2012 IEEE 36th International Conference on Computer Software and Applications Workshops

# Development of InfoVis Software for Digital Forensics

Dr Grant Osborne
School of Computer and Information Science
University of South Australia
e-mail: grant.osborne@unisa.edu.au

Dr Benjamin Turnbull
e-mail:Benjamin.Turnbull@gmail.com

Prof Jill Slay
University of South Australia
Polytechnic of Namibia
e-mail: Jslay@polytechnic.edu.na

***Abstract*— Information Visualisation techniques are one method that may be used to combat the growing complexity and data sizes associated with digital forensic investigations. This work outlines the processes, challenges, trials and tribulations of developing proof-of-concept forensic software designed to create interactive Information Visualisations from digital evidence sources.**

***Keywords-Digital Forensics, Software Engineering, Information Visualization, InfoVis***

## I. INTRODUCTION

It has long been established that the technical evolution of devices capable of storing digital evidence is outpacing the development of the toolsets used by digital forensic analysts and technician to analyse them [1, 2, 3, 4]. Both the complexity and volume of digital evidence are growing rapidly. The complexity of digital evidence refers to the heterogeneous and idiosyncratic nature of digital evidence. Information that practitioners must review is spread across networks or multiple devices each with their own unique way of storing and presenting data. The volume of digital evidence refers to large amount of binary data required for analysis in a given case. This volume of data will continue to grow as cheap storage, faster internet and consumer devices capable of storing large quantities of information become more pervasive in our digital lives. The literature in the domain supports these as key issues for analysis for study.

It is against these issues that we have turned to Information Visualisation techniques. Information Visualisations are adjustable, interactive mappings of abstract data to a visual form [5, 6, 7]. Information Visualisations connect the language of the eyes (shapes, colours and animation) with the language of the mind (relationships, processes, models and behaviours). The use of Information Visualisation techniques on digital evidence data facilitates a way to addresses the "information overload" faced by investigators when attempting to analyse digital evidence [7]. It also provides an efficient and visually appealing method with which to present findings of analysis. Information Visualisations empower technically skilled analysts and non-computer savvy investigators alike to be "data detectives"; highlighting patterns and connections that matter to them within a set of digital evidence. It enables a compression of potentially millions of sources of information into a one graphical view. These graphical views require less cognitive effort to be understood by the user when compared to textual displays.

This work details the process of developing software based on a framework developed specifically to integrate Information Visualisation techniques into the field of digital forensics. The Explore, Investigate, Correlate framework (EIC) was until this point theoretical, and there was a need to test it. To do so required codifying the concepts into software. This paper discusses that process.

## II. AN INTRODUCTION TO THE EIC PROCESS

This software was specifically developed to visualise events extracted from digital evidence sources; particularly social interactions between entities. The entities within the dataset represent people or places, connected via events (such as an Email or Phone Call). This method of representing data was chosen as it provided a way to homogenise information from multiple datasets extracted from different digital sources. One of the primary purposes of this work is to abstract the technical details from the context whilst still allowing access to the raw data as required.

Rather than developing yet another visualisation tool, this software focused on implementing the EIC Process [8]. The EIC process is grounded in information visualisation best practice. Furthermore, it takes into consideration the context of a digital forensic investigation, specifically the tasks and goals. The software presents the user with an Explorative view (overview visualisation) of the information first, enables them to investigate (drill down, focus, and refine) with common digital forensics tasks; and finally shows the relationships and behaviours within the dataset (through a correlative visualisation technique). Through this, the software implements the three main phases of the EIC process (as shown below in Figure 1). These phases are the explore, investigate and correlate phases (thus EIC).

978-0-7695-4758-9/12 $26.00 © 2012 IEEE
DOI 10.1109/COMPSACW.2012.47

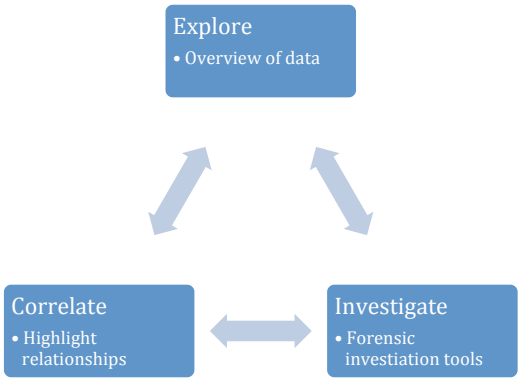

Figure 1 Three phases of The EIC Process

### A. *Explore Phase*

The explore phase is the first component of the EIC process. Within this phase, there are three tasks. The purpose of these tasks is to determine the property that the user wishes to provide an explorative visualisation of and to select an explorative visualisation technique from the dataset of this category of techniques.

The primary purpose of the Explore phase is to provide an overview of the types of data that exist in a given data source and to provide some initial focusing and filtering of the dataset. The goal of the Explore phase of the process model is to provide an investigator with a rapid overview of the data on a digital evidence source and to enable them to focus on information related to a particular crime.

This phase involves the user reading the properties of the dataset to determine those they wish to visualise at a high-level. Such properties could include a categorisation of the documents on a system by type, author, date created or digital evidence sources. Other dataset properties could be an overview of the different file types or events that have occurred across multiple devices. Once the evidence properties are determined, the user must select an explorative visualisation technique with which to render them.

The overarching goal of this phase addresses includes:

- Making the evidence visible,
- Reducing the relative size of the evidence and
- Providing high-level overviews of evidence.

### B. *Investigate Phase*

The Investigate Phase provides investigators with tools required to focus the current visualisation technique and filter out unwanted data. This is achieved using common digital forensics analysis techniques such as filtering of file types through properties such as extension, size, author and creation times. In addition to filtering of data types, digital forensics searching tasks can also be undertaken on the dataset that is being visualised, in order to help focus and enrich the visualisation.

The Investigate Phase begins when a user wants to "dig deeper" into the digital evidence. The user can flow in and out of the Investigate phase when the data is filtered to a satisfactory level. They may choose to move to the Explore or Correlate phase. The purpose of this is to allow the investigator to focus the dataset, then move onto a different visualisation or update their current visualisation in real time.

The Investigate Phase helps to address the following goals of the EIC process:

- Reducing the relative size of the evidence,
- Provide explanations of the origin or significance of evidence,
- Providing support to help identify items of probative value, and
- Reconstruction of events and relationships.

The investigation filtering tasks enable the user to reduce the size of the evidence that is displayed by the shared visualisation components. The searching tasks of the Investigate Phase help to identify items of value to an investigation. The searching tasks also enable investigators to identify events and relationships within the visual database.

### C. *Correlate Phase*

The Correlate phase is executed after the user has progressed through the Explore phase and the Investigate phase at least once. As discussed previously, within the Investigate Phase a user can filter and search the database and eventually make a choice to switch visualisation techniques. The Correlate Phase is entered after a user decides to switch to a correlate visualisation technique. This choice is based objectively on the user wishing to dig deeper into the data by progressing away from explorative visualisation techniques.

The goal of the Correlate Phase of the EIC process is to enable investigators to identify the relationships, behaviours and processes that exist within a set of visual data. The tasks within this phase are similar to the Explore Phase, although there is a major difference in the types of techniques and evidence properties that are selected for visualisation. The evidence properties focus on important links between individual nodes within the visual database, rather than aggregations of the overall content. As a result, the visualisation techniques are also more focused on the presentation of networks, links and multidimensional information as opposed to high-level overviews.

The Correlate Phase facilitates the following EIC process goals:

- Reconstructing the events and relationships that exist in the evidence,
- Providing explanations of the origin or significance of some evidence,
- Providing support to help identify items of probative value and
- Facilitating the presentation of findings to other investigators or in court.

## III. DEVELOPMENT TECHNOLOGY DECISIONS

Given the deployment environment, a state policing agency Electronic Crime department [9], it was decided to use a Web Base application to deliver the software to users. The benefits here are mostly administrative; it could be assumed that end users would have a Web Browser, but no

other software installed. Given the system administration overhead required for installing a client on multiple machines, and the fact that the software might require upgrading during the installed period, it was much easier to build the application as a web based tool. There were some disadvantages in this decision; namely the lack of fully developed visualisation toolkits network latency and the fact that use of different browsers might alter the user experience (due to inconsistencies in the HTML5 and CSS3 standards). However, as this software was proof-of-concept and was to be tested in a client location, eliminating the need for multiple installs was the pragmatic decision.

Given the research prototyping nature of the task, there was a need for a programming language allowing for quick development. For this, Ruby[1] was chosen. Ruby is dynamic object oriented programming language that provides a wealth of external 'gems' that make development fast and fairly painless. It was decided that if the speed of specific processes became an issue, core components could be rewritten at a later date.

Given the decision to use a web front end, the Ruby on Rails web application framework was used [2]. Rails implements the Model View Controller pattern and proudly touts a 'Don't Repeat Yourself' (DRY) mantra. The framework enabled the datasets required by the EIC process to be represented easily (using Models) and visualisations to be built on top of this (retrieving data through controllers that output data as JSON). Rails was chosen over other web application tools (such as Sinatra) due to its ability to easily represent data using Active Record (through Models).

HAML[3] and SASS[4] were also used as time savers, but are not strictly necessary unless you place a price on sanity. These enable the markup of the various layouts used by the tool to be developed in a much cleaner way, enabling more time to be spent on developing interesting visualization techniques, and less time spent reinventing the wheel.

Given the data volumes of metadata to be stored in an organized fashion, the need for a database was obvious. The proof of concept program has one. It was decided to use Sqlite3, as it is the default module used by Rails. On deployment, switching this backend to another database such as MySQL, would be trivial – as Rails uses the models developed to create the schema required. The database holds all the data after the initial processing and is heavily polled. As such, it is heavily indexed, making it slower for insertion and alteration of data but much quicker for reading and searching.

The Visualisations used in the software were implemented using JavaScript, taking advantage of the Canvas element provided by browsers that support HTML5. The primary toolkits used were the JavaScript Information Toolkit (The JIT[5]) and JQuery[6]. The JIT was extended to include more information on demand (such as hovering over nodes) and other interactive tasks. JQuery was used to provide AJAX requests to the Rails-based server side code, to stream in data for visualisation. Furthermore, it was used to handle updating the page in real time based on user filtering in the Investigation Phase, by streaming data in from the server as JSON and then injecting this into the current visualisation (effectively providing real-time updates based on user interactions and filtering).

## IV. Software Workflow

The first step of the software tool that implemented the EIC process was to import data. In this task, the user imports a set of visualisable data and stores it in the Sqlite3 "Visual database". The visual database represents a database of the current set of digital evidence ready to be focused, filtered, searched and visualised by the EIC process.

In the proof of concept software, data within the system is represented as a series of events that could have been extracted from digital evidence sources. These include Email, SMS and Phone Calls events, with a source and destination person. The implementation provided a simple way for users to create events via a web interface.

### *A. Explore Phase Implementation*

The implementation of the Explore phase of the EIC process was initially the most challenging, as it required the development of key infrastructure that was then used in other stages; specifically, the development of a visualisation library. Specific visualisation techniques were chosen that provide an overview or aggregation of data for initial filtering. Examples are shown in Figure 2 sand Figure 3.

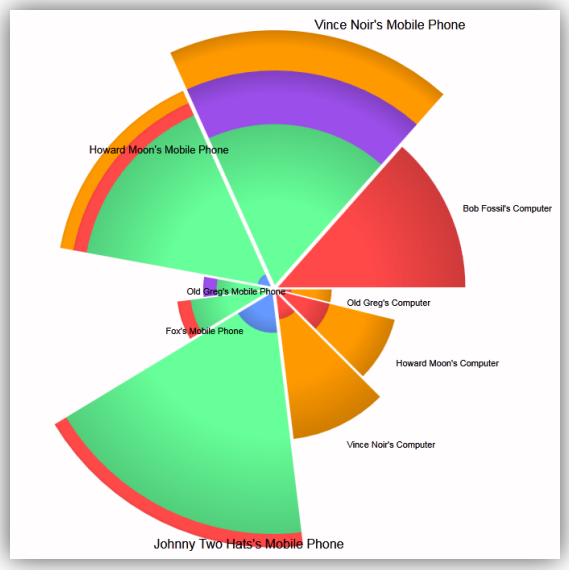


Figure 2 Sunburst Visualisation

[1] http://www.ruby-lang.org/
[2] http://rubyonrails.org/
[3] HTML Abstraction Markup Language - http://haml-lang.com/
[4] Syntactically Awesome Style Sheets - http://sass-lang.com/
[5] Nicolas Garcia Belmonte - http://thejit.org/
[6] http://jquery.com

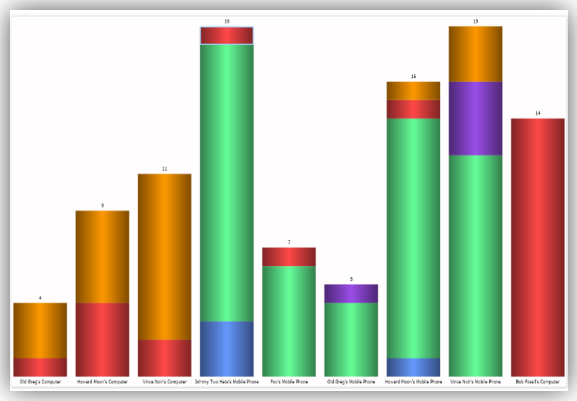

Figure 3 Barchart Visualisation

### B. *Investigate Phase Implementation*

The Investigate Phase provides investigators with tools required to focus the current visualisation technique and filter out unwanted data. This is achieved using common digital forensics analysis techniques such as filtering of file types through properties such as extension, size, author and creation times. In addition to filtering of data types, digital forensics searching tasks can also be undertaken on the dataset that is being visualised, in order to help focus and enrich the visualisation.

The proof of concept implementation of the EIC process provides the user with access to a set of filters. These filters enable investigators to show or hide events based on types, sources device and source suspect within the visual database. In addition, investigators can set an event date range that will hide events that do not occur between the earliest and latest dates. Upon entering the investigate phase for the first time, the evidence is entirely unfiltered and an explorative visualisation will be used to render the visual database.


Figure 4 Filters software implementation

When the user enters the investigate phase and chooses to focus the visualisation with the "Filter and Focus based on Dataset Properties" task, they are able to (in the proof of concept) disable or enable entities, event types, sources and enter a date range. Figure 4 provides an example filtering of the visual database, showing that four suspects have been removed from the view and that only email events will be shown.

### C. *Correlate Phase Implementation*

The proof of concept correlation visualisations includes implementations of the fundamental principles defined as the foundations for the EIC process. The fundamental principles implemented include the ability for a user to interact with the graph, by clicking and dragging nodes, zooming and panning the interface, removing items and finally the ability to obtain details of the source of the evidence on demand. The details on demand implementation can be seen in Figure 5. The details of the events coming into an out of a node (In this example the suspect named "Fox") are displayed to the user in textual form, containing all of the explicit details from the visual database. By providing these details, an investigator is able to extract the required information to investigate the digital evidence source from which the event was extracted.

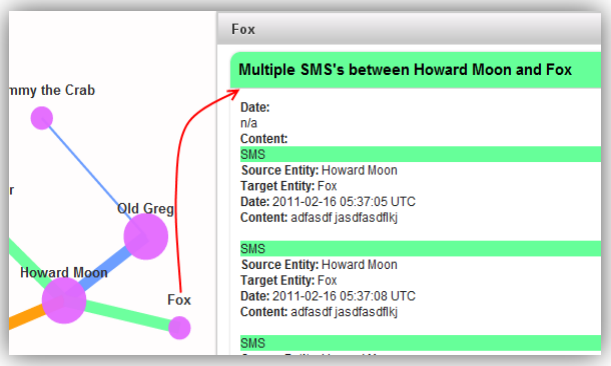


Figure 5 Details on demand

## V. LESSONS LEARNED

Beyond the creation of a software artifact that could be used to test our hypothesis, there were several valuable lessons learned in the creation of this software. Some of these are lessons in implementation; some may frame larger research questions.

### A. *Lessons in Implementing InfoVis*

Originally, the EIC software made use of the Prefuse visualization toolkit[7] [10]. However, in testing it was found that the performance required could not be achieved at the data grew. Even as proof-of-concept software, there was always the requirement to work with moderately large data sets and Prefuse has a high overhead and did not scale well as the amount of data grew. At the time of development, Prefuse had a small community following it, and was still in active development. However, it is the opinion of the researchers that it was still unpolished. Prefuse was tested but it became a bottleneck in the program. Eventually the decision was made to replace the visualization engine with the JavaScript InfoVis Toolkit. The JavaScript InfoVis Toolkit (TheJIT) is less complete than Prefuse in many ways, but lighter weight. Particularly of interest to the authors was the fact that TheJIT did not have capacity to

[7] http://prefuse.org

filter data at the visualization level. To replace this functionality, custom code was developed to stream filtered event data into HTML as JSON to allow the visualisations to update in real-time.

During the initial development phase of this software, D3.js [11] had not yet been released. Some of the visualization choices may have been different had this existed; D3 appears to have a lively community, be under active development and be more efficient in its use. D3 also conforms to Javascript norms in its use. However, without testing, the authors cannot know if D3 is a viable visualization engine for our purposes.

### B. Lessons in Implementing Forensic Software

In many ways it was relatively easy to abstract away the challenges in forensic software, provided that:

- Links are made back to original files / systems at all times
- Auditing is possible
- Minimal alteration of data

We only touched original data sources only once, and even then only to extract the metadata we required. After the initial data ingestion, we held a referral back to the original source but never again dealt with it directly. This held with good forensic practice. Given that the developed application only used metadata, this was possible. However, if the scope of the application expands to file content, preprocessing everything to a database becomes more complex and time consuming. Heavy preprocessing is used by some commercial forensic products, but creates other issues.

### C. Lessons in Implementing InfoVis Software

Given the amount of data that is created and visualized in even small cases, early filtering of data is required both for user and technical reasons. If a user opens an application to information overload, they are less likely to continue using it. This is reflected by the EIC process - which dictates that an investigation of digital forensics data should start with an overview first, to enable the user to "explore" or get a feel for the data.

Another lesson learned was the importance of minimizing the risk of exploration for the user. This is well-backed in literature [7] but the ability to undo alterations to visualisations, move back to previous states and understand their workflow are important aspects for effective program use. Again this was supported by the tool, by allowing the users to easily un-filter and reintroduce deleted items.

### D. Other Weighty Thoughts and Initial Findings

As soon as you try to put intelligence into forensic software, there is user backlash. Even simple correlation must, at all times, point directly to the original files and drives that contain that information. Moreover, if any reasoning does occur, it must be transparent and obvious to the end user. This project did very little reasoning, really only doing data fusion and entity extraction, but even at this level, there has been some suspicion over the use of an unknown (and therefore untrusted) reasoning process. These are only preliminary findings based on personal observation though and further work is required to truly understand this.

## VI. Conclusion

This work detailed the creation of software that needed to both encompass information visualization best practices and also conform to digital forensic best practice. The outcome has been software that operates on digital evidence in a different way than current commercial offerings, providing a more intuitive understanding of the copious data found in the investigation of digital evidence.

## References


[1] N. L. Beebe, J. G. Clark, G. B. Dietrich, M. S. Ko, and D. Ko, "Post-retrieval search hit clustering to improve information retrieval effectiveness: Two digital forensics case studies," *Decision Support System,* 2011.

[2] B. Schatz, "Digital Evidence: Representation and Assurance," Information Security Institute, Queensland University of Technology, 2007.

[3] G. Mohay, "Technical Challenges and Directions for Digital Forensics," in SADFE: Proceedings of the First International Workshop on Systematic Approaches to Digital Forensic Engineering, Washington, DC, 2005, p. 155.

[4] E. Casey, Digital evidence and computer crime: forensic science, computers and the internet, 2nd ed.: Elsevier, 2004.

[5] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings In Information Visualization: Using Vision To Think*: Morgan Kaufmann Publishers Inc. San Francisco California., 1999.

[6] E. H. Chi, "A Taxonomy of Visualization Techniques Using the Data State Reference Model," presented at the IEEE Symposium on Information Visualization, 2000.

[7] B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations," in *IEEE Symposium on Visual Languages*, 1996, p. 336.

[8] G. Osborne, B. Turnbull, J. Slay, "The 'Explore, Investigate and Correlate' (EIC) Conceptual Framework for Digital Forensics Information Visualisation, 5th International Conference on Availability, Reliability and Security, Krakow, 2010.

[9] B. Turnbull, B. Blundell, R. Taylor, "Anatomy of Electronic Evidence:Quantitative Analysis of Police E-Crime Data", 4th International Conference on Availability, Reliability and Security, Fukuoka, Japan, 2009.

[10] J. Heer, S. K. Card, J. Landay "Prefuse: a toolkit for interactive information visualization". *Proceedings of the SIGCHI conference on Human factors in computing systems*: 421-430, Portland, Oregon, USA: ACM, 2005.

[11] M. Bostock, V. Ogievetsky, J. Heer, "D3: Data Driven Documents", IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis), 2011.