, e.g. the detailed planning takes time. Finally, the school shooters often leave a message to the world before they perform the attack and they leak their plans openly or less openly. Nowadays this often happens in social media forums (YouTube, Twitter, private web pages, blogs, etc.). Thus, there is window of opportunity to capture the perpetrator or to protect the target before the perpetrator is able to launch the attack. But could the earlier signs emitted to the inter-subjective space be interpreted correctly?

In general, there is a lot of information about individuals in every society. Before a child is born, there is no information about him or her (well, perhaps in the dreams of the mother and father, though). After the pregnancy has been confirmed the parents and in many countries also authorities will create pieces of information about the unborn child. After the birth information about the child further grows. In the technically developed states this information is often encoded into the digital sphere either by people themselves, other individuals or by authorities. But even in the most developed countries a large amount of information about individuals remains in the consciousness of the other people – and of the person him- or herself. A part of the information will exist in inter-subjective but analog form, on pieces of paper as text, drawings, as photographs, or entire books, etc. These kinds of pieces of information can be very valuable when somebody tries to estimate the potential of a person to attack a school, as the case in Columbine has been shown [3,4]. Unfortunately, the information about a student is often scattered around and nobody has a complete view on it [17, pp. 83-85]. This is partially deliberate and aims at protecting the privacy of the individuals, but partially this follows from the scattered nature of the control and support systems.

The overall situation concerning the information about an individual is depicted in Fig. 1. All digitally encoded public information is roughly information that can be accessed from a usual desktop everywhere in the world, if the user just knows how to search for it. This mostly contains pieces of information the person him- or herself has created and put into a blog, electronic discussion group or other social media platforms, or then somebody from the social of the person network has created and published it. The public digital information can also contain pieces of information collected and released by authorities, such as car ownership, taxation details, or school performance. Finally, some companies might publish some information about their customers. The widest set of digital information contains all the public and private records about a person, as well as information collected by the authorities (complete identity, driving and gun licenses, medical records, criminal record, drug incidents, domicile, etc.) and companies (customer profile, pseudonyms at a site, purchase history, credit card #, phone #, etc.). In Europe it is currently required by a directive [6] that the context information of every phone call, Internet (TCP/IP) connection, or SMS must be stored for 6 to 24 months into a database of an operator. The access to this information is naturally restricted only to authorities and only to cases where police suspects that the person is involved in criminal activities.

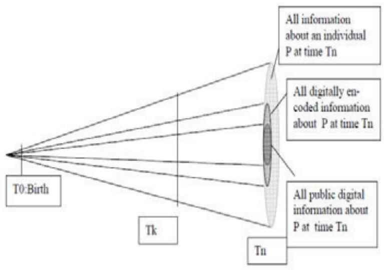

All information about an individual P at time Tn
All digitally en-coded information about P at time Tn
All public digital information about P at time Tn
T0:Birth
Tk
Tn

Fig. 1. Information about an individual over time

We apply the idea that any child goes through a certain development path while becoming an adult. In fact, there are extremely many possible paths. The totality of development paths is still given by the genes of the person. For instance, the maximum height of the person, his or her physical appearance, the potential to be brilliant or less brilliant in learning and using languages or mathematics is determined genetically. And trivially, only women have the potential to become a biological mother and men the biological father. (This potential does not, however, become real for about 19 % of the women in the contemporary USA, according to Wikipedia). During the millions of interactions with the outside world, especially with the social environment, certain potentially possible developments are chosen and others possibly prohibited. A very simple example is the question what languages can the person speak when he or she is 20 years old. If the school did not offer studies in Chinese and English is spoken at home, it is extremely improbable that the kid would understand Chinese when he or she becomes 20. On the other hand, if Chinese was offered at school then he or she might be rather fluent in speaking, writing and reading that language, instead of German or Spanish that he or she might have also learned. That is to say, he or she might have the genetic and mental potential to learn Chinese, but it requires certain conditions in the environment to be realized. Further, if one studies certain languages in the school, it is usually hard afterwards to enhance the language skills. This is because the ability learn new languages becomes worse with the growing age but more importantly, people do not have time any more to study when they have family and work. Thus, choosing certain languages at school prohibits certain others to be ever learned. In many countries with common liability to military service all males are taught to use weapons during the military training, but there is no reason why not almost every female would learn how to use them. Still in many countries this does not happen, because female persons are not given military training and for cultural and other reasons women are not allowed to learn these skills in civil life. Thus, majority of women in the world do not have shooting skills in practice.

Now the above said, are there people that would not become school shooters on any possible individual development path spanned by the genes, i.e. no matter how the life would treat him or her, the kid would not become a school shooter during his or her life. The answer is that with very high probability there are such kids, because we know that there are kids who were treated very badly, but did not organize a rampage. Further, female students hardly ever go on rampage. We can also pose the opposite question, are the people that will become school shooters no matter how their life would treat them? That would mean that they would be predestined by their genes to become school shooters. Here the answer is with extremely high probability no. This is because this way or reacting is given historically and different circumstances and external interventions can prohibit the development. The idea that it is question of development during childhood and adolescence has been elaborated e.g. in [15-18] and in many other sources. Newman's book [18] and [15] argue that these kinds of reactions are culturally determined, and it is the way the young (white) men in suburban and rural communities have begun to solve their problems with regard to social status – and psychological problems that are in some cases caused by the family and further social situation. According to [4] a part of the perpetrators are psychopaths (e.g. Eric Harris, Columbine), some are deeply traumatized (Mitchell Johnson, Arkansas massacre), and some are psychotic (e.g. Dylan Klebold, Columbine). Evidently, such people have gone to the schools also earlier, but they did not organize a massacre. The circumstances in the society have changed in order for this to become an option for individuals.

Now, the model would contain all possible developments the individual would have based on the genes. The concrete interactions with other people and physical environment then enable some of these and at the same time prohibit some of these. This is depicted in Fig. 2. A path in the tree models crucial possible development paths of an individual. The nodes model the crucial changes in external and internal circumstances of the individual. E.g. a divorce of the parents might be such an event, or relocation of the family – or beginning of a series of traumatic events, such as molestation or bullying. How many paths in this tree of possible developments are such that contain the possibility of school shooting? And if the development takes a certain course, does it stay on that course irrespective of the other influences from the environment. The larger the set of different circumstances that cannot misguide the development from the paths that lead to "normal life", the more resilient is the person against troubles in life. And vice versa, the less circumstances and paths lead to "normal life", the more there are risk potential for such an individual. At which phases of the individual development, and with what means, is it possible to keep the development on such paths that does not lead to these kinds of attacks - or any kind of violent attacks within the society? The problem is that – against a common belief – the "middle-class normality" in life would automatically lead to a development towards a "normal life".

How to determine, on which path the individual development is? If some external actor had all possible information about an individual, especially about his feelings and thoughts and the information the person is seeking in the digital sphere and elsewhere, this would be easy. This actor would know when the person started to feel bad, why is this so in his opinion, who and what he hates, when he starts to make plans for an attack, why, how, with whom. But in reality there is no such actor, although the person himself has of course the sufficient information in his consciousness. The path should be determined from the inter-subjective information emitted by a person to other people – or computer programs.

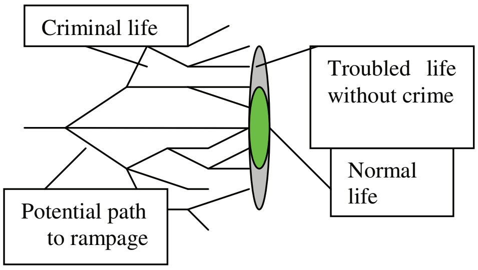


Fig. 2. Potential development paths of an individual

The latter can be used at least in two ways. The first is to design an expert system that is being used by the professionals. The expert system contains the typical characteristics of a path that might lead to a school shooting, especially towards the end of the path. These have been extracted from the past cases and general psychological and criminalist knowledge. The system poses questions that the user should answer concerning the case. The system responds with risk potential analysis. This approach is followed in Dyrias [9,10].

Another approach is to evaluate the public and possibly non-public digital information accessible in the digital sphere and try to find individuals that might be on a path leading to a rampage. This can be done automatically or semi-automatically. What is the information that should be sought? Where? In which relation is this information in the digital sphere to the information known by the peers and other people in the local social environment? Can we draw some conclusions about the path youngsters are on based on the public information? These are questions that we cannot answer at this stage of the research. Rather, we first design a general probabilistic model that should be able to measure the success and failure rates of discovering the dangerous development. It is the matter of further study to estimate the probabilities.

4 The Analytical Model

Based on the above considerations we can design a model that predicts how well it is possible to grasp the development of a person towards a school shooter. We will use a probabilistic approach. The main reason is that there is not a known set of sufficient conditions detectable in the person's life history that would predict with certainty an attack of this type. Also the set of five necessary conditions established e.g. in [18] can be challenged, at least as concerns the access to fire arms. The attack performed by Georg. R in Ansbach [19], Germany was performed without fire arms and several attacks in Far-East have happened with knives. As is known now [3], in the Columbine attack on April 20, 1999 the main weapons were explosive devices, the fire arms were just a backup. Still, in USA it is probably wise to keep the access to fire arms on the list of necessary conditions, because the culture is so gun oriented.

Another reason for using a probabilistic model is that it seems to be the only way try to measure the correctness of the reactions of the environment to various information. E.g. in the Kauhajoki shooting case the local police knew about the shooting videos in YouTube several days before the school shooting , but the decision of a senior police officer on Sept. 22nd, 2008 was that Saari (shooter) can keep his weapon, because the videos are harmless and because Saari seems to be "normal young man". He evidently [20] ignored the text in Saari's YouTube profile, next to the videos. It first speaks about forgotten war that returns, about dead children and how mothers screamed when war came and killed their children in fire, for revenge. It ends "… Whole life is war and whole life is pain; and you will fight alone in your personal war; War; This is war." Beneath are his hobbies: computers, guns, sex and beers. Movies he likes Saw1-3, and further horror movies. Can there be a clearer hint that the person is in a private war and that he will use his gun in it? Still, a study counselor at his school stated that it was impossible to detect any problems in his behavior [20]. This is one of the false negative cases whose probabilities should also be included into the model.

An essential assumption behind the model is that the more information the environment has about the person, his real thoughts and plans, the more probable it is that the correct interpretation of the path is possible. And vice versa. These are two independent events in our model. We model separately the phase of the individual's development until the point where he decides to perform an attack, and the development after that. We assume that it would be possible to redirect the person's development towards normal or at least in the first phase with a certain probability, assuming that the information about him is correctly interpreted. In the other phase he is already on a path leading to the rampage during which the attack is being planned and perhaps also other peers recruited into the plot or informed (cf. the German cases in [14]). In this phase the environment should try to interpret the possible signals correctly and prevent the attack. We distinguish between analog information and digital information in this respect, because the digital information can be mined later and automatic tools can perhaps be used to determine the path. Of course, the person's development can hardly be redirected by remote people, only by local community the person lives in. An attack plan can be disclosed by remote people, perhaps helped by software, but schools or other possible targets can only be protected by local people.

The set of people we primarily target in our model (person i) are the male's currently roughly 10 to 25 years old and having access to Internet. The "other people" that might take preventive actions or influence the development are potentially the entire mankind, but in practice a small group of mostly local people.

For each of those cases we define the following probabilities:

P_i^{dana} = probability that a person i discloses information in an analog form (face-to-face, on a piece of paper, during a phone call) to other people that are relevant to determine that he is on a path that might lead to a rampage (up to a possible attack planning)

P_i^{ddig} = probability that a person i discloses information in the digital sphere (social network sites, www pages, emails, text messages, chat rooms, etc) that are relevant to determine that he is on a path that might lead to a rampage (up to attack planning)

P_i^{iana} = probability that another person j or group of people correctly interprets the analog information and determines the dangerous path the person i is on correctly

P_i^{idig} = probability that another person or a group of them correctly interprets the digital information and determines the dangerous path person i is on correctly

$P_i^{iattack}$ = probability that the potential perpetrator i releases the attack plan including the schedule and perhaps suicide message before attack in an analog form to other people

$P_i^{dattack}$ = probability that the potential perpetrator i releases the attack plan including the schedule and possibly a suicide message before the attack in a digital form either on a social network site or similar or stores it into his computer.

[r,a] = time interval between the release of the attack plan information or exposure of the attack (e.g. weapons missing, perpetrator with weapon sighted) and the planned start of the attack.

$P_i^{avert}(t)$ = probability that police or other people in the community or outside it can avert the attack of i t seconds after they get the information that it is going to happen or is exposed

$P_i^{redirect}$ = probability that the potential perpetrator i is moved to a “normal path” in his life before attack by other people, after his being on the dangerous path has been identified by the local or remote community.

It is clear that the shorter the interval [r,a], the more difficult it is to prohibit the attack. It is also true that t < a-r must hold for averted cases. The probability to avert a planned attack is shown at (1).

$$P_{i,r,a}^{avert} = P_i^{avert}(t) \mid P_i\,(t < a - r) \tag{1}$$

$P_{i,r,a}(t < a - r)$ = probability that the time between the exposure of deliberate disclosure of the plans and the planned start of the attack of perpetrator i is t seconds.

We also conjecture that if the person is moved from a possible development path that might lead to a rampage to a path leading only to “normal life”, the probability of a rampage organized by this person becomes zero. Thus, rampage can only follow in the model, if redirection attempts did not take place at all or failed. In the model, the latter can only be attempted, if suitable signals emitted by person i were observable and were interpreted correctly. Thus, we get an overall measuring function (2).

$$\begin{aligned}&P_i^{noharmtoothers} = \\ &(P_i^{dana}\cdot(P_i^{iana} \mid P_i^{dana}) + P_i^{ddig}\cdot(P_i^{idig} \mid P_i^{ddig})\cdot P_i^{redirect} + P_i^{redirect})\times \\ &\times P_i^{iattack}\cdot(P_i^{avert} \mid P_i^{iattack} + P_i^{dattack}\cdot(P_i^{avert}(t) \mid P_i^{dattack}))\end{aligned} \tag{2}$$

It is our goal to assess the above probabilities based on the existing cases. The function f is already loosely based on real cases. If the attack plan is published on a web site e.g. 5 minutes before the attack begins, there is no much chance the attack could be prevented by authorities. It is almost as probable as foiling the attack when the perpetrator tries to access the school and weapons are discovered at the gate or at the door.

The Kauhajoki case is an example of the false negative interpretation of the available information (see above). It seems that it was done in order to avoid false positive. Matti Saari was interpreted by the senior police officer to be on a normal enough path and he thought that he would just make the life of Saari more difficult, if he confiscated the weapon. Avoiding both false positives (3) and negatives (4) should be the goal. Our model must be slightly enhanced to capture these.

P_i^{dnana} = probability that a person i discloses information in an analog form (face-to-face, on a piece of paper, during a phone call) to other people that are relevant to determine that he is on a normal path or at least not going to organize a school shooting

P_i^{dndig} = probability that a person i discloses information in the digital sphere (social network sites, www pages, emails, text messages, chat rooms, etc) that are relevant to determine that he is on a normal path or at least not going to organize a school shooting

P_i^{wnana} = probability that another person j or group of people wrongly interprets the analog information and concludes that person i is on a path leading perhaps to a rampage although he is not

P_i^{wndig} = probability that another person or a group of them wrongly interprets the digital information about i and concludes that person i is on path leading perhaps to rampage although he is not

Now,

$$P_i^{\text{false positive}} = (P_i^{wnana} \mid P_i^{dnana}) + (P_i^{wndig} \mid P_i^{dndig}) \tag{3}$$

$$P_i^{\text{false negaitive}} = P_i^{dana} \cdot \left(1 - P_i^{iana} \mid P_i^{dana}\right) + P_i^{ddig} \cdot \left(1 - P_i^{idig} \mid P_i^{ddig}\right) \tag{4}$$

It is for further study, what is the exact relationship of the false negatives and false positives. Intuitively, the weaker the external signals of the potential problems in the development of an individual, the bigger the error towards false negative. But what kind of signals make the environment to err towards false positive? And what are the consequences for an individual in this case? Several cases have been lately reported in Finland, where the peers have revealed a student to police that has then taken him to custody and questioned him. It turned out that in one case this was a way of bullying the boy that did not quite fit into the class.

In order to assess the model and find ways of increasing the probabilities of detecting those who are on a path to become a school shooter or who have already determined to launch an attack, we must evidently look at old cases. What were the signals that should have been noticed? While going through the past school shooting cases, it would be tempting to collect as much information as possible concerning the perpetrators, including the medical history, social networks, relationships with opposite sex, social status of the family, complete history of the actions in the social media, etc. This is unfortunately impossible in most cases, especially for the cases lying back tens of years. Gathering even all the available information from the sources mentioned in [2] is as such a big task. Further, we are especially interested in foiled cases, because they make it possible to assess the probabilities with which the attack plans fail. Unfortunately, media coverage is not extensive in these cases, and following the development path of an individual backwards to a point where he turned towards a rampage is difficult due to a lack of available public information.

Many of the interesting pieces of information are such that we cannot capture them in the digital sphere at all or not without breach of privacy, but the more the people disclose information about themselves in the digital sphere the more we can find it and take action.

5 Technologies to Increase Detection Probabilities

The basic idea of using technology is to try to find the signals in the digital sphere that would help the environment to find out which path a young person is on. As it has turned out that the persons closest to the troubled individuals are not able to recognize the signs or interpret them correctly, and because latter sometimes leave clear information about their behavior into digital sphere, we will look into these signs. The first step would be implementing an ontology that captures relevant issues from the above studies.

We as researchers that can only use the public information have roughly four problems:

1) to compile an ontology that would capture the behavior patterns of possible perpetrators;
2) to filter out of hundreds of millions users at various social media sites those persons that might be on path to a rampage;
3) to evaluate closer the cases based on the information gathered;
4) to report towards the local communities if a person has announced targeted threats or a concrete attack plan.

At first it is necessary to compile the profile of potential perpetrator, including the keywords, video patterns and so on and make the set of the inference rules for the ontology system. Then, based on the ontology, system should capture the information from the most versatile web-sites for youth communication and add it to our storage.

The second problem is solved by a web crawler that accesses suitable sites and attempts to gather “interesting contents”, i.e. such contents that contain signals about a person that might launch a rampage. The gathered multimedia content is stored in a database along the meta information that corresponds to the ontology above.

The social network information is also gathered and stored into a social network analyzer. The idea is to investigate, whether the people have connections to earlier perpetrators or to persons that had.

The third issue is basically solved by evaluating the multimedia material, such as videos, audio, text from blogs, etc. that is, multimedia data mining. For instance, finding shooting videos should be rather easy based on the sound of shots. Threatening contents is already a more complicated thing and requires further techniques. We will elaborate these issues below.

Below is presented a basic view on the architecture of the envisioned prototype system, without splitting it to different tiers.

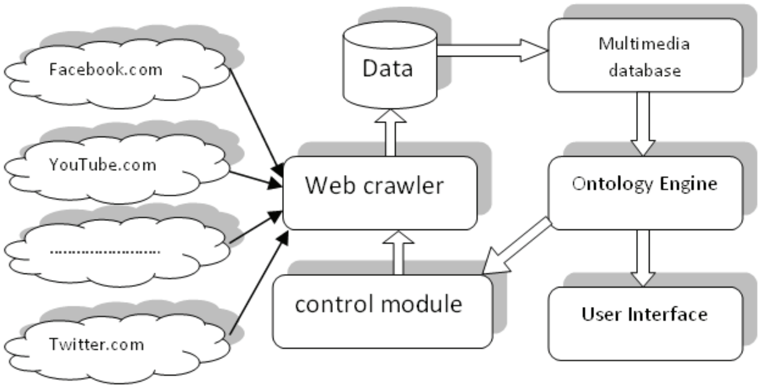


Fig. 3. Potential architecture of the detection and evaluation system

Information is going to be collected from the web with help of *web-crawler* that could fetch the data from public web-pages (like blogs) and protected areas (deep-web). For example, Facebook pages are closed to unregistered persons. Crawler should properly handle these situations. Crawler could be written using PHP or Python, also it is possible to use 3rd party software, like mnoGoSearch [21] or ASPSeek[22].

Fetched information should be stored into a multimedia database[23,26], in this case PostgreSQL will be used. Audio stream should be extracted from video and stored into a separate field of the table.

Data should be mined from multimedia database, using natural language processing, image and video data mining, speech recognition, mood and emotion detection (speech emotion detection, face emotion detection [24,31]), social network analysis (in case we have fetched social network to the database), face recognition techniques, authorship analysis. The latter can be used to find e.g. the texts of the same person at different online resources.

Rules for the data-mining and web-crawling should be contained in ontology referred to above. It can be described using suitable ontology languages (e.g. *OWL*). We can also consider using an expert system like CLIPS [25] for making set of the rules for describing and controlling the mining process.

After disclosing the information to another people in analog form it becomes possible that these people could disclose that information in digital form, for instance, in their blogs [27] (or at least their opinion about the intentions or advices to potential perpetrator). Natural language analysis, emotion detection and social network analysis could increase the probability of fetching such information. Authorship analysis could be useful, if the perpetrators or other people describe the event on some anonymous forums (or under pseudonyms). In the few cases that occurred during the last weeks we observed that people used the same pseudonym on different sites. That makes finding their traces in the digital sphere easy. Just use search engines.

After person has released the information about his intentions on the web it is becoming possible to analyze it using natural language analysis or emotion detection. Basically, it is possible to crawl some web-sites, containing video data (YouTube, Facebook, …), collect the videos into multimedia database, then use multimedia data mining for detecting harassment videos: containing shooting, screams, school shooting descriptions, bomb making process and so on, and analyze the commentators with the help of emotion detection and natural language processing. If a video commentator is positive about such contents, one could argue that P(ddig,i) >0 for that commentator. System should evaluate the contents by various means, comparing to the profile (offending video contents, texts, pictures contents analysis), using expert systems [25], natural language processing, authorship analysis (generating writeprint) [30], social network analysis, sentiment and affect analysis [28,29], image and video analysis [32,33], speech analysis. Finally, human expert should analyze the automatically mined information and change the rule-set of the system.

6 Conclusions

This article discusses basic issues in the rather new social phenomenon of school shootings. We first present a short history of the phenomenon and some recent foiled cases, where information in the web was crucial. We argue that it is possible to use ICT technologies to detect potential perpetrators, at least some of them, based on the traces they leave on various sites in Internet. We present in the paper a basic analysis and categorization of the information about individuals that can be used in general to detect certain behavioral patterns. We also present a mathematical model that can be used to evaluate the probability with which the possible perpetrator would be identified and his attack prohibited, if he advances to that point. The real values of the probabilities must be evaluated from the past cases and are for further study. Finally, we present prototype system architecture. The idea is that we will begin to search in the web such material, guided by an ontology, and evaluate it based on multimedia data mining techniques and social network analysis.

Further research requires a lot of work. One issue is to test the ontology against real cases, and modify it. Further, we will try to deduce upper bounds for the probabilities of finding the potential perpetrators by semiautomatic means and by local community actions. We will also investigate, what are the real values for the false positive and false negative cases.

References

1. Wikipedia, School shooting, `http://en.wikipedia.org/w/index.php?title=School_shooting&oldid=350266831`
2. Wikipedia, listed attacks on schools, `http://en.wikipedia.org/w/index.php?title=List_of_school-related_attacks&oldid=350247039`
3. Cullen, D.: Columbine. Old Street Publishing Ltd., UK (2009)
4. Langman, P.: Why Kids Kill; Inside the minds of school shooters. Palgrave-MacMillan, USA/UK (2009)
5. National School Safety Center, `http://www.schoolsafety.us`
6. DIRECTIVE 2006/24/EC. Official Journal of the European Union, April 13 (2006), `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?Uri=CELEX:32006L0024:EN:HTML`
7. School safety working group in Finland, intermediary report (Oppilaitosten turvallisuustyöryhmä: väliraportti September 14 (2009), `http://www.intermin.fi/intermin/images.nsf/www/oppilaitosten_turvallisuus/$file/oppilaitosten_turvallisuustr_muisto_140909.pdf`
8. Spiegel online: 83 recommendations against school attacks (83 Empfehlungen gegen Amokläufe), `http://www.spiegel.de/schulspiegel/wissen/0,1518,652315,00.html`
9. Dyrias (Dynamisches Risiko Analyse System). Institut Psychologie und Bedrohungsmanagement (2009), `http://www.institut-psychologie-bedrohungsmanagement.de/index.php`
10. Lea, W.: School shootings: Software for early detection (Amoklauf; Software zur Früherkennung). Stern Magazine (March 15, 2009), `http://www.stern.de/wissen/mensch/amoklauf-software-zur-frueherkennung-657867.html`
11. Elstermann, H., Buchwald, P.: School shootings and severe violence in German schools – state of the art (Amokläufe und schwere Gewalt an deutschen Schulen –Stand der Dinge). Bergische Universität Wuppertal, Fachbereich für Bildungs- und Sozialwissenschaften, Study Report, `http://www.petra-buchwald.de/ExamensarbeitAmok.pdf`
12. Vossekuil, B., Fein, R., Reddy, M., Borum, R., Modzeleski, W.: The Final Report and Findings of the Safe School Initiative. U.S. Secret Service and Department of Education, Washington, DC
13. Robertz, F.J.: School Shootings. On the relevance of fantasy for teenagers that killed many (Über die Relevanz der Phantasie für die Begehung von Mehrfachtötungen durch Jugendliche). Verlag für Polizeiwissenschaft, Frankfurt am Main, Germany (2004)
14. Sanomat, H.: The trial against police officer who handled the gun license of the Kauhajoki shooter started (December 12, 2009) (in Finnish), `http://www.hs.fi/kotimaa/artikkeli/Kauhajoen+kouluampujan+aselupaa+koskevan+jutun+k%C3%A4sittely+alkoi/1135251191383`
15. Preti, A.: School Shooting as a Culturally Enforced Way of Expressing Suicidal Hostile Intentions. The Journal of the American Academy of Psychiatry and the Law 36(4), 544–550, `http://www.jaapl.org/cgi/content/abstract/36/4/544`

16. Lieberman, J.A.: School Shootings; what every parent and educator needs to know to protect our children. Kensington Publishing Corporation, NY (2008)
17. Bartol, C.R., Bartol, A.M.: Juvenile delinquency and antisocial behavior: A developmental perspective, 3rd edn. Pearson/Prentice Hall, New Jersey (2009)
18. Newman, K.: Rampage; The social roots school shootings; Why violence erupts in close-knit communities – and what can be done to stop it. Basic Books, NY (2005)
19. Welt, D.: A new turn in Ansbach, Die Wende von Ansbach (September 20, 2009), http://www.welt.de/die-welt/politik/article4573857/Die-Wende-von-Ansbach.html
20. Alueet, Y.: The trial concerning the Kauhajoki school shooting about to end tomorrow, http://yle.fi/alueet/teksti/pohjanmaa/2009/12/kauhajoen_koulusurmaoikeudenkaynti_loppusuoralla_1253891.html
21. MnoGoSearch, http://www.mnogosearch.org
22. AspSeek, http://www.aspseek.org
23. Donderler, M.E., Saykol, E., Arslan, U., Ulusoy, O., Gudukbay, U.: BilVideo: Design and Implementation of a Video Database Management System. Multimedia Tools and Applications 27(1), 79–104 (2005)
24. Maglogiannis, I., Vouyioukas, D., Aggelopoulos, C.: Face detection and recognition of natural human emotion using Markov random fields. Personal and Ubiquitous Computing 13(1), 95–101 (2009)
25. CLIPS: A Tool for Building Expert Systems, http://clipsrules.sourceforge.net
26. Kosch, H., Dollar, M.: Multimedia Database Systems: Where Are We Now?, http://www.itec.uni-klu.ac.at/~harald/MMDBoverview.pdf
27. Li, X., Yan, J., Fan, W., Liu, N., Yan, S., Chen, Z.: An online blog reading system by topic clustering and personalized ranking. ACM Transactions on Internet Technology (TOIT) 9(3), Article 9 (2009)
28. Dunker, P., Nowak, S., Begau, A., Lanz, C.: Content-based Mood Classification for Photos and Music: a generic multi-modal classification framework and evaluation approach. In: Proceeding of the 1st ACM international conference on Multimedia information retrieval, Vancouver, pp. 97–104. ACM, New York (2008)
29. Posner, J., Russell, J.A., Peterson, B.S.: The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology. Development and Psychopathology 17(3), 715–734 (2005)
30. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology 60(3), 538–556 (2009)
31. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. ACM Computing Surveys (CSUR) 35(4), 399–458 (2003)
32. Chen, X., Zhang, C.: Interactive Mining and Semantic Retrieval of Videos. In: Proceedings of the 2007 International Workshop on Multimedia Data Mining (MDM/KDD 2007), in conjunction with the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Jose, CA, USA (2007)
33. Aiyuan, J., Roy, G.: A nearest neighbor approach to letter recognition. In: Proceedings of the 44th annual southeast regional conference, Melbourne, Florida, pp. 776–777 (2006)

III

A GENERIC ARCHITECTURE FOR A SOCIAL NETWORK MONITORING AND ANALYSIS SYSTEM

by

Semenov, A., Veijalainen, J., Boukhanovsky, A., 2011. A Generic Architecture for a Social Network Monitoring and Analysis System. In Barolli, L., Xhafa, F., Takizawa, M. (Eds.), The 14th International Conference on Network-Based Information Systems. Los Alamitos, CA, USA: IEEE Computer Society, pp. 178–185., DOI: 10.1109/NBiS.2011.52

2011 International Conference on Network-Based Information Systems

A generic Architecture for a Social Network Monitoring and Analysis System

Alexander Semenov, Jari Veijalainen*
Dept. of CS&IS
University of Jyväskylä
40014 Univ. of Jyväskylä, Finland
{alexander.semenov, jari.veijalainen}@jyu.fi
*Work was done when the author was the doctoral student of two universities: University ITMO and University of Jyväskylä

Alexander Boukhanovsky
e-Science Research Institute,
The National Research University of Information Technologies, Mechanics and Optics (University ITMO)
197101, Russia, Saint Petersburg, Kronverkskiy pr., 49
avb_mail@mail.ru

This paper describes the architecture and a partial implementation of a system designed for the monitoring and analysis of communities at social media sites. The main contribution of the paper is a novel system architecture that facilitates long-term monitoring of diverse social networks existing and emerging at various social media sites. It consists of three main modules, the crawler, the repository and the analyzer. The first module can be adapted to crawl different sites based on ontology describing the structure of the site. The repository stores the crawled and analyzed persistent data using efficient data structures. It can be implemented using special purpose graph databases and/or object-relational database. The analyzer hosts modules that can be used for various graph and multimedia contents analysis tasks. The results can be again stored to the repository, and so on. All modules can be run concurrently.

Softwares architecture, web crawling, Social network monitoring, social network evolution, dynamic social network analysis

I. Introduction

One of the most rapidly developing phenomena of modern age is the Internet and the simultaneous digitally encoded information accumulation on it. All digitally stored or produced data can be divided to public, semipublic and private or proprietary. The first category covers such data that can be accessed using a publicly available URI by anybody in the world. Often, the URI can be accessed through search engines. Semipublic data are those that one can access by knowing the URI but sometimes also a userid and password are needed. Typically, closed Internet forums and pages hidden from search engines belong to this category. Finally, the last category contains data that can only be accessed by a person, the government or a company owning the data. Digital voice streams in telecom networks and in the Internet (VoIP) also belong to this category. In general, private data is not accessible to the public or search engines through the Internet. Extracted portions can be made accessible, though, to all or some Internet or mobile users in some cases (cf. vehicle register, various statistics, EU and local legislation, position of persons, etc.). In this context we are primarily interested in the public and semipublic data that is generated by individuals and can be accessed at social media(SM) sites, such as Facebook (FB) or LiveJournal (LJ).

The above data carries information that is mostly *about individuals* and generated *by individuals.* A person can expose some information about him- or herself on SM sites, or about others. Most SM sites offer the possibility to record “friend” or other similar relationships, or group memberships between individuals. These form the basis for social group analysis.

As a consequence of the fact that acquisition of the various kinds of information about persons at SM sites and further WWW sites on the Internet is technically easy, such research methodologies as virtual ethnographies have appeared [22]. Their main principle is the qualitative description and analysis of the information collected from the web by some means. Virtual ethnography research may resort to qualitative description of the static aspects of the events at SM sites, or to analysis of the dynamics of their development.

On the other hand, many methods and tools for quantitative analysis of the data have been developed, such as different methods of data-mining and knowledge discovery. These were mainly incorporated in such tools as OLAP and business intelligence software – tools for analyzing historical and current data in order to carry out predictive analysis [23]. Such tools have found their usage in such fields of the industry as marketing, business process analysis and so on. One important direction of the research in this field is the application of computational data-analysis methods to various kinds of data available about the individuals, for the purpose of the detection of the potential perpetrators of serious crimes, such as school shootings, or terrorist acts [1, 21, 24]

What makes social media [45] fascinating as a source of information is the fact, observed e.g. in [34], that some people discuss online topics that they would never have raised “outside” of the Internet, in their real lives. Such discussions can also give impulses to build e.g. hate-group communities and amplify these tendencies. Members of such communities could be influenced by these discussions and finally even commit some kinds of hate crimes as a consequence. Or, a determined individual could find such a hate-community and intentionally influence its members. Important example of similar processes is the phenomenon of school shootings. As described in [1], almost all of the major perpetrators were

978-0-7695-4458-8/11 $26.00 © 2011 IEEE
DOI 10.1109/NBiS.2011.52

IEEE computer society

familiar with the previous cases and evidently also knew discussion groups surrounding school shootings.

In this paper we describe the research, devoted to building a decision support system that utilizes social networks analysis methods and other data analysis methods and automatically collects the data from the web. Only the public data is considered in this phases. The system design as such allows any kind of multimedia data to be used in the analysis, including voice and video streams, text messages, location data, web access logs, medical records, police reports, interrogation protocols, judgments, etc i.e. also this kind of private or semi-public data.

II. Related work

Because of the multi-faceted nature of the topic we will describe related work in several related fields, focused web-crawlers, community detection algorithms, dynamic social network analysis, and similar existing systems.

A. *Focused web-crawlers*

A web-crawler is software that traverses the web relying on some algorithm to select nodes and links, and using appropriate stop policies. Generally, it is possible to divide web-crawlers into two categories: general purpose crawlers and focused crawlers. Goal of the general purpose crawlers is to collect the elements of the graph that are formed by viewing the web pages pointed by an URL as nodes and the URLs pointing inside the pages to other web pages and resources as links. Search engines use this approach. Because there is a path from the used seed nodes through URL-page-URL chains to almost any data in the public data portion (cf. above), a search engine can index the vast majority of the public data. Semi-public and proprietary data are usually not indexed. The public data are thus accessible by key word searches to everybody.

A typical example of social media site crawler is a crawler that tries to fetch the "friends", friends of friends, etc. of a certain person X thus traversing the social network around X at a particular social media site. In this case the nodes of the network are persons (their profiles accessible through an URI) and links are the found "friend" relationships. These might be expressed by URIs pointing to the profiles, but other syntax might also be used. The purpose of the focused crawlers is to collect nodes and links that suit to specific defined topic. Main research directions in the area of the crawlers are: parallelization of the crawler [5], node selection algorithms, crawling of AJAX enabled web-pages [4, 12], refreshing policies [3]. Paper [7] describes architecture of the crawler of twitter messages, which was deployed on two cloud platforms: Amazon AWS and Google AppEngine.

Main difference between a "traditional" and focused crawler is the selection of the pages which suit to particular topic only. There are general algorithms which describe the way of traversing the nodes (in a sense – diffusion of the topic between the nodes), e.g. fish-search and shark-search [2, 6]. An important aspect of focused web-crawlers is the determination of the node topic. The authors of [8, 9, 10, 12, 14] describe the usage of the ontologies for focused crawlers' construction and detection of the topic of the page and description of its structure. The authors of [11] propose the usage of social network tags for topic discovery and describe automatic algorithms for social network website pages' detection (profile page, list page, detail page) based on DOM [30] structure classification. In addition, the body of the research in topic detection could be applied to focused crawlers [13]. Comparison of performance and other characteristics of several different types of focused crawlers is provided in [15].

B. *Community detection algorithms*

Existing algorithms for community detection are mostly based on graph theory [16, 18]. There are several types: graph partitioning methods, clustering methods, modularity based methods, divisive algorithms, spectral algorithms, and methods based on statistical inference. It is also possible to classify community detection algorithms into two categories based on the necessity of the presence/absence of the entire graph for detection. There are algorithms which need entire SN (social network) graph, but also algorithms that only operate on a partial graph ("local communities") [17] and assume that graph will be collected during the process of community discovery. Paper [18] describes the design of the system intended for monitoring blog communities (for allowing "supervisors" to easily detect them) and experiments on Chinese web-sites and blogs.

C. *Dynamic social network analysis and existing software for online intelligence*

The authors of [20] describe the aspects of longitudinal studies of social networks and approach the visualization of the results. The paper contains descriptions of two cases: visualization of CSCL citation network and visualization of the data taken from mail-list of discussions on open source project "OpenSimulator". Interesting from our point of view is the approach for the extraction and visualization of timeline data.

The authors of [40] discuss the aspects of the analysis of historical data on social media site, where the role of edges is played by the data on the recent viewers of the page, represented as the list of identities. The shortcoming mentioned by the authors is mainly the special representation of these data: if the same user (*A*) is watching the profile page of user B at least two times during a time interval, and a number of other users are watching the page between the two visits of user *A, then* only the latest visit of *A* will be represented in the list of recent viewers. Thus it is necessary to query the page more often than users do in order to guarantee accurate analysis.

Paper [21] describes system architecture for the early prediction of terrorist threats. It examines the problems with existing systems and presents a description of the new approach. The architecture of EWAS introduced in [21] contains the following components: 1) acquisition cluster -;– part of the system, responsible for getting the information from the internet and government databases; 2) extraction cluster – responsible for semantic analysis of acquired text and storing it in the internal database; 3) investigation system – system responsible for social network analysis based on the data produced by the extraction cluster; 4) warning generation system – system which sends the warnings for subscribed users, based on the set of rules.

III. Modeling considerations

Nowadays, one of the most popular types of web-sites are online social media sites. The largest one today is Facebook [www.facebook.com]: it currently contains over 600 millions user profiles. In addition, many other web-sites maintain infrastructure intended for building virtual user communities. *Virtual community* in this sense is a social network of individuals who interact through specific media, potentially crossing geographical and political boundaries in order to pursue mutual interests or goals [26]. By *individual* we understand a physical person, but also *application identity* [25] - identity used by the person to access specific social media site. Relations in this sense could be explicitly technically described as links between the profiles, or associations inferred by implicitly defined attributes. There exist many different types of explicit relations between the users: “friends”, “subscribers” etc. All these relations enable the users to stay in touch and communicate with some members of the Social Media Site more often than with other individuals. Implicit relations could be expressed through, for instance, appearing in the same forum thread, or having the same interests [27]. There exist many different definitions of the term “community”, e.g. group of people sharing the same interests; group of people who communicate with members within the group more often than with people outside of the group [28]. The motivation of individuals to join online communities could be explained by several theories: anticipated reciprocity, increased recognition, sense of efficacy, and sense of community [33, 46, 47].

In the present paper we adopt the approach of modeling the social media site with the multidigraph. Nodes of the graph are the identities used at the social media, and edges the relations between them. Basically, social media site is modeled with multidigraph, because many different kinds of edges could exist simultaneously (e.g. “friends”, “subscribers” etc). Edges in multidigraph are directed, and in case that social media site contains undirected ties, edges in the multidigraph are directed mutually. We consider the node as the set of the attributes and their values. Different social media sites could provide different attributes, such as name, picture, set of the favorite videos or music. The values of these attributes could be located on several web-pages, which altogether form the complete profile of the person on the social media site, having a particular identifier. Usually these pages are hyperlinked with one “main” profile web-page. In addition, modern social networks often contain not only profiles of separate users, but for organizations as well. Introduction of such profiles doesn’t affect, generally, the possibility for social media site to be modeled by the multidigraph, but such kinds of nodes are not considered in the present architecture.

One of the important processes, taking place in virtual and other communities is the community evolution [39, 42]: change of the members of the community and links between them as time goes by. A community may be built deliberately by someone [31] or created and developed by a group, depending on the change of the environment: sometimes people tend to unite in groups and act together (e.g. if they share interests and have the same goal). In addition, some individuals may influence other individuals of the community for some actions intentionally or unintentionally (social influence theory) [29].

Why we are devising this software:

- In order to find out how the VCs develop over time in general (size growth etc.)
- What are the main characteristics of different kinds of communities, like hate groups, political movements (cf. the development in Arabic countries[41]), hobby groups, fan groups
- How do the contents submitted to a particular VC influence the opinions of the members and their possible behavior
- Finding out faked identities in VCs
- Detecting threat potential of individuals or VCs
- Following the development of the contents
- Etc.

For all this we need a tool that stores the history of VCs so that longitudinal analysis of them becomes possible.

Collection of the VCs is a multistage process. The first step is to choose the target site, such as Facebook (FB), LiveJournal (LJ), Twitter, etc. There are two main options to start the collection. Either, one can try to extract all the individuals and their relationships at that site and store them into a suitable multidigraph. The second option is to build “local” or partial communities applying incremental search among the individuals belonging to a particular VC at a particular site. This means that the multidigraph is incomplete at any point of time, but evolves towards a complete model of a particular VC at that site. It is important to mention, that the process of collecting of a complete graph is challenging, because the multidigraph could consist of several unconnected components and therefore, one should be able to guess “seed” nodes for each separate VC. Another inherent problem is that the VCs often evolve over the time, and thus the formed multidigraph might not accurately present the real state of the VC at the site. In this paper we assume that time spent in traversing the VC is significantly smaller than what is required for the VC to change substantially. Thus, each formed multidigraph version would rather adequately represent a VC in the reality.

The first approach is computationally intensive and requires the traversal of all the VCs at a particular site. It is hard to carry out in a short time. LJ, for example, has currently ca. 18 million profiles; assuming a processing time of 100 ms/profile would require 1800 000 s, i.e. over 20 days for complete traversal and easily hundreds of terabytes storage to store the multidigraph. The second approach could be implemented as a part of the monitoring systems for online hatred detection, for instance.

IV. System Requirements

In this section we present the requirements and architecture of the system which will focus not only on analysis of static snapshot data, but will also store the history of changes in the network and supports methods for their analysis. We consider monitoring of the changes as more convenient than reconstruction of the timelines, because *a)* Often, profiles related to especially hate content are removed by web-site administrators rapidly *b)* Not all of the changes can be reconstructed from the timeline. For example, SM sites usually don't keep history of changes for the information on edges ("friends") and don't record the time of the addition/removal.

As development methodology for the system we have selected spiral method [38]. The development proceeds as follows: first initial requirements are elicited, then architecture is designed, after that a prototype of the software is developed and tested. This again leads to an extended architecture design and elaboration of the new requirements (and also since one of the main tasks of the software is knowledge discovery, new knowledge about the domain area). For example hate group evolution is planned to be analyzed by the semi-automatic tools and that are used by the experts in sociology and forensic science. Based on the experiences and emerging needs, new requirements will be elicited and added.

The current version of the requirements (see below) was built based on the literature analysis of the related systems, experience of the authors with earlier version of the prototype, and detailed observation of school-shooters' and related groups' online behavior, partially documented in [1]. Such process could be seen also as agile [44] development, because of frequent communication between the involved researchers and quick evaluation of used technologies and methods of data analysis.

The following key stakeholders were identified for this research prototype software and its versions: researchers and target persons. Should this kind of system be really deployed in real use then one must think which actors would use it and possibly benefit or be harmed by its use. Because the use of the system raises privacy concerns, only authorities with corresponding responsibility and legal authorization should use such a system. A more detailed stake holder analysis requires interviews with police and justice department, as well as with social and medical services. It is our intention that we can show what we can do with this kind of software and can then ask them (police etc.) what they think of it.

Based on the literature and on the use of the earlier versions of the prototype we arrived at the following requirements.

A. Functional requirements

- F1 Capability to let a human user describe which sites will be the target of data collection, when this will take place, what data is to be collected, and how they can be accessed (a metadata description of the structure, e.g. ontology)
- F2 Capability to access various SM sites and other WWW sources and retrieve any accessible raw data stored at them, including profile data, "friends" or similar relationships, multimedia contents attached to the profile, comments attached to the contents or profile by other users, etc.
- F3 Capability to retrieve profiles or contents from SM sites based on keywords or on a larger ontology description given by a human user
- F4 Capability to repeatedly retrieve a social network around a particular person (profile) on a particular site based on the "friends" or similar relationships recursively to a given distance ("friends", "friends" of "friends", etc)
- F5 Capability to retrieve the entire social network at a particular SM site, i.e. all profiles, relationships and multimedia contents
- F6 Capability to automatically create a set of (graph) models of the social networks at a SM site, based on the profiles and other data retrieved from them
- F7 Capability to store the raw multimedia data and the above models persistently and in such a way that different instances of the social network model (e.g. around a certain person) and data can be distinguished and placed on a time line
- F8 Capability to automatically or semi automatically analyze various properties of the stored models (and raw data) and visualize the results to a human user and/or store them persistently for a later use

B. Non-functional requirements

- NF1 The system must run in parallel so that data collection (crawling) can happen simultaneously with analysis
- NF2 The crawling performance should be such that all changes at SM sites around the targeted persons can be captured and stored

V. Description of the Architecture

The system consists of three main components: 1) a crawler that continuously crawls the social media sites and other relevant web sites and the activities of their users specified by the user using a suitable ontology, 2) a persistent repository that stores the information gathered by the crawler, and 3) an analysis component that the user can ask to analyze the information gathered and stored into the persistent repository from the social media and other sites. The analysis component also hosts the ontologies that are used in controlling the system.

Figure 1 contains the diagram depicting the architecture of the system.

A. Crawler

Crawler – module of the system which continuously – or on request - crawls the web, according to the task (deep crawling, superficial crawling etc). Task in this sense is a combination of appropriate web-data extraction rules and traversal algorithm (cf. F1).

Connectivity module – submodule which is responsible for maintenance of TCP/IP connection to target web-site and handling of necessary protocols (HTTP, HTTPS) (cf. F2)

Web-Site Parsing Module – submodule responsible for extraction of meaningful information from the retrieved web-page (DOM parsing with XPath) and possibility of usage of API provided by the web-site (e.g. Twitter API) is provided as well. Information can be as well extracted from Social Media site considering multi-page profile structure (e.g such that Facebook maintains: separate pages for user pictures, info etc).the functionality of this module is similar to the "portlets" containing implemented interfaces for extracting the information from different kinds of social media sites (F2)

Traversal algorithms module is responsible for storing the web-graph traversal program, irrelevant to a concrete web-site (BFS traversal, focused crawling etc. (cf. F3-F5).

Parallelism module is responsible for parallelizing crawler traversals of the SM sites (cf. NF2).

Web-site structure cache contains cache of the ontology of web-site structure, fetched from the repository in pre-compiled form (for fast processing) (cf. F2, NF2).

Crawler is connected with repository server by means of ICrawl interface (crawled data transfer, web-site ontology data transfer, task data transfer) and it can also be controlled from UI (cf.F1).

B. *Repository*

Repository is the module storing the database and hosting the necessary applications that handle the graphs (models) and other contents data.

User Profiles temporal storage – temporal graph database storing the structure of social media site at different intervals of time (according to the tasks). It also stores user-profiles data. In this sense structure of SM site at time interval is topological structure of the network and time of the collection. The data stored here are obtained by the crawler module (edges existing in multidigraph depend on the type of the task). During the analysis phase these edges are properly recognized.

Projects storage – schema storing the information on tasks and projects, existing at the system. We assume that the system will be monitoring large number of the social media sites simultaneously, thus set of rules for monitoring will exist in the form of separate projects – user defined tasks for the system (cf. F3,F4).

Analyser storage – storage for the information, produced by analyzer modules. (cf. F8)

Web-site structure storage – metadata storage containing structure descriptions of different web-sites which are subject to be crawled and rules for extraction of the information from the pages (cf. F1)

C. *Analyser*

Analyser contains mainly modules, which are elaborated for analyzing and finding the patterns in the data from data storage by analyzing the crawled data: tools for social network analysis, dynamic social network analysis, and modules for recognition of the pictures and authorship attribution [35]. One of the key design principles of the analyzer is to use the interface IData towards the Repository and make various analyses on the data retrieved. It is also able to create workflows using 3rd party software. Output data from workflow components should be, if configured, sent to input of other modules. That would lead the system to have complicated multi-stage analyzer tasks and it could analyze social networks with implicit ties which is computationally hard to obtain. E.g.: get all pictures from collected social media site -> recognize all the pictures with 3rd party software and extract images of the guns -> take the nodes having these pictures -> carry out SN analysis to find hidden patterns.

Analyser can run continuously and in parallel with the Crawler.

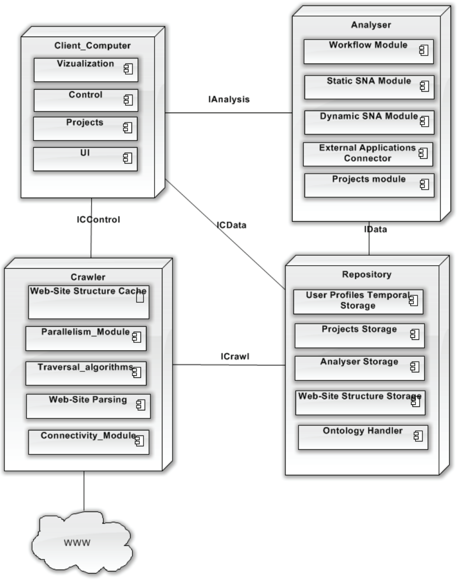


Figure 1. Architecture diagram

VI. IMPLEMENTATION DETAILS

Currently, prototype is implemented using Python programming language [36]. Crawler is implemented as a separate module, which is able to carry out focused crawling, with page selection according to selected criteria (currently as indicator of the affiliation of the page to certain topic we use simple keywords search (bag of words approach). In addition, search of the content using 3rd party search engine is implemented. This allows one to find the given terms on a specific web-page, but the crawler is able to only access pages that are allowed it to access. Additionally, the crawler is able to traverse graph of social media site using breadth-first traversal, for a predefined depth (cf. F4).

The crawler creates, while crawling the site, the mentioned models of social media sites producing multidigraphs. The graph structure is stored at a PostgreSQL database. Database also stores the time of accessing the profile page (time of the collection for node), thus providing the possibilities for analysis of the development history of the groups, based on the accumulation of sufficient quantities of the data (cf. F7).

Currently, we store the graph structure in two tables of a relational database: the first contains nodes, and the second contains the representation of the (labeled) edges between the pairs of nodes. Currently, the types of the edges for multidigraph are explicitly defined. For example, in the current implementation for LiveJournal we use “friend” and “friendof” relations as the edges. Figure 2 depicts the table structure of the database: 3 tables in 3rd normal form. Table “nodes” is relation between internal variable “id”, “username” – SM username, “date” – date of the addition of this node to the database, “color” – column, representing whether the full information on the node (currently – interests) was stored in the DB, and “dist” – service variable, used for traversal of the graph (represents the depth of the traversal). “SM_type” represents the type of the social media, in current implementation it is set to “LiveJournal”. “Edges” table contains information on edges (“id_from” and “id_to” - incidence list used for storing the graph). “Interests” table - “id” – “interest” – one to many relation, storing user’s LJ interests.

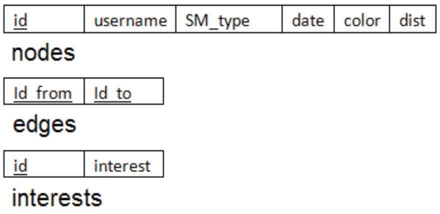


Figure 2. Database structure

As concerns identifying different objects in the models, we assume that social media sites have some symbolic or digital value, which acts as the identifier of the user at the site. The currently considered social media sites adopt the approach where URL for the profile of a user could be inferred based only on the unique identifier of the user at that site. Therefore, we use this identifier as identifier of the node within the digraph in the database, but internally these identities are represented as digital values. In the present version of the software we explicitly describe the type of the social media site in a separate column, and use different tables for representation of the content in different social networks (e.g. interests and other profile data for LJ; and info and uploaded videos for YouTube).

Analyzer module now consists of SNA tools, implemented using NetworkX [37] library for Python language (library which provides graph operations and algorithms). We also partially export data to Pajek and perform analysis using it (cf. F8).

The structure of the social media site pages is being extracted by means of XPath, and also LiveJournal API is implemented.

VII. AN EXAMPLE OF A RETRIEVED SOCIAL NETWORK

As an example, we collected the graph of such users of LiveJournal, who are interested in “mass murder”, “mass murders”, and “mass murderers” (key words). Figure 3 represents the graph of the users retrieved from the site based on the above keywords (by the time of crawling, at LJ there were 533 users having interest “mass murder” – data extracted by LJ API. 220 such nodes were collected by crawler, and 59 of them – who has at least one friend with such interest are represented at figure 3). Edges of the graphs on the pictures represent existence of at least one relation: “friend” or “friendof”, nodes – user profiles. Graph consists of several not connected components. Important from our point of view is at least one subgraph, depicted in figure 4. One of the nodes from this subgraph (“wekillemall”) is a mutual friend of “resistantx” – profile of school shooter Bastian Bosse [1]. So far, we’ve collected about 890 000 unique nodes from LJ network (cf. F5).

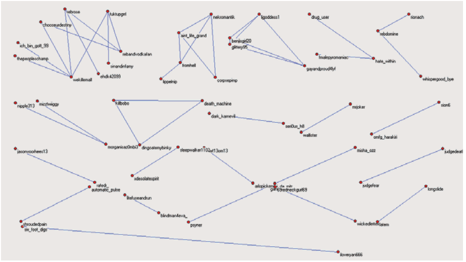


Figure 3. Results: graph formed out of “mass murder” keyword

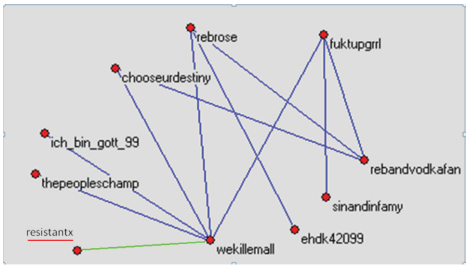


Figure 4. Part of the graph

VIII. CONCLUSIONS AND FURTHER RESEARCH

This paper presents requirement analysis, overall architecture design, and the description of an implemented prototype with some results that have been obtained by running the prototype. The system goal of the system is to gather longitudinal data from social media sites and render it for analysis. Further research is directed to improving the design of the architecture and the software, and analysis of the obtained networks with static and dynamic SNA methods. Important further directions in the architecture research are construction of crawler which would be more effective than the current one, and to extent the extraction of the data from the social media sites. As envisioned in the requirements, one should construct a more flexible system of the "rules" that could guide the process of data gathering from the social media sites.

We currently use only text search in the analysis, but modern social media sites contain large amounts of other multimedia data. Thus, retrieval and analyzing methods for videos, images and audio should be implemented in order to enhance the capture of interesting data and persons.

Another task is the construction of the data repository, which could efficiently unite graph-databases and temporal databases: we collect history of changes for the social media site, and we foresee the following problems emerge should we store a complete instances of a particular social network (i.e. multi-graph), and calculate the differences between them upon a request; or, should we only store the differences (i.e. "deltas") found between two traversals of the social network in the repository? The former makes easy to retrieve the entire graph at a particular moment (or interval) if needed, but difficult to calculate the changes, whereas the latter approach has opposite characteristics. Also, an important aspect is spatio-temporality of the data: the graph might contain also spatial data (city, country, or even address), so special methods should be used for storing this information.

Architecture of the repository for efficient storing and querying structured data from different social media sites is a research problem as well. The current prototype runs even several days on an average laptop performing a single analysis task on the 890000 nodes data.

Regarding the analysis part of the system, we are planning to analyze collected data with different methods, and adopt mathematical models to describe and forecast the processes taking place in complex networks. A further important aspect is that it is necessary to collect large amounts of data, and presumably only after substantial data reservoir it is possible to carry out interesting analyses. We anticipate that we have to store tens or hundreds of terabytes data.

Although the presented research is mainly software engineering, as concerns methods and outcomes, it has strong connections also to other disciplines. The analysis part of the system should especially draw on the knowledge of diverse areas of science such as graph theory, sociology, criminology, and psychology. Currently, we have implemented a prototype that is able to collect data from LJ, store it in the database and carry out basic social network analysis. Earlier versions of the prototype consisted mainly of the crawler that produced data into files in a suitable format. These could then be analyzed by PAJEK [32] or other social network analysis software. As a result we present dataset, collected by the software, which represent the structure of particular online hate-groups related to real crimes.

IX. ACKNOWLEDGEMENTS

The research of the first author was supported by the University of Jyväskylä Graduate School in Computing and Mathematical Sciences (COMAS), as well as by National Research University of Information Technology, Mechanics and Optics (NRU ITMO). The authors gratefully acknowledge Jorma Kyppö from University of Jyväskylä for his valuable comments to the paper and Prof. Andrei Rybin from NRU ITMO for providing the expertise that supported the research.

REFERENCES

[1] A. Semenov, J. Veijalainen, and J. Kyppö, "Analysing the presence of school-shooting related communities at social media sites", International Journal of Multimedia Intelligence and Security (IJMIS), vol. 1 i. 3, 2010, pp. 232-268

[2] M. Hersovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalhaim, and S. Ur, "The shark-search algorithm. An application: tailored Web site mapping", Comput. Netw. ISDN Syst. vol. 30, i. 1-7, Apr. 1998, pp. 317-326.

[3] Q. Tan, and P. Mitra, "Clustering-based incremental web crawling", ACM Trans. Inf. Syst., Vol 28, i. 4, Article 17, Nov. 2010, pp. 1-27 DOI=10.1145/1852102.1852103 http://doi.acm.org/10.1145/1852102.1852103

[4] C. Duda, G. Frey, D. Kossmann, and C. Zhou, "AJAXSearch: Crawling, Indexing and Searching Web 2.0 Applications". Proc. VLDB Endow. 1, 2, Aug. 2008, pp. 1440-1443. DOI=10.1145/1454159.1454195

[5] D. H. Chau, Sh. Pandit, S. Wang, and Ch. Faloutsos, "Parallel Crawling for Online Social Networks", Proceedings of the 16th international conference on World Wide Web, ACM, May 2007, pp 1283–1284.

[6] Zh. Chen, Jun Ma, J. Lei, Bo Yuan, and Li Lian, "An Improved Shark-Search Algorithm Based on Multi-information", Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery - Volume 04 (FSKD '07), vol. 4. IEEE Computer Society, Washington, DC, USA, pp. 659-658.

[7] P. Noordhuis, M. Heijkoop, and A. Lazovik, "Mining Twitter in the Cloud: A Case Study". In Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD '10). IEEE Computer Society, Washington, DC, USA, pp. 107-114.

[8] Sh-Y. Yang, "Developing of an Ontological Focused-Crawler for Ubiquitous Services", Proceedings of the 22nd International Conference on Advanced Information Networking and Applications - Workshops (AINAW '08). IEEE Computer Society, Washington, DC, USA, pp. 1486-1491.

[9] Wei Fang, Zhiming Cui, and Pengpeng Zhao, "Ontology-based focused crawling of deep web sources", Proceedings of the 2nd international conference on Knowledge science, engineering and management (KSEM'07), Springer-Verlag, Berlin, Heidelberg, pp. 514-519.

[10] L. Kozanidis, "An Ontology-Based Focused Crawler", Proceedings of the 13th international conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems (NLDB '08), Springer-Verlag, Berlin, Heidelberg, pp. 376-379.

[11] Zhiyong Zhang, and Olfa Nasraoui.. "Profile-based focused crawling for social media-sharing websites". J. Image Video Process, a. 2, Jan. 2009, pp. 1-13

[12] A. Juffinger et al, "Distributed Web2.0 crawling for ontology evolution," 2nd International Conference on Digital Information Management. ICDIM '07, Oct. 2007, vol.2, no., pp.615-620.

[13] M. Paul, and R. Girju, "Cross-cultural analysis of blogs and forums with mixed-collection topic models", Proceedings of the 2009 Conference on

Empirical Methods in Natural Language Processing: Volume 3 - Volume 3 (EMNLP '09), Vol. 3. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1408-1417.

[14] Sheng-Yuan Yang; Chun-Liang Hsu, "An ontology-supported web focused-crawler for Java programs," 3rd IEEE International Conference Ubi-media Computing (U-Media), July 2010, pp.266-271

[15] S. Batsakis, E. G. M. Petrakis, and E. Milios, "Improving the performance of focused web crawlers". Data Knowl. Eng., vol. 68, i. 10, Oct. 2009, pp. 1001-1013.

[16] S. Fortunato. (2009, June) "*Community detection in graphs*"., 103 p. [Online]. Available: http://arxiv.org/PS_cache/arxiv/pdf/0906/0906.0612v2.pdf

[17] C. Jiyang, O. Zaian, R. Goebel , "Local Community Identification in Social Networks," Proceeding of International Conference on Advances in ,Social Network Analysis and Mining, 2009. ASONAM '09. July 2009, pp.237-242

[18] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection", Proceedings of the 19th international conference on World wide web (WWW '10). ACM, New York, NY, USA, pp. 631-640

[19] Naizhou Zhang, Shijun Li, Wei Cao, "Applying a Multi-Attribute Metrics Approach to Detect Contents of Blog Communities", WiCOM '08. 4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008.

[20] A. Harrer, S. Zeini, and S. Ziebarth, "Visualisation of the Dynamics for Longitudinal Analysis of Computer-mediated Social Networks-concept and Exemplary Cases", From Sociology to Computing in Social Networks. Theory, Foundations and Applications, 2010, v.1, pp. 119 - 134

[21] R.A.Qureshi,, U.K.Wiil and and N.Memon, "EWAS: Modeling Application for Early Detection of Terrorist Threats", From Sociology to Computing in Social Networks. Theory, Foundations and Applications, 2010, pp. 135 – 155.

[22] A. Ameripour, M. Newman, and B. Nicholson, "A Convivial Tool? The Case of the Internet in Iran." Journal of Information Technology v. 25, 2010, pp. 244-257.

[23] Wikipedia, (2011, March 13) "*Businness intelligence tools*", [Online]. Available: http://en.wikipedia.org/wiki/Business_intelligence_tools,

[24] H. Chen, E. Reid, J. Sinai, A. Silke, and B. Ganor, "Terrorism Informatics: Knowledge Management and Data Mining for Homeland Security" (1st ed.). Springer Publishing Company, Incorporated. 2008

[25] O. Mazhelis, J. Markkula, and J. Veijalainen, "An integrated identity verification system for mobile terminals", Information Management & Computer Security, vol. 13, i 5, 2005, pp. 367-378.

[26] F.S.L.Lee, D. Vogel, and M. Limayem, "Virtual Community Informatics: A Review and Research Agenda", The Journal of Information Technology Theory and Application (JITTA), vol. 5, i. 1, 2003, pp. 47-61.

[27] M. Smith, C. Giraud-Carrier, and N. Purser. 2009. "Implicit affinity networks and social capital". Inf. Technol. and Management vol. 10, i. 2-3, Sept. 2009, pp. 123-134. DOI=10.1007/s10799-009-0057-2 http://dx.doi.org/10.1007/s10799-009-0057-2

[28] G.A. Hillery, Jr., "Definitions of Community: Areas of Agreement", Rural Sociology, vol. 20 i. 4, 1955, pp. 111-122

[29] L. Rashotte, "Social Influence", The Blackwell Encyclopedia of Social Psychology, Malden: Blackwell Publishing, 2007, pp. 562-563

[30] "Document Object Model", (2011, March 15) [Online]. Available: http://www.w3.org/DOM/

[31] K. de la Pena McCook. A Place at the Table: Participating in Community Building. Chicago: American Library Association, 2000

[32] Pajek, (2011, March 18) [Online]. Available: http://vlado.fmf.uni-lj.si/pub/networks/pajek/

[33] A. Java, X. Song, T. Finin, B. Tseng, "Why we twitter: understanding microblogging usage and communities", Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis, 2007, pp. 56–65

[34] E.S. Orr, M. Sisic, C. Ross, M. G. Simmering, J. M. Arseneault, and R. Orr, "The influence of shyness on the use of Facebook in an undergraduate sample". CyberPsychology and Behaviour, vol. 12, i.3, 2009. pp. 337-340

[35] E. Stamatatos, "A survey of modern authorship attribution methods". J. Am. Soc. Inf. Sci. Technol., vol. 60, i. 3, Mar 2009, pp. 538-556. DOI=10.1002/asi.v60:3 http://dx.doi.org/10.1002/asi.v60:3

[36] Python Programming Language, (2011, March 10) [Online]. Available: http://www.python.org/

[37] NetworkX, (2011, March 13) [Online]. Available: http://networkx.lanl.gov/

[38] R.L.Nord and J.E. Tomayko, "Software architecture-centric methods and agile development" . IEEE Software, March-April 2006, pp. 47-53.

[39] W. Richards, and N. Wormald. "Representing Small Group Evolution", Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04 (CSE '09), vol. 4, IEEE Computer Society, Washington, DC, USA, 2009, pp. 159-165.

[40] Jing Jiang et al. "Understanding latent interactions in online social networks". In Proceedings of the 10th annual conference on Internet measurement (IMC '10). ACM, New York, NY, USA, 2010 pp. 369-382.

[41] "*The Cascading Effects of the Arab Spring*", (2011, February 23) [Online]. Available: http://www.miller-mccune.com/politics/the-cascading-effects-of-the-arab-spring-28575/

[42] C. Aggarwal, Social Network Data Analytics,. (Ed.) 1st Edition., 2011

[43] O. Preiss, and A. Wegman , "Stakeholder's Discovery and Classification based on System Science Principles", Proceedings of the Second Asia-Pacific Conference on Quality Software, IEEE Computer Society Washington, DC, USA, 2001, 0-7695-1287-9/01

[44] J. Shore, & S. Warden, "The Art of Agile Development". O'Reilly Media, Inc.

[45] A. M. Kaplan, M. Haenlein, "Users of the world, unite! The challenges and opportunities of Social Media", Business Horizons, Vol. 53, i. 1, Jan.-Feb. 2010, pp. 59-68, ISSN 0007-6813

[46] H. Krasnova, T. Hildebrand, O. Guenther, O. Kovrigin, and A. Nowobilska, "Why Participate in an Online Social Network? An Empirical Analysis". ECIS 2008 Proceedings, 2008

[47] P. Kollock, "The Economies of Online Cooperation: Gifts and Public Goods in Cyberspace", 11 New Fetter Lane, London EC4P 4EE: Routledge, 1999, ch. 9, pp. 220–239. [Online]. Available: http://www.sscnet.ucla.edu/soc/faculty/kollock/papers/economies.htm

V

A REPOSITORY FOR MULTIRELATIONAL DYNAMIC NETWORKS

by

Semenov A., Veijalainen J., 2012
International Workshop on Complex Social Network Analysis (CSNA), The
2012 International Conference on Advances in Social Networks Analysis and Mining,
ASONAM 2012, Istanbul, Turkey. DOI 10.1109/ASONAM.2012.174

Reproduced with kind permission by IEEE.

2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

A Repository for multirelational dynamic networks

Alexander Semenov
Dept. of Computer Science and Information Systems
The University of Jyväskylä
Jyväskylä, Finland
NRU ITMO, Saint-Petersburg, Russia
alexander.v.semenov@jyu.fi

Jari Veijalainen
Dept. of Computer Science and Information Systems
The University of Jyväskylä
Jyväskylä, Finland
jari.a.veijalainen@jyu.fi

***Abstract*— Nowadays, WWW contains a number of social media sites, which are growing rapidly. One of the main features of social media sites is to allow to its users creation and modification of contents of the site utilizing the offered WWW interfaces. Such contents are referred to as user generated contents and their type varies from site to site. Social media sites can be modeled as constantly evolving multirelational directed graphs. In this paper we discuss persistent data structures for such graphs, and present and analyze queries performed against the structures. We also estimate the space requirements of the proposed data structures, and compare them with the naive "store each complete snapshot of the graph separately". We also investigate query performance against our data structure. We present analytical estimation results, simulation results, and discuss its performance when it is used to store entire contents of Livejournal.**

temporal social network analysis, multirelational social network analysis, temporal databases

I. INTRODUCTION

Social media sites are rather recent phenomena in the Internet. They have appeared during the last 5-7 years, and attracted large numbers of users. Examples of the largest social media sites are Facebook with over 900 million users, Twitter with 200 million, and recently opened social network Google+, with170 million users. Social media sites are populated with user generated content, such as various text messages, pictures, and other multimedia data, which are linked with each other in different ways. Much of these contents are public, and they are easily accessible by third parties using web-browsers.

Information existing within social media sites models some part of the social reality, and thus by analyzing this information it is possible to make inferences about the state of the affairs in the real world. Most social media sites allow for observation of the data produced by a person or a set of persons at a particular moment of time. Extraction of this data from social media sites can be done quite easily, and obviously it would be nearly impossible to do it for the people in reality. In most cases it is not possible to extract full history of the states, though. Typically, after an entity (a message or a link to another user profile, etc.) is removed, the social media platform does not explicitly report the deletion to the users not directly affected. Later, the traces of the earlier existence of the deleted entity may vanish altogether from the site Thus, in general, for having the history of the states of the social network at a site, it is necessary to poll the web site continuously - or often enough- or to use a streaming API offered by the site. In the case of polling the rate should be higher than the rate of changes at the site [3]. Collection of the data external to the web sites is usually facilitated using special software for data gathering, *web crawlers* [19]. They are used by search engines to collect data, and there are many 3rd party crawlers which can be used for collection of the data from the social media sites. Also, a number of social media sites provide special APIs for data extraction: e.g. Twitter offers a streaming API [1] that pushes the data continuously.

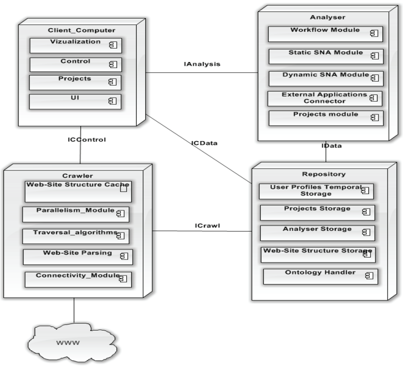


Figure 1. System architecture, from [2].

As discussed in [3], one can distinguish three levels of models in the social media monitoring activity. The social media sites model a certain portion of the immediate social reality. We call these *first level models*, conceptualized as *site ontologies.* These are then remodeled by a monitoring site by the *second level model* that in our approach is a multirelational directed graph that should capture all the site ontologies. The *third level model* is then the persistent data repository level, where we need to store the second level models persistently. In practice, we must map the above general graph structure onto the data model of a database management system, because using the file system facilities would not offer the functionality needed to analyze the graph. In this paper we discuss the architecture of a repository that allows persistent storage for

978-0-7695-4799-2/12 $26.00 © 2012 IEEE
DOI 10.1109/ASONAM.2012.174

time varying, multirelational directed graphs in a condensed form, i.e. only storing "deltas", not the entire snapshot. We also present an OO interface of the repository and the algorithms for populating the repository and extracting the data from it. Further, we compare this approach with the naïve approach storing persistently complete snapshots of the graph at specific moments of time.

II. Repository Architecture

A. Repository interface

Figure 1 depicts the architecture of the system. Repository keeps data of the evolving multirelational graphs. It also provides query interface for analysis tasks. The collected time-varying multirelational directed graph instances are characterized by unique instance identifiers that relate the parts to each other. The graph operations below are the central parts of the repository interface (Table 1).

#	Description	Name	Return
1	Add graph instance X to the repository	addGraph(X,name)	ID of the instance , X

2	Add/remove node N of type C to graph X at time T	addNode(X,N,T)
		rmNode(X,T,N)
			Node

3	Add/remove edge E connecting nodes N1 and N2 to graph X at time T	addEdge(X,N1,N2)
		rmEdge(X,T,N1,N2)
			Edge

4	Add attribute A to node N/edge E of graph X at time T	addnAttr(X,A,N)
		rmnAttr(X,T,A,N)
		addeAttr(X,A,E)
		rmeAttr(X,T,A,E)	

5	Extract graph G (slice of the instance X) at time T, in predefined format	getGraph(X,T,F,T)	Graph i format F
6	Extract validity interval [t1, t2] for instance X	getInt(X)	time interval
7	Extract sequence of the graphs {G} during time interval [t1, t2] from instance X	getSeq(X,t1,t2)	list of graphs G

8	Extract inserted/deleted nodes/edges in X during [t1,t2]	getNewNodes(X,t1,t2)
		getNewEdges(X,t1,t2)
		getDelNodes(X,t1,t2)
		getDelEdges(X,t1,t2)
			list of nodes/edges

9	Extract graph, induced by certain types of nodes/edges from the graph during [t1,t2]	getGraph(types)	Graph
10	Get attribute value of node N/edge E of X at T	getVal(X,A,T)	value
11	Get node by attributes	getNode(attr A,X,type)	Node
12	Get edge by two nodes and type	getEdge(N1,N2,X,type)	Edge

Table 1 - operations

Operation 1 is called from the control module; then, crawler starts data collection and adds nodes, edges, and attributes to the repository. Node/edge/attribute removal operations can be called from control module. Query for the entire graph extraction (5) in predefined format (such as GraphML, Pajek format, etc) can be used for feeding the graph into 3rd party graph analysis software, such as Pajek or Gephi, or implemented within analyzer module algorithms; same for the query (7). Query (8) can be used for e.g. application of stream algorithms or visualization of graph dynamics. Query (9) can be used for extraction of the graphs, consisting of nodes of predefined types only, and further sending them as input to various analysis sub-modules. Node, Edge, and Graph (from table 1) are internal datatypes, storing attribute values and types of the entities (thus, type is not presented in list of parameters).

B. Repository design

There are many ways to represent the graphs, including adjacency matrix, adjacency list, incidence matrix, and incidence list. These data structures can be stored in RDBMS such as PostgreSQL, in plain files, or in distributed file system like HDFS, accessible by Hadoop [4]. Also, graph databases, using their modelling facilities and physical schemes can be used.

III. DB Schemes

A. Snapshot DB Scheme

When temporal multirelational directed graph is stored as a sequence of the snapshots, persistent data storage should contain an individual snapshot of the graph G, for those time moments T, when an instance of the multirelational graph was captured. Since crawling of the web-site takes place during a time interval rather than in a single moment, we consider T as the time stamp marking the collection start. This can be represented as the following DB scheme:

NODES(i_id, node_id, node_type, T);
EDGES(i_id, node_from, node_to, edge_type, T);
X_ATTRS(i_id,node_id, attr_id, attr_value, T);
MDATA(i_id,name);
S_GRAPHS(s_id, i_id, node_id);
S_GRAPH_META(s_id,i_id, name);

NODES is the table with node data, node_id is the identifier of the node, and node_type is the type of the node. EDGES table contains edges of the graph, where node_from and node_to are both node identifiers, node_id, which are end points of this particular edge, and edge_type represents the type of the edge in question. Tables N/E_ATTRS contain attributes for the nodes and edges respectively, attr_id is identifier of the attribute, attr_value is the value of the attribute, and node_id and edge_id are identifiers of the node and edge attribute related to. T is the sequence number of the snapshot, i_id is number of crawling task, relating instance to metadata (table MDATA associates instance i_id to text description). S_GRAPHS stores nodes representing induced subgraphs.

While crawling, the crawler keeps i_id and T in the main memory. When *addNode* operation is called, its implementation checks the existence of the *Node* in the N_ATTRS table (by selecting its attribute, e.g. username for LJ), and adds a row, identified by node_id to the NODES table. If the node does not exist in N_ATTRS, its attributes are added to N_ATTRS and *addEdge* function adds row to EDGES table. Extraction of the graph G at moment T from the snapshot database is straightforward and includes only extraction of a particular snapshot from the DB, identified by T and the instance id (i_id). The graph might be represented as a list of its nodes and a list of edges. In addition, queries extracting subgraphs of nodes or edges of various types are also possible. In that case node_type, or edge_type should be specified in the query. Operations getting the difference between the graphs might be implemented using SQL EXCEPT statements. Although data insertion and extraction procedures for the snapshots method of storage are conceptually simple and can be easily implemented, the problem of this method is the quick growth of the data. Thus, for graph having N nodes there can be N*(N-1) edges maximum, and from 0 to A_N attributes for each node N, and from 0 to A_E attributes for the edges. Because the number of users can be hundreds of millions, already the number of the rows in the edge table can reach 10^16 in theory for the biggest sites. The total size of the table in bytes would be:

$$MAXSIZE(G) = SIZE\_NODE*N + SIZE\_EDGE*N*(N-1) + SIZE\_ATTRIBUTE*(N+E) \quad (1)$$

Thus, the maximum size of the graph would grow in proportion to the square of the number of the nodes in the worst case. In addition, the same or almost the same graph would be copied into the different snapshots, although almost all nodes and edges would remain the same. Thus, the overall space consumption would be MAXSIZE*N_OF_SNAPSHOTS. On the other hand, very few graphs modeling social networks would be fully connected opposite to the worst case above. E.g. recent study of the number of friends in Facebook showed that it is in average 100-200 [6] (cf. Dunbar's number).

B. *Temporal DB scheme*

Scheme of the example of temporal storage is presented in [3]. Tables represent the same concepts as in snapshot storage described above; however, instead of having snapshot identifier T for each table, columns t_st and t_end are inserted. These columns represent validity time intervals of the entities [7]. The database scheme is modified as follows:

NODES(i_id, node_id, n_type, t_st, t_end);
EDGES(i_id, node_from, node_to, e_type, t_st, t_end);
N/E_ATTRS(i_id, n_id, a_id, a_value, t_st, t_end);
MDATA(i_id,name);
S_GRAPHS(s_id, i_id, node_id);
S_GRAPH_META(s_id, i_id, name);

For this schema, no snapshot identifier is used. Rather, while an element is inserted, its validity period should be checked in the repository. In case the element is valid for the current moment of time, its t_end is denoted as NOW(). If element does not exist, current value of the time is assigned to start moment of its validity interval (t_st), and a new identifier is created. If the element was valid at some other time interval before, the same procedure is repeated, but its identifier is taken from the previous validity interval. If element is still valid, then it is not changed. Existence of the node is checked by selecting its identifier by the attribute values, for edge by start and end nodes and type. MDATA table contains metadata.

i_id	node_id	n_type	t_st	t_end
1	1	"profile"	0	5
1	2	"profile"	0	3
1	3	"profile"	2	5
1	4	"video"	0	5
1	5	"video"	3	5
1	6	"video"	3	5
1	7	"profile"	4	5

a. NODES table

i_id	node_id	attr_id	attr_value	t_st	t_end
1	1	"name"	"Alice"	0	5
1	2	"name"	"Bob"	0	3
1	3	"name"	"John"	2	5
1	4	"video_name"	"video1"	0	5
1	5	"video_name"	"video2"	3	5
1	6	"video_name"	"video3"	3	4
1	7	"name"	"Marylin"	4	5
1	6	"video_name"	"video4"	4	5

b. N_ATTRIBUTES table

i_id	node_from	node_to	edge_type	t_st	t_end
1	1	4	"upload"	0	5
1	1	5	"upload"	3	5
1	1	6	"upload"	3	5
1	7	1	"subscribe"	4	5
1	2	1	"subscribe"	0	5

c. EDGES table

Tables a, b, and c represent example of the populated data storage. It models evolution of the SMS, having two node types, profile and video, two relation types, "upload" and "subscribe", and one attribute per node (name and video_name): at the time moment 0 there were two nodes of type "profile" and one node of type "video", and two relations: node with id 1 uploaded node 4, and node with id 2 was subscribed to node 1. At time moment 1 nothing happened, and at time moment 2 profile node with id = 3 appeared. At time moment 3 two "video" nodes appeared, with identifiers 4 and 5, at same moment relations "upload" appeared from node with id 1 to these nodes. That models uploading of the videos. The presented scheme allows for extraction of the graph G at the moment T by specifying the time moment T. Rows are added in the following cases: a completely new node is added, a node is added after deletion (gap in validity interval appears), a completely new edge is added, a gap in validity interval of edge appears, attribute value is added or changed. Thus, data is repeated not for every snapshot, but only for the changes. Total data size would be dependent of the speed of change of the social media site. Now we do not present exact formula for the growth and leave this subject for further studies.

We present the analysis of the data growth, estimated using constructed discrete time simulation model. Simulation model models the network with the following configuration: types of the nodes: {User, Video}, types of the attributes: User: {username, name_of_channel}; Video: {video_url, video_name, header, likes_num, dislikes_num}, types of the edges: Subscriber, Upload.

Action	P	Action	P
Add node	0.1	Remove subscriber	0.1
Remove node	0.1	Change the channels name	0.1
Add video	0.1	Change video name	0.1
Remove video	0.1	Update likes	0.9
Add subscriber	0.1	Update dislikes	0.9

Probabilities

Probabilities of the events are shown at the table "Probabilities". Additional rules: video can be uploaded by existing user, when user is removed all his videos are removed. "Subscribers" relation exists between the users only. Initial graph is modeled as Erdős–Rényi [8] graph with 10 as the number of initial nodes, and 0.1 as probability of edge between two nodes. We run the model for N = 709 timesteps (for temporal DB). Figure 2 depicts insertions of the rows to NODES table, where X axis represent time moment, and Y axis represent number of additions for the interval [X, X+10). At the very end 331 rows were added and table had 7396 rows.

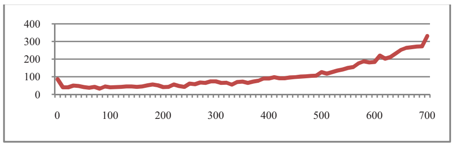


Figure 2. NODES table insertions

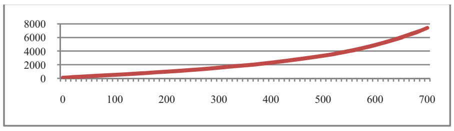


Figure 3. NODES table growth

Figure 4 depicts insertions of the rows to EDGES table, and Figure 5 depicts the growth of the table. At the end there were 272456 rows. At the end table N_ATTRS had 103807 rows.

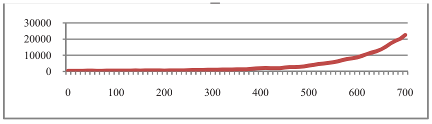


Figure 4. EDGES table insertions

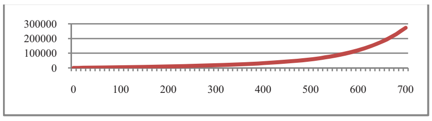


Figure 5. EDGES table growth

From the pictures we can see, that the growth of the EDGES table is faster than NODES.

IV. Related work

There are a number of papers devoted to social network analysis. Survey of the results can be found in [9]. Paper [10] introduces a model of the social network as multi-layered graph, where each type of relations is presented at different levels. Paper [2] introduces multidigraph: directed graph, having multiple types of nodes and relations, paper [10] discusses multirelational networks. Paper [11] presents time aggregated graphs. Dynamic graph algorithms [12] are relevant for our research as well. This research is related to data stream analysis [13]. A body of research is devoted to construction of temporal [7] and spatio-temporal[14] databases. There are also industrial solutions which aim at storing various graphs and networks: graph database Neo4j [15] and Oracle Network data model [16]. There is research on community detection algorithms [17] and influence detection algorithms [18].

V. Conclusions and Further research

Future research direction is analytical estimation of the size of storage for the temporal scheme, and analytical estimation of the procedures for insertion and extraction of the data. Another direction is more exact estimation of the parameters of simulation model for achieving more accurate results of the simulation. This includes substitution of the constant probabilities with functions, depending on the network characteristics and estimation of functions' parameters by analysis of growth of real multirelational graphs. An additional direction is implementation of the dynamic algorithms (which make use of results of the previous solutions) and its adoption to temporal scheme of storage of multirelational graphs.

VI. References

[1] "The Streaming APIs | Twitter Developers." [Online]. Available: https://dev.twitter.com/docs/streaming-apis. [Accessed: 16-May-2012].

[2] A. Semenov, J. Veijalainen, and A. Boukhanovsky, "A Generic Architecture for a Social Network Monitoring and Analysis System," 2011, pp. 178–185.

[3] A. Semenov and J. Veijalainen, "A modeling framework for social media monitoring," *IJWET*, 2012.

[4] U. Kang, "Mining Tera-Scale Graphs: Theory, Engineering and Discoveries," Carnegie Mellon University, USA, 2012.

[5] A. Semenov and J. Veijalainen, "Ontology-guided social media analysis System architecture," accepted at the SCOE 2012, ICEIS, 2012.

[6] L. Backstrom, P. Boldi, M. Rosa, J. Ugander, and S. Vigna, "Four Degrees of Separation." 01-Nov-2011.

[7] C. S. Jensen, "Temporal Database Management," vol. 1, pp. 32:1–15, 2000.

[8] P. Erdős and A. Rényi, "On the Evolution of Random Graphs," in *PUBLICATION OF THE MATHEMATICAL INSTITUTE OF THE HUNGARIAN ACADEMY OF SCIENCES*, 1960, pp. 17–61.

[9] C. C. Aggarwal, Ed., *Social Network Data Analytics*, 1st ed. Springer, 2011.

[10] P. Kazienko, K. Musial, E. Kukla, T. Kajdanowicz, and P. Bródka, "Multidimensional Social Network: Model and Analysis," in *Computational Collective Intelligence. Technologies and Applications*, vol. 6922. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 378–387.

[11] B. George and S. Shekhar, "Time-Aggregated Graphs for Modeling Spatio-temporal Networks," in *Advances in Conceptual Modeling - Theory and Practice*, vol. 4231, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 85–99.

[12] M. R. Henzinger and V. King, "Randomized fully dynamic graph algorithms with polylogarithmic time per operation," *Journal of the ACM*, vol. 46, no. 4, pp. 502–516, Jul. 1999.

[13] C. C. Aggarwal, Yuchen Zhao, and P. S. Yu, "Outlier detection in graph streams," in *2011 IEEE 27th International Conference on Data Engineering (ICDE)*, 2011, pp. 399–409.

[14] M. Erwig, R. Gu¨ting, M. Schneider, M. Vazirgiannis, M. Erwig, R. Gu¨ting, M. Schneider, and M. Vazirgiannis, "Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases," *GeoInformatica*, vol. 3, no. 3, pp. 269–296, 1999.

[15] "neo4j: World's Leading Graph Database," 2012. [Online]. Available: http://neo4j.org/. [Accessed: 19-Mar-2012].

[16] "Network Data Model Overview." [Online]. Available: http://docs.oracle.com/cd/B28359_01/appdev.111/b28399/sdo_net_concepts.htm. [Accessed: 10-Apr-2012].

[17] S. Fortunato, "Community detection in graphs," *arXiv:0906.0612*, Jun. 2009.

[18] J. Sun and J. Tang, "A Survey of Models and Algorithms for Social Influence Analysis," in *Social Network Data Analytics*, C. C. Aggarwal, Ed. Boston, MA: Springer US, 2011, pp. 177–214.

[19] B. Liu, B. Liu, and F. Menczer, "Web Crawling," in *Web Data Mining*, Springer Berlin Heidelberg, 2011, pp. 311–362.