Enhancing the investigation of malware-related crimes using semantic technologies

Rodrigo Carvalho
Linacre College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2018

This thesis is dedicated to my family.

Acknowledgements

Firstly I would like to thank my supervisors, Michael Goldsmith and Sadie Creese, without whom this thesis would not have been possible. They have provided me the support and encouragement I needed throughout my Dphil and, most importantly, the freedom to explore my ideas.

I must also thank everyone at the Cyber Security Analytics research group. It was a honour having being part of it for three years and meeting all of its members. The have definitely made the whole experience of a Dphil at the University of Oxford most enjoyable. Special thanks goes to Arnau Erola, Ioannis Agrafiotis, Jason Nurse and Jassim Happa for being always available to help. They also proved to be dear friends, together with Adrian Duncan, Carolin Weiss, Maria Bada and Pedro Antonino. Finally, thank you Lorna Swadling and Katherine Fletcher for your administrative help!

I am also grateful to Andrew Martin, Maureen York and David Hobbs, on behalf of everyone at the Center for Doctoral Training in Cyber Security, for their enormous effort in making the CDT an outstanding reference in Cyber Security research worldwide.

My cohort (CDT-13) is not to be forgotten: they were such an amazing, diverse and fun group, and have provided me with a comprehensive perspective about Cyber Security. Thank you all, specially Eduardo dos Santos, Florian Egloff, Oliver Farnan, Vincent Taylor and Yudhistira Nugraha for becoming such special friends as well.

Last but not least, I extremely grateful for all the love and care provided by my family. It is impossible to mention everyone, but on behalf of my wife Stefanye (who have shared with me this whole Dphil journey, supporting me when I most needed it), my parents Luzia and João (who have always inspired me to follow my dreams), my brother João, my sister Mariana, my nephews Pedro Henrique and Luiz Eduardo, and specially my 95-year-old grandmother, Vó Fefinha, thank you all from the depth of my heart and soul.

This dissertation was funded by Capes - Brazilian Government, and supported by my employer, the Brazilian Federal Police. I owe a debt of gratitude to both institutions for making this project possible.

Abstract

The expansion of technology connectivity and the pervasiveness of data in our society pose both challenges and opportunities for the government and the private sector. Big companies like Google and Facebook are in the forefront of successfully tackling the challenge of extracting meaning from this data deluge: building rich profiles of people and networks enable them to monetise and make profits by selling such profiles for targeted marketing purposes. For most organisations, though, the challenge of generating actionable intelligence from the available data sources is still daunting.

In the government sector, one of the sectors that could benefit significantly from data-driven intelligence is that of Law Enforcement. However, the deficit of specialized personnel and tools which extract meaningful information from data (as Chapter 1 shows) is directly linked to weak investigation capabilities, ultimately hampering catching serious organised organisations.

As the literature review shows, the available forensic tools are just starting to change the focus from improving processing performance to facilitating investigation and exploration. One example is the increasing adoption of domain taxonomies to describe data.

This thesis addresses the capability gaps by demonstrating that analysts working in law enforcement would benefit from an data exploration tool leveraging specific semantic features. In addition to semantic search and integration of data (features already provided by many semantic data exploration tools), allowing the investigators to materialise classes, object properties and datatype properties could help them shaping their knowledge during the course of an investigation. Moreover, the ability of expressing knowledge in terms of semantic queries and rules could enhance information exchange between analysts.

A prototype was developed to assess the feasibility of the idea and validate it with actual investigators. Their feedback after testing the prototype indicated that such computer-provided features could indeed support the reasoning of the human analyst, making cybercrime investigation more efficient.

Contents

List of Figures

List of Tables

Acronyms

BI Business Intelligence. 3

EXIF Exchangeable Image File. 12, 18

GUI Graphical User Interface. 9, 34–36, 46, 90, 106, 107, 117, 118, 121–123, 125, 126

IOC Indicator of Compromise. 3, 4, 6, 22, 25, 43, 51–53

LEA Law Enforcement Agency. 3, 4, 11, 26, 29, 50, 124

OSINT Open Source Intelligence. 22, 44, 109

P2P Peer to peer. 28, 29

POC Proof of concept. 9, 45, 46

RDF Resource Description Framework. 32, 34, 35

SPARQL SPARQL Protocol and RDF Query Language. 32, 35–37, 40, 47, 64, 99, 101

SQL Structured Query Language. 64, 99, 108

SWRL Semantic Web Rule Language. 21

TTP Tactics, Techniques and Procedures. 20

UX User Experience. 86

Chapter 1

Introduction

1.1 Digital forensics

Digital forensics is an activity dedicated to recovering and investigating information from digital devices which could inform a trial about the occurrence of a crime. Despite being a relatively new practice (digital forensics is roughly 40 years old), it is currently facing many challenges due to fundamental changes in the computer industry [1].

Not so long ago, digital evidence about a whole criminal organization would be contained within a handful of devices, comprising mostly desktops and laptops. The *cloud* was not commercially implemented yet and mobile phones did not store anything but contacts, call logs, SMS messages and perhaps few notes. Therefore, the few devices seized during a search warrant were rather relevant, as they would contain a wealth of criminal evidence that could completely incriminate the device's owner, or whoever was proved to have had access to it. Moreover, the volume and variety of data stored in such devices was limited, and recovering it was not a difficult task.

Currently, the situation is quite different, as a single citizen might own multiple digital devices, such as smart-phones and tablets, which are frequently associated with other devices and constantly connected to multiple cloud services (e.g. social networks and e-commerce). In addition to intensifying the problem of scale in digital forensics [2], the current cloud context means that evidence could be found in storage devices that cannot be seized, or are in control of an entity outside the jurisdiction of the investigating country. These and other challenges have motivated the development of novel research areas such as *cloud forensics* which explores the technical, organizational and legal aspects of the cloud paradigm. Nonetheless, there is still little research emphasis on improving the cybercrime investigation techniques which could lead to identifying potentially relevant evidence, whether stored in the cloud or not.

1.2 Malware cybercrime

This expansion of connectivity within our society is very related to the evolution of cybercrime [3]. It not only increases the attack surface in terms of entry points to systems, but the resulting data and meta-data associated with activity in cyberspace provides opportunity for more sophisticated targeting of victims.

General cybercrime can be divided in two basic categories [4]:

- *Cyber-enabled crimes*, or crimes that are critically enhanced by leveraging Information and Communication Technology (ICT), but could still be committed without them. One good example is banking fraud: although phishing[1] scams boost reaching potential victims, this crime can still be committed by deceiving someone over a phone call.

- *Cyber-dependent crimes*, or crimes that can only be committed using an ICT device. Some examples are intrusion to computer networks and disruption of ICT infrastructure.

Most *cyber-dependent crimes* and some *cyber-enabled crimes* critically depend on the ability of a criminal to install malware [2] on a victim's device [5]. And the necessary specialized skills to do so, which could act as a natural deterrent on the growth of such crimes, are not an obstacle any more. After all, there are many places on the Internet in which cybercriminals trade malware and infrastructure services, such as botnet[3] rental, at competitive prices, see [6].

Thus, in addition to the expansion of connectivity in our society, this ever-growing commoditization of the malware ecosystem also help to explain the expansion of computer attacks [7], which are reflected in surveys from organizations such as the Anti-Phishing Working Group(APWG) [8]:

- "The total number of unique phishing sites observed in the second quarter of 2016 was 466,065. This was an all-time high."

- "APWG member PandaLabs found 18 million new malware samples in Q2, an average of more than 200,000 a day. This is 10 percent lower than in the previous quarter, when 20 million new samples were found."

[1] Phishing is the attempt to acquire sensitive information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), often for malicious reasons, by masquerading as a trustworthy entity in an electronic communication. *Source:* `https://en.wikipedia.org/wiki/Phishing`, *accessed in* 26/04/2016.

[2] Malware, short for malicious software, is any software used to disrupt computer or mobile operations, gather sensitive information, gain access to private computer systems, or display unwanted advertising.*Source:* `https://en.wikipedia.org/wiki/Malware`,*accessed in* 26/04/2016.

[3] A botnet is an interconnected network of computers infected with malware without the user's knowledge and controlled by cybercriminals. They're typically used to send spam emails, transmit viruses and engage in other acts of cybercrime. *Source:* `https://usa.kaspersky.com/internet-security-center/threats/botnet-attacks`, *accessed in* 27/04/2016.

Despite these alarming figures, the investigation of malware cybercrime (i.e. analysing the relationships between different malware campaigns which could lead to the criminals behind them) is still a recent and unexploited challenge, especially from an academic perspective. Empirical research in this area is rare, and innovation is mostly driven by commercial companies [9]. For instance, AV vendors, who form a large part of the security community: even though some of them have created dedicated teams to investigate malware campaigns on the internet, most of them focus on detecting and black-listing the malicious files and web servers distributing them. This makes sense since their main commercial focus is securing the computers of their clients, and not investigating the associated campaigns.

Threat intelligence companies tried to fill this gap in investigation by providing advanced dashboards containing a wealth of information about Indicators of Compromise[4] (IOCs) collected from multiple sources. Fireeye Mandiant, for instance, is one of the most well-known companies in the field of threat intelligence. In addition to providing the client with information from multiple sources such as uncovered vulnerabilities and profiles from threat actors, it also offers faceted search capability, comprising a fixed range of threat intelligence concepts: actors, malware, target industries and more.

Notwithstanding their importance, a recent survey about these platforms indicated that they primarily focus on collecting and sharing IOCs instead of analysis, which is often limited to searching, browsing and attribute-filtering [9]. On the other hand, the actual investigation of malware cybercrime normally involves tasks which modify data such as grouping artifacts from the same campaign together or taking notes which could help a fellow investigator. As pointed in Section 2.4, there are plenty of platforms with rich dashboards and powerful BI capabilities directed to the security teams of client corporations (such as Fireeye Insight Threat Intelligence), but very few aiding the industry analysts in conducting the malware investigations which will feed such platforms.

Determining the attribution of a malware campaign is typically the responsibility of Law Enforcement Agencies (LEAs) because it is a criminal act. Yet, LEAs have not developed enough in terms of personnel, methods and tools to deal with the increasing investigative demand. Therefore, most agencies focus on disrupting the computer infrastructure used to spread malware. Operation Avalanche [10] is a one of the most recent documented cases.

Although demanding a major cooperative effort spanning several jurisdictions, disruption is an effective strategy to stop malware distribution at large scale, ultimately leading to increasing the costs of cybercrime activity. However, criminals on the loose can always update their Tactics, Techniques, and Procedures (TTP) (e.g. hiring different infrastructure and acquiring upgraded malware), in a

[4]Indicator of compromise (IOC) — in computer forensics is an artefact observed on a network or in an operating system that with high confidence indicates a computer intrusion. *Source:* `https://en.wikipedia.org/wiki/Indicator_of_compromise`, *accessed in* 07/05/2017.

continuous “cat and mouse” game. On the other hand, more productive investigations could not only bring them to justice but also act as a deterrent.

1.3 Cybercrime investigation

One of the most challenging issues in investigating a malware campaign is producing comprehensive evidence against its perpetrators. Often, probative and court-admissible evidence can only be found after seizing the devices owned by the criminal organizations. Obtaining such devices is a complex task, even if they are within the same jurisdiction of the LEA: investigators must hypothesize about a high volume of heterogeneous, supposedly unrelated IOCs for patterns and relationships which could eventually support a search warrant.

However, a United Nations global study on the topic states that less than one per cent of the total police officers in all countries surveyed are specialized in investigating cybercrime. [3, p.154]. Indeed, *General cybercrime investigations* and *Computer forensics & evidence* were the most commonly reported topics in a survey regarding technical capacity building [3, p.179].

There is an important distinction between those two topics (General cybercrime investigations and Computer forensics), as they belong to different phases of a criminal case. The former involves traditional police investigators with varying degrees of expertise in computer science who, following the reporting of a *notitia criminis* (e.g. a forum post advertising the sale of fake passports), start searching for clues indicating whether the reported crime effectively happened and whether it is worth investigating.

Once a official case is opened and the cybercrime investigator has collected enough information justifying a judicial warrant, the digital devices of the suspect are seized and submitted to the computer forensics lab. That is when the computer forensics analysts join the case. They are responsible for analysing all devices in search of sound evidence about authorship and materiality regarding the crime reported in the *notitia criminis.*

This perceived lack of computer-specialized personnel mentioned in the United Nations study could lead LEAs to allocate digital forensics analysts (who would only normally be demanded to examine seized devices)to other non-forensic but important tasks. Some examples include helping operations planning (which the author has already participated within the Brazilian Federal Police), tools development [11] and staff training [3, p.156]. Ultimately, some digital forensics analysts could take responsibility for cybercrime investigation, which is a flagship service provided by LEAs. In addition to their digital forensics diligence, the computer science background from such analysts could greatly improve investigating highly-technical cybercrime such as malware-related crimes. Moreover, well-designed investigations could make the digital forensics process more efficient, whether in device triaging (seizing less, more relevant devices) or in providing historic insights which could facilitate

the digital forensic analysis (which words to search for, password guessing for dictionary-attacks and others).

Notwithstanding computer forensics analysts start conducting cybercrime investigations, there would still be not enough investigators to fight back the increasing rates of cybercrime and data to analyse. More than ever, "Today's tools must be re-imagined to facilitate investigation and exploration" [1]. This is precisely what the present thesis proposes: a method comprising a tool which assists cybercrime investigators not only in querying data, but in actively assessing and enriching it with their own investigative knowledge in a structured, automated and collaboratively way.

1.4 Research topic

Investigation is essentially a human skill, which might explain the following distinction: the computer deals with the processing, and the human with the reasoning. This is true for most current digital forensics and threat intelligence frameworks, which favour data extraction and processing rather than supporting analysis. The literature review contains more details about them.

However, the current stage of development of semantic technologies (as discussed in Section 3.3) suggests that this distinction is not as clear as before. Aiming at a better balance between human reasoning skills and computer processing capabilities, this thesis will experimentally establish whether semantic technologies could make the investigation of malware-related crimes easier to the investigator. More specifically, the hypothesis is that the investigation needs listed in Table 1.1 (and further discussed in Section 2.4), once addressed by the correspondent semantic capability in the same Table, could facilitate investigative tasks such as hypothesis testing and insight generation.

Table 1.1: Mapping needs of an investigation to compatible semantic capabilities.

	Common investigation needs	**Compatible semantic capabilities**
1	Clustering entities	Defining classes with property restrictions
2	Establishing links	Creating object properties
3	Inserting tags or comments	Creating datatype properties and updating their values
4	Enriching data about an entity	Materialising the object property "owl:sameas"
5	Rolling back in case of dead end	Restoring previous version of KB into the triplestore
6	Reproducing investigation steps	Reapplying rules to the same dataset
7	Sharing investigative knowledge	Exchanging queries and rules
8	Exploring the dataset	Semantic facet query and graph browsing
9	Dataset integration	Bespoke ontologies and graph merging
10	Establishing provenance	Namespaces and URIs

There is a paper describing how features 8, 9 and 10 from Table 1.1 have improved the investigation of human trafficking in a law-enforcement setting [13]. Nonetheless, this thesis will demonstrate that providing the investigator with the other seven semantic capabilities as well could further enhance data exploration and insight generation within the malware investigation domain.

Features 1 to 4 are mostly used for building knowledge bases before the users of linked-data exploratory systems can browse and search them. This holds true for most systems described in [14] which, despite sharing some similar objectives with our prototype, do so using different techniques. For instance, using in-session memory to memorize the user browsing sequences and suggesting queries to inspire exploration.

The novelty of the approach proposed in this thesis lies in actually allowing the investigators to manipulate these features in order to shape knowledge which would inform their investigative hypotheses. Feature 5 and 6 refer to navigating distinct versions of the knowledge base, which could be useful for restoring or resuming investigations to speficic milestones. They are relevant in the sense that the bottleneck observed when using threat intelligence platforms derives from the multiple manual tasks that the user has to perform [9]. Finally, feature 7 would suit the reported need for producing and sharing intelligence instead of raw IOCs.

In order to assess this approach, the following questions were considered:

1. **Feasibility and suitability**: Could the semantic features in Table 1.1 be used to perform the corresponding tasks in a real investigation scenario?

2. **Learning curves**: How difficult would it be to people with a background in cyber-security to operate a prototype implementing such features and understand the underlying technology?

3. **Effectiveness and efficiency**: Do this semantic approach increases the flexibility of the expert investigators in exploring and manipulating data, when compared to their current tools and methods?

Different research methods were used to assess each research question, which are the main themes of Chapters 5, 6 and 7 respectively. In addition to justifying why these specific methods were employed, the next section will also discuss the role of each of the others chapters in the development of this thesis.

Running these assessments critically depend on the development of a prototype implementing all the features listed in Table 1.1. After all, despite the fact that there is already research on how semantic technologies could improve digital forensics and cybercrime investigation processes, very few prototypes have been developed to demonstrate that (Section 2.3). In addition, to the best of the author's knowledge after conducting the literature review, none of such prototypes leverages the semantic capabilities the way this thesis does.

Thus, the contribution to science provided by this thesis is the extension of an existing technique: semantic technologies as a method for data exploration. The tool developed during this thesis extends the range of application [15] of such method by providing the end-users with the capability of materialising their own knowledge (in the form of new classes, object properties and datatype

properties) and also integrating different datasets during the course of an investigation, using a prototype. Furthermore, it will be demonstrated how semantic rules can provide a better structure to the whole investigation in terms of reproducibility of tasks and knowledge sharing.

1.5 Structure of this thesis

The chapters of this thesis are organized as follows:

Chapter 2 – Literature review

The literature review chapter starts by exploring traditional digital forensics frameworks in the light of the challenges brought by the huge increase in data to process and analyse. Next, it discusses some evidence representation schemas which were suggested to facilitate these analysis. In addition to peer-reviewed literature, *grey* literature is very relevant for the malware-cybercrime topic, as it has not gained a lot of attention from an academic perspective yet.

Thus, the literature review also makes reference to investigation reports, white papers and informal interviews with two experts from antivirus companies which have implemented dedicated teams to do conduct such investigations. These have proved useful in many ways: to verify that the investigation needs specified in Table 1.1 are indeed relevant, to reveal the limitations of current methods and tools used to investigate malware-cybercrime; and to define some requisites for a prototype aimed to facilitate intelligence generation.

Finally, the literature review lists a few initiatives applying semantic technologies to digital forensics and cybercrime investigation, suggesting that other researchers also realised the potential of such technology in this domain.

Chapter 3 – Semantic cybercrime investigation

Having confirmed some gaps in current investigative approaches and tools (such as poor analysis support, lack of automation, difficult integration with distinct datasets) the aim of Chapter 3 is to present the initial questioning that motivated this thesis. Based on the literature review and the experience of the author in conducting digital forensics analysis for more than seven years within the Brazilian Federal Police, this chapter makes some considerations about how digital evidence is handled and consumed in a law-enforcement setting involving digital forensics analysts and crime investigators, and expand them to propose a context-enhanced evidence

The key to this model was considering both the criminal context (i.e. what the investigation wants to prove and how to investigators work) and the technical context (i.e. what kind of evidence could be retrieved and how they could be relevant to the investigation). This analysis suggested that, by providing contextual information to raw data (e.g. strings) which can be processed by computers, semantic technologies could indeed be effective in advancing not only digital forensics

analysis (as pointed out by some researchers already), but actually to the cybercrime investigation as well.

The conclusions of the exercise of building such a model led to a position paper discussing how ontologies could potentially enhance information searching and reasoning in the context of a criminal investigation involving digital evidence. It took the case of online banking fraud investigations, and suggested classes and relationships for technical data (handled by the digital forensics analysts) and non-technical data (handled by the crime investigator).

[1] R. Carvalho, M. Goldsmith, and S. Creese, "Applying Semantic Technologies to Fight Online Banking Fraud," presented at the European Intelligence and Security Informatics Conference (EISIC), 2015, pp. 61–68.

The same idea was discussed through a poster and a symposium presentation:

[2] R. Carvalho, M. Goldsmith, and J. R. Nurse, "Online Banking Malware Ontology," presented at the International Crime and Intelligence Analysis Conference (ICIA), 2015.

[3] R. Carvalho, "Semantic technologies applied to digital forensics analysis and evidence modelling," presented at the European Symposium on Research in Computer Security (ESORICS), 2015.

Further on, this chapter will also explain other native capabilities of semantic technologies, and how they would fit to cybercrime investigations using the context-enhanced evidence model proposed.

Chapter 4 – Implementation

The interaction between semantic technologies and digital forensics/cybercrime investigation is still very recent (as mentioned in Section 2.3). Being largely unknown to the wide public of cybercrime investigators makes it difficult to discuss the applicability of the former into the latter. In order to obtain an informed response from experts about whether semantic technologies could enhance malware investigations, a prototype was necessary Thus, this chapter will describe the implementation of the prototype in details:

- What technologies were chosen to develop the prototype and why;
- How each of the features listed in Table 1.1 were implemented, describing the potential challenges and explaining some design decisions;
- The evolution of the prototype during the course of this thesis, specifying the main updates.

Chapter 5 – Case study

The first assessment of the prototype comprised an exploratory case study. The objective was to validate the research question regarding the feasibility of applying the semantic features listed in Table 1.1 in a real investigation scenario. For that, a specific malware investigation report was

chosen (based on the arguments presented in Section 5.1), and the investigative steps taken were reproduced using the prototype.

This chapter will show that it was possible to reach the same conclusions of the authors by leveraging some of the native semantic capabilities from Table 1.1 (rows 1, 2, 3, 5 and 8). Furthermore, it will extend the investigation report by adding more data to the knowledge base in order to demonstrate the other implemented capabilities (rows 4, 6, 9 and 10 from Table 1.1). The case study described in this chapter was the subject of the two following papers:

[4] R. Carvalho, M. Goldsmith, and S. Creese, "Malware investigation using semantic technologies", presented at the Intelligent Exploration of Semantic Data workshop (IESD), 2016.

[5] R. Carvalho, M. Goldsmith, and S. Creese, "Investigating Malware Campaigns with Semantic Technologies", accepted for publication on the IEEE Security & Privacy magazine, special issue on Digital Forensics, Jan 2019.

The latter differs from the former on:

- Explaining semantic concepts to a broader audience;
- New insights about applicability to investigations;
- Demonstrating how the conclusions of the report could be easily extended by adding new data to the knowledge base;
- Conducting the investigation using a GUI.

Finally, this case study used real data (which was published together with the report) and did not involve any stakeholders operating the prototype other than the author. After all, the prototype was still on its first version (being more like a POC): applying the semantic features was not intuitive yet, the results needed to the manually analysed, and the lack of a fully-functional GUI meant it was not ready for third-parties to assess it yet. Having positively answering the empirical enquiry "Could the same analysis steps and results from the investigation report be reproduced by using the prototype?" validated the first research question.

Chapter 6 – Usability assessment

Chapter 6 describes an usability assessment of the prototype, aimed at answering the second research question. This assessment was designed to check whether the user could apply and understand the main features of the prototype and to identify any issues with the experiment design ahead of the main validation with the expert investigators (Chapter 7). In addition, this assessment also served to estimate the learning curve of the prototype. It was based on the malware investigation report described in Chapter 5 because of the knowledge obtained by the author of this thesis regarding it,

making it easier to identify potential difficulties and differences in the performance of each participant.

For that, ten participants were recruited among students and researchers from the Cyber Security Analytics Group at University of Oxford. The restriction to this department was because although the experiment does not require working knowledge about malware campaigns, a basic understanding of concepts related to cyber security such as "file payload" and "command and control servers" is recommended in order to create the necessary queries.

The chosen research methods used in this assessment, further detailed in Section 6.1, were:

- Collection of users opinions using pre and post-experiment questionnaires;
- Experiment tasks: necessary to estimate if the users could operate the prototype, and collect suggestions for improvement before the main validation;
- Observation and usage monitoring: logging(collecting clicks and time taken to complete the tasks), user observation (which tasks were more troublesome, if the users seemed to need help in executing the tasks) and think aloud protocol (to try to intrinsic issue not predicted by the experiment).

Chapter 7 – Expert testing

Chapter 7 is the main validation of the thesis, and will assess the fitness and the expressiveness of the prototype. For that, it will recruit police officers to execute a series of tasks resembling real online banking fraud investigations. The focus is not on the usability itself (besides not being the scope of this thesis, improving usability would require multiple iterations with the end users), but to demonstrate the semantic features of the prototype and get the users feedback about whether they could indeed be useful to their daily routine investigation tasks.

The research methods used were basically the same from Chapter 6, with the exception of the final questionnaire, which was divided into a semi-structured interview and a questionnaire, in an effort to identify if the participants really understood the prototype, and to obtain more complete answers to its discursive questions.

Even though the format was the same, the content had changed. The experiment tasks were designed according to the domain knowledge of the online banking fraud unit within the Brazilian Federal Police, and counted with the informal advisory of one investigator. Moreover, the questions from the questionnaire and the interview explored the daily activities of the investigators, trying to capture if the semantic features could make those more efficient.

Chapter 2

Literature review

Digital forensics emerged as a distinct activity around 40 years ago, and its main objective was recovering data from digital devices [1]. Therefore, most of the research in this field have been focused on improving related technologies, whether in terms of processing speed or information extraction.

However, the sheer volume of data currently stored in an ever increasing variety of devices has been pushing research into information correlation, which is mostly performed during the analysis phase. Thus, the following subsections will review academic and non-academic work regarding three related topics:

- Current digital forensics analysis frameworks, discussing their implementation and challenges;
- Suggestions for evidence representation, highlighting efforts in defining a common standard among practitioners;
- Semantic technologies applied to cybercrime investigation.

The critique and relationship between these three topics is expected to support the evidence model suggested in Chapter 3.

2.1 Research in digital forensics tools

In a recent United Nations study, respondents from law enforcement agencies (LEAs) recognized the increasing levels of cybercrime. Their argument was that "...both individuals and organized criminal groups exploit new criminal opportunities, driven by profit and personal gain." [3]. Consequently, LEAs and other cybercrime-investigation stakeholders are increasingly more dependent on electronic data, which is the result of the digitalization of our daily life [16]. Among this data there may be evidential information regarding criminal actions. If considered relevant and complete, digital evidence could be used in pre-trial and also court procedures.

Traditionally, all digital forensics tasks (from duplicating the storage device to producing the report) are performed by computer-forensics experts. The main argument is that only these would have the necessary expertise to access, search and interpret digital evidence while keeping the chain of custody. However, the current overload of devices to be forensically examined is causing a bottleneck in the evidence-production phase. At the same time as it raises the criticality of researching digital forensics, it also forces forensic practitioners to take alternative measures.

For instance, some law enforcement agencies are employing more people to better balance the whole digital forensic workflow. Similarly to the Brazilian Federal Police, the Dutch Federal Police implemented a framework [11] to acquire, process and make forensic images ready for analysis in a controlled environment, accessed remotely. The investigators are provided with a read-only interface with tools such as indexed searching and mailbox extractor. This allows them to search the seized storage devices for readily-available evidence as soon as previous automated processing tasks have finished. In addition to making the whole process more efficient, the investigation team (who are mostly interested in the information) would gain access to the evidence in the early stages of the forensic process. Then, if issues arise, whether interpreting a specific data type or gaining access to a internal business system, computer forensic experts are contacted to perform a more detailed analysis, regarding a specific set of information.

This work process maintains the chain of custody at the same time as it better balances the load of work. After all, investigators usually outnumber computer forensics experts, and have more knowledge about the case being investigated. The underlying technology supporting the Dutch framework is called XIRAF (XML Information Retrieval Approach to Digital Forensics) [17], developed by the Netherlands Forensic Institute. It integrates a range of both bespoke and publicly available forensic tools (e.g. registry parsers, EXIF extractors and carving tools) in order to process information commonly found in evidence storages (such as hard disk images).

XIRAF IS more than a mere collection of tools: it implements wrappers for standardizing function calls and data input/output among them. This operational layer allow, for instance, that different tools be called in sequence, automating the information processing phase. One example mentioned in the paper is the automatic creation of a timeline from the output of an EXIF extraction tool. Representing the whole set of information under a common standard has another advantage: it enables more expressive queries to be made. In the case of XIRAF, the results from all forensic tools processing are stored under a single XML file, and can be retrieved using XQuery language. Future work intends to augment XIRAF with expert knowledge on digital traces (e.g. the most important windows registry keys), which indicates the authors' concern in facilitating investigation rather than simply processing information.

Despite helping to tackle the backlog, such semi-automatic-processing approaches are still presenting the human analyst all information stored in a single device, which makes evidence-production

more time consuming and susceptible to missing evidence. A case with multiple seized devices makes matters worse: even if it is possible to make all of its images available at the same time, there is just too much information to relate and extract evidence from by the few investigators in charge. In other words, the problem is not being solved by a more intelligent use of computer capabilities to facilitate human insight generation towards relevant pieces of evidence. Instead, it relies on increasing automated processing and employing more analysts, who have to search for evidence across all available data.

Raghavan [18] also recognizes the challenges brought by the sheer volume of data to analyse. Based on his findings about current state of the art in digital forensics research, one of his conclusions is that the abundance of meta-data in digital devices should better benefit an analysis framework aimed at recognizing and associating evidence items stored on them. In this regard, the digital forensic corpora [19] can be very useful. In addition to network packets, memory images and files randomly downloaded from the internet, it comprises more than 2000 hard drives acquired in the second market and containing real data from people and organizations. By providing a standard dataset, the authors expect to foster research in digital forensics, making it easier to test and validate new models, tools and techniques.

Among the first experiments conducted with this dataset, Garfinkel tried to uniquely identify and correlate a subset of 750 hard drive images [20]. For that, he defined similarity features such as email messages' subjects and ID, US social security and credit card numbers, cookies and timestamps. Two analysis were then performed: comparing total instance counts from each drive, and applying a weighted scored function regarding identical instances found in different drives.

Besides identifying the most critical drives based on their level of correlation according to relevant features instances, another interesting outcome from such approach are the so called "CDA stop lists". They consist in instances of features that, being so ubiquitously found in all devices (e.g. email addresses present in X.509 root certificates), could be safely ignored when processing new hard drives. CDA stop lists could be used as an alternative to the KFF (Known File Filter), an effective technique to decrease the number of files displayed to the analyst. KFF identifies relevant or ignorable files during the processing phase by matching files' hashes with a previously specified hash list. On the other hand, the stop lists are more flexible than KFF, in the sense that it compares content features and does not need a input list beforehand. Nevertheless, if loosely defined, they could ignore relevant files.

There were not many correlations found by Garfinkel's experiment, but that might come from the high discrepancy of the analysed drives, obtained from distinct, non-related origins. However, this fact does not invalidate the article's claims that cross drive analysis could benefit the development of automated and intelligent forensic tools. The existence of previous information about a case involving

most probably related devices (which is common in a Law Enforcement setting) could better tailor features to be applied in smaller device subsets, possibly increasing the level of correlation.

This is the case from Marturana et.al. [21], who applied cross drive analysis to aid forensic device triaging in the context of an Italian police agency. Based on real datasets and investigations, they devised tailored features regarding two specific crimes: copyright infringement and child pornography. The former considered features such as the number of installed P2P, MP3 converter and crypto applications, as well as the quantity and size of ISO files found in a storage device. On the child pornography cases, the idea was to establish a criminal pattern based on smart-phone features such as number of downloaded and created image and video files, and also the time of sending and receiving phone calls and short message service (SMS) messages. This criminal pattern could differentiate phones used by child pornography suspects to the ones used by other crimes' suspects, such as extortion and human trafficking.

Besides measuring their performance, the authors state that the three classifiers used (bayesian network, decision tree and locally weighted learning) could correctly classify more than half of the 23 phones used in the test. Better results were achieved after expanding the training set to 21 devices and considering only independent features. In addition to using a larger dataset, the authors consider that understanding the "...plausible connections between investigated crime and potential digital evidence" could improve the classification results. This requirement is compatible with the general objective of this thesis: to embed the empirical experience of the investigator into the analysis tool.

The last two papers described useful approaches to relate the content and the metadata from files found in multiple similar devices, whether hard disks or smart-phones. Nonetheless its importance, regularly forensic analysts have to examine evidence from different types of information related sources, such as network capture files (PCAP) and memory dumps extracted from a single system.

For instance, Case et. al. [22] suggested that automatically parsing and correlating such information would result in a more coherent view of the system being analysed. One of their great contributions is *ramparser*, a memory analysis tool for Linux systems which recover detailed information regarding open files, running processes and sockets. This is a previous and necessary step to feed their correlation engine, which also receives input from other forensic tools such as system log and network capture parsers.

After being processed by the correlation engine, this information is displayed in so-called "thematic views" according to their binding to specific users and processes. Each view presents the investigator with links to more detailed information regarding others sources of evidence, for instance, by filtering the files opened by a specific user instead of displaying all files. Such a feature could enhance visual detection of a file that has been exfiltrated over a FTP connection. The thematic-views concept is a step towards forensic tools that make the investigation more intuitive

and interactive. Thus, they are aligned with the current trend of making tools to better assist in investigations rather than simply searching for evidence [1].

So far this review has discussed some interesting approaches for relating information stored in distinct devices, suggested by researchers. Although successfully tackling specific challenges of forensic analysis, most forensic practitioners and other stakeholders still adhere to traditional, all-in-one forensic frameworks, like the Forensic ToolKit (FTK) [23] and Encase [24]. These tools were designed in a time when most information to be analysed came from a small set of storage devices (i.e hard disks, CDs or USB drives), which were processed one device at a time. However, the multitude of devices where information can be stored today, especially considering the cloud and the internet of things paradigms, might lead to a change in how digital evidence is produced. Therefore, the next section will discuss current evidence representation schemes, which are vital for defining novel frameworks accounting for the diversity in data sources.

2.2 Evidence schemes: from XML to semantic tuples

With that in mind, Garfinkel proposed the DFXML - Digital Forensics XML in 2009 [25], and revisited it in 2012 [26], discussing its motivation, comparing it to similar initiatives and citing where it has been adopted. Similarly to XIRAF, it also advocates to storing the output from different forensic tools in a common XML format. However, it is not a framework comprising a tool repository: the main idea behind DFXML is to provide a standardized way for both storing and exchanging forensic information within the forensic community, whether practitioners or tool developers. Specifically regarding the latter, one good contribution is a Python API for manipulating data in DFXML format. For instance, functions to retrieve the content and the creation time of files are implemented. Garfinkel argues that this could foster forensic tools developers adopting the format.

In addition to metadata regarding the evidence itself (e.g. dates, size, offset and content type of the files), DFXML can represent information regarding the tools used for processing, metadata of the forensic image and file hashes. Doing so, DFXML enables a fine granularity into searching and auditing evidence. Although a vital step for information management, the actual correlation of distinct evidence is not tackled. Nevertheless, using DFXML and its API for analysis purposes like describing malware families is suggested as future research.

Also focusing on information representation and exchange, the Advanced Forensic Format 4 [27] (AFF4) was suggested in 2009. It is a backward-compatible evolution of AFF1 [28], an open and extensible disk image format. The authors argue that AFF1 was distinct from others formats in the sense that it compresses the processed images and allows storage of arbitrary information such as disk metadata within them. As it was designed to simply image a collection of hard drives, its

data model did not consider, for instance, chain of custody information about who had access to the disks, nor representing related evidence found in distinct drives.

In addition to technical improvements (e.g. enhanced encryption and compression techniques), AFF4 can deal with multiple related devices. For example, it allows that distinct practitioners, in different locations, analyse the same group of devices. In order to do that, a global management of evidence is necessary. For implementation purposes, it seems reasonable that each piece of information stored in the hard drives should have a unique identifier. But what is information, in the intended context?

Therefore, the designers of AFF4 data model devised abstractions based on computing concepts, such as volume, segments and streams. These are considered AFF4 objects, with specific attributes and which can be related to other objects. All metadata regarding the relationships and properties is defined using the tuple notation (Subject, Attribute, Value), which is also an advance from AFF1. The latter stored arbitrary metadata using name/value pair notation. Because of their scope, AFF4 objects, properties and relationships do not comprise investigation concepts. Nevertheless, their work is also aligned with one of the main characteristics of semantic technologies, applied by the thesis: the tuple notation, which, in the case of AFF4, is used for evidence management.

The European Union also recognized the importance of global management of digital evidence. It is currently funding the European Informatics Data Exchange Framework for Courts and (Evidence project) [16], which aims at developing best practices and procedures for digital evidence gathering and sharing across its members. Being divided in eleven working plans, the second one is targeted at categorization, and has identified eight main classes representing the whole digital evidence investigation domain: crime, source of evidence, process, electronic evidence, requirement, stakeholder, rule and digital forensics.

The project relies on semantic technologies to describe concepts from each of the previous classes and their relationships. The electronic evidence, for instance, has the concept `Internet log metadata` (which is a subclass of `Automatic Generated` evidence), and the relationship `Electronic Evidence` *is contained in* `Digital source of electronic evidence`. Although modelling only high level concepts, which are targeted at classification and management tasks, the choice of using semantic technologies by a relevant project reinforces the big potential for further research in ontologies regarding digital evidence.

One digital evidence representation scheme that uses semantic technologies to actually aid investigation, in addition to classifying related concepts, is the Digital Investigation Dialog (DIALOG) [29]. It models digital investigation, which has a smaller scope than global evidence management, in four dimensions. Accordingly, they resemble some dimensions suggested by the Evidence Project: *Information* (Electronic Evidence), *Evidence location* (Source of Evidence), *Forensic resource* (Digital Forensics) and *Crime case* (Crime).

Differently from the AFF4, it comprises non-technical abstractions as well, such as `TheftCase`, a subclass of `CrimeCase`, and `ConventionalLocation`: any non-digital location in which evidence can be found. Notwithstanding the good intent in detailing a large set of abstractions, the authors might have over-specified some of them, or even chosen non-intuitive names. `Conventional Location` and `UtilitySoftwareObject`, the latter being a subclass of `PersonalApplicationSoftwareObject`, are some examples that might make investigation tasks more complex, consequently hampering practitioners to adopt it.

Regarding the technological dimension, the authors successfully demonstrated the information ontology use by enhancing the registry analysis tool *RPCompare*. RP stands for Restore Points, a technique that enables Windows systems to be reverted to previous states, as long as snapshots were taken. RPCompare works by matching the registry hives within each snapshot: if a specific registry key is present in one snapshot, and not present in the following one, the tool states it has been deleted, for instance.

Because such comparisons normally result in a large number of differences, the authors created abstractions and rules representing Windows Registry concepts to automatically deal with them. `RegistryKeyObject hasSubkey RegistryKeyObject` and `RegistryKeyObject isInHive RegistryKeyHive` are some examples. By grouping related subkeys and defining the meaning of relevant keys (e.g. `HKLM\Software\Microsoft\Windows \CurrentVersion\Run` indicates the application that start running after a user log in), RPCompare can automatically infer, for instance, which software have been installed between snapshots.

Despite its broad description scope and lack of automated tagging (i.e. the user has to manually input values for `Registry Path`, so the software can process them), this work highlights the authors' "...attempt to formalise the rules by mimicking the human investigator reasoning process" [29].

Another approach regarding the mapping between the automatic analysis performed by the computer and the reasoning process of the investigator was suggested in 2012 by Hargreaves [30]. He proposed a timeline reconstruction tool to automatically relate low level events such as Windows Registry updates. The resulting high-level events would indicate that, for instance, the suspect searched for the expression "PDF viewer" in a Chrome browser just after he connected a USB drive to his computer.

In a forensic setting, Hoelz and Ralha [31] argue that the absence of formal definitions regarding the output data from different forensic tools hinders automated evidence processing. After all, ad-hoc parsers would be required to interpret data which, although representing the same thing, come from different sources. Adopting one common ontology for all digital artifacts in a forensic process could facilitate the stages of the investigation following the data extraction by removing potential ambiguities, simplifying data integration and enabling computer-aided reasoning. Section 2.3 shows

that this challenge (further discussed in Chapter 3) was considered relevant by Europol, and is thus the base for this thesis.

For demonstration purposes, Hoelz and Ralha presented a case study in which EXIF information was automatically analysed and annotated. Coming from 2 distinct EXIF extractors, the data differed in terms of variety, attribute names and value types. After integrating the data using an EXIF ontology, the picture instances were automatically enriched with annotations. For instance, the maker and the model metadata was used to distinguish their origin between "smartphone" and "digital camera". Moreover, specific face recognition metadata was used to indicate which pictures contained faces. In addition to proposing the semantic integration and enrichment of digital information, the authors demonstrated the feasibility of such technologies in a real forensic setting. As future work, they suggest engineering ontologies for other domains, and research visualisation strategies for easily spot relevant annotations.

Although exploring the same theme (semantic technologies applied to the investigation of digital artifacts), [29], [30] and [31] differ from the problem being tackled in this thesis in the sense that they suggest ontology-based solutions for the *automatic* interpretation (or semantic annotation) of forensic evidence. Although relevant, this thesis proposes that semantic technologies could be more useful for the investigation as a whole if the same capability used to define such ontologies were given to the investigators, allowing them to explore and augment intelligence data with their own perspectives and knowledge. For instance, none of methods suggested by those three works allows the users to define bespoke classes or relationships. And such flexibility in analysing data is necessary because the "deeply cognitive" tasks concerning the thorough investigation of a criminal case still require a human for "connecting the dots", and thus cannot be fully automated.

One of the most complete works which implemented a prototype for representing, integrating and linking digital evidence is the Master thesis of Dossis [32]. As the main benefits, he mentions streamlining the analysis and reasoning within a digital investigation, and also lowering the barrier for new professionals joining this field, which are two of the objectives of this thesis. The author suggests that the semantic integration of information obtained from different sources (e.g. storage devices, network captures and memory dumps) could enable the computer to perform automated reasoning. Although there have been previous approaches to relating different sources, such as the FACE framework [22], doing so using ontologies is very recent.

Five distinct ontologies were proposed, modelling forensic-relevant concepts regarding network capture, storage devices, Windows firewall logs, Whois lookup service, malicious networks and malware detection. Their integration is made possible through mappings like PcapIPToFWLogHost, which links the IP address from the network capture ontology with a host from the Windows firewall one.

In addition to integrating similar concepts from different ontologies, the proof of concept allows temporal and mereological (e.g. the connections between part of an entity and the entity itself) correlation among different types of evidence. That way, it is possible to perform an integrated query looking for "...which files have been accessed in a short time period after the download of a malicious files from the web".

For evaluation purposes, Dossis adopts the "Goal-Question-Metric" approach [33]. According to it, the alignment of the proposed system is measured against a previously defined measurement model, consisting in tuples (Goal, Question, Metric). One proposed criteria is:

- Goal: The method should be flexible and scalable.
- Question: Can the method deal with new sources of data or being able to seamlessly integrate new forms of ontologically-expressed knowledge and rules.
- Metric: The ability of the method to process new data and accept additional ontologies or rules without the need of major (possibly even none) modifications on the existing steps. It can be measured by the amount of configuration or code modifications such changes may require.

Dossis justifies this qualitative metric due to the "...lack of a common and established approach on evaluating digital forensics methods and tools along with the interdisciplinary nature of the proposed method." This thesis is directly aligned with one of the suggestions from Dossis for future work: studying if and how the approach he demonstrated could benefit real investigations. Dossis also suggests researching visualisation techniques that could enhance the interaction between human and computer in the process. Although not the focus of this thesis, it is indeed considered relevant, and will be tackled as future work.

One of the latest evidence representation schemes suggested, which also starts making use of semantic technologies, is the Digital Forensic Analysis Expression (DFAX) [34]. It was proposed in 2015, and is based on the Cyber Observable Expression (CybOX), "..a standardized schema for the specification, capture, characterization, and communication of events or stateful properties that are observable in the operational domain" [35].

Some target operational domains in which Cybox can be used are malware characterization, intrusion detection and incident response. As a suggestion for the evidence management domain, DFAX merges a subset of Cybox technology abstractions (e.g. `modified time` and `registry entry`) with an ontology comprising forensic concepts such chain of custody and case management (e.g. `Case` ***`has`*** `Examiner` and `Forensic Action` ***`analyses`*** `Evidence Record`).

In comparison with other schemas like XIRAF and DFXML, the authors argue that one key feature from DFAX is its capability of representing actions like a usb drive connecting to Windows. This abstraction, derived from digital traces such as registry key values, could enhance forensic analysis by providing a higher-level view of the activities that happened in the system being analysed.

Although DFAX was implemented using a distinct language, the authors consider migrating to OWL in the future. However, they intend to do so only after "...DFAX matures and an explicit ontology has been agreed upon within the digital forensic community."

This is one of their criticism to the AFF4 format. Although recognizing that its tuple notation enhances the flexibility of the model, they state that there is still not a clearly defined and previously agreed ontology for the model. Therefore, they suggest DFAX as a basis for community consensus.

This supposed need for previous general consensus might hold back applied ontology research within the digital forensics field and defining a large set of concepts at once might hamper objectivity, as such a complex prototype would be difficult to implement and validate. Moreover, mapping current schemes of the users to a very comprehensive taxonomy might not be cost-effective, as the practitioners might not envisage the benefits of doing so.

Conversely, a representation scheme which is really expected to be adopted should evolve incrementally and in close contact with the users from the target domain, focusing on usability and proving to be effective to their analysis needs. In order to achieve those, a working prototype exploring has to be implemented and validated first, and this is how this thesis aims to demonstrate that semantic technologies could indeed help investigation and analysis tasks.

Therefore, the next subsection will discuss more practical approaches to forensic analysis and cybercrime investigation which rely on or are compatible with semantic technologies. Ontologies for malware relationship analysis are of particular interest. After all, despite the wealth of possibly related open source information available (e.g. online sandboxes, malicious hosts databases, vulnerability disclosing platforms and forums), it is still difficult to analyse malware-related campaigns. Both the high-availability of data about it and the intrinsic analysis challenges (e.g. multiple data sources, constant evolution of TTP from the criminal organisations, the advance of new technologies in contrast to the existence of legacy equipment in the same domain) were decisive to scope this thesis to the malware investigation topic.

2.3 Malware investigation using semantic technologies

Any investigation using semantic technologies must start with the definition of an ontology describing the domain being explored. One interesting example is an ontology dedicated to malware analysis, which is presented in a paper [36] authored by researchers from the Computer Emergency Response Team (CERT) at Carnegie Mellon University. Its concepts are based on a previous effort from the same authors to define a controlled vocabulary for malware analysis, the Malware Lexicon [37]. In order to create it, they applied text mining techniques to a dataset of emails from the CERT malware analysis team spanning 10 years. They also scanned the indexes of relevant books and considered some open source and grey literature, which they recognize could be further explored.

Their goal is to provide a more scientific approach to malware research, and they hope other experts will adopt their ontology, thus starting to "speak the same language". This could help addressing the "lack of comparability" issue between researches regarding digital evidence and professional malware analysis reports [4]. Although related to this thesis, it appears that the main goal of the CERT ontology is to standardize the description of the inner details of malware, enhancing information sharing between malware-analysis teams. On the other hand, this thesis focuses more on the relationship between distinct samples of malware and other IOCs rather than the analysis of the malware itself. For instance, it will consider external information such as IP addresses of malicious servers. Even though not directly related to a particular malware family, such information might help revealing relationships between different malware.

Also considering malicious remote hosts, Jiang and Li [38] published an article discussing automatic identification of botnet topologies. For that, they devised a small ontology comprising concepts like `BotMaster`, `C&C server` and `IPAddress`, and rules such as `BotMaster` *`controls`* `C&C server`. The instances were created using data from the site *dshield.org*, which the authors did not give further details about. One concern is that their validation does not measure the effectiveness in identifying botnet topologies. Rather, it solely compares the inference times (i.e. the time taken to infer logical consequences from the provided rules) using three popular reasoners. While they claim that their method is more accurate and has lower inference time than other anomaly and misuse detection systems, the connection between the experiments and the conclusions is not very clear. Nevertheless, the research idea seems valid and worth further exploring.

A different perspective about applying ontologies into the malware analysis domain is proposed by Huang et.al. [39]. The authors suggest bridging semantic technologies with fuzzy methods in order to enhance decision-making towards malware identification. Their argument is that ontologies are not able to represent imprecision and vagueness knowledge, inherent to malware identification. So, they defined three fuzzy variables: File Hash (FH),obtained by comparing the fuzzy hash [40] of a malware with a list of hash codes from previously identified malware; ConnectIP (CI), the number of TCP/IP connections created by an application; and FileChange (FC), the number of changes in registry between clean state and malware execution.

These three variables are used to describe each malware instance, and can have one of the following fuzzy terms assigned: `very less`, `less`, `normal`, `much` and `very much`. Each value is defined according to output of the bespoke tool Taiwan Malware Analysis Net, used to process the malware instances. Then, SWRL rules representing the variables and terms would define the similarity between two samples of malware. For instance, `FH` *`normal`* AND `FC` *`very much`* AND `CI` *`less`* THEN `SI` *`medium`*.

In this case, semantic technologies are used in support of fuzzy techniques, allowing domain experts to edit SWRL rules [41] as necessary. Although comprising only very simple concepts and

relationships - perhaps the same functionality could be obtained without an ontology, the approach of the paper to measuring similarity could be useful in assessing two pieces of malware before merging them using the object property `owl:sameas`, as described in Section 4.2.2.

This thesis will not suggest an ontology for malware investigation, as there are already comprehensives taxonomies whose concepts could be borrowed to define one, such as MALWG [42], MAEC/STIX [43, 44] and OpenIoc [45]. The last two also describe relationships between entities, which approximate them to ontologies. Their development is a consequence of the modern understanding that cyber-threat intelligence cannot be effectively represented as an uncontextualized flat list of IOCs, as it describes complex and richly connected entities [46].

It is debatable if current threat-intelligence platforms, whether leveraging such taxonomies or processing raw IOCs, are actually producing intelligence. Most of them still focus on data collection rather than analysis, which is often limited to searching, browsing and attribute-filtering [9] as opposed to knowledge and intelligence creation. MISP [47], a well-known platform, is one example: it favours sharing IOCs among different organisations rather than analysis. For the latter, it comprises a graph (which is often cluttered) and tags which can be assigned by any analyst.

Nonetheless, it is a relevant community-driven tool which recently started to support the concept of *objects* (i.e. in addition to attributing raw IOCs - such as *md5 hash* and *domain name* - to an event, it now allows to insert them into objects - in the case, *file* and a *domain* objects, respectively - which might hold relationships among themselves. This new perspective of *objects*, coming from a community-driven tool, only confirms the tendency of expressing malware knowledge using ontologies rather than descriptive taxonomies with no support for relationships.

Finally, it is worth mentioning that Europol also recognized the relevance of semantic technologies in acquiring and processing digital evidence. In May 2017, Europol's European Cybercrime Centre (EC3) has announced that the market leader companies in digital forensics have agreed into supporting a open-source data format for representing information [48] A standard format such as the Cyber-investigation Analysis Standard Expression (CASE) was long awaited by forensic analysts, who had to deal routinely with different data formats adopted by distinct tools: "CASE is a community-developed standard format, defined as a profile of the Unified Cyber Ontology (UCO). As such, CASE leverages contextually relevant components of the UCA; extending, constraining or renaming them as appropriate. CASE is specified at a semantic level and supports various serialisations, its default serialisation being JSON-LD".

The increasing adoption of JSON-LD [49] could motivate OSINT data providers to publish their JSON data in JSON-LD. After all, the compatibility between both formats means that only small modifications would be necessary to convert the former into the latter. This fact, in addition to the active research regarding the scalability of current reasoners, could foster the development of novel semantic approaches to the exploration of the available data sources on the web, and eliminate the

data conversion step prior to using any prototype leveraging sema fntic technologies, as described in Section 5.2.1.

As detailed in the previous paragraphs, there are already good initiatives regarding malware knowledge representation and sharing. Thus, this thesis aims to be a step towards completing the intelligence cycle by presenting a prototype which leverages semantic technologies in order to facilitate analysis tasks such as:

- Data integration across disparate data sources: because it might be infeasible for all stakeholders to adopt a particular schema, a straightforward way to define mappings between them is recommended;
- Facet querying: in addition to interactive filtering, it helps understanding the relevant dimensions of a particular domain [14];
- Establishing provenance, or efficiently tracing back any piece of information to the original data source.

Szekely et al describe how such tasks have improved the investigation of human trafficking in a law-enforcement setting [13]. This thesis will go further than that by considering other native capabilities which could enhance relationship-searching and hypothesis-testing within the malware-campaign investigation domain, as detailed in Table 1.1. From the best of the author's knowledge and the literature review conducted so far, there is no other approach providing such capabilities to end-users whose primary duty is to analyse data.

2.4 Requirements for analysing intelligence information

The investigation needs listed in Table 1.1 were selected for their relevance to any criminal investigation task which involves searching for and assessing relationships between items in a dataset. In other words, network charting and analysis, which are techniques listed in [12] as some of the most useful for the Intelligence Community. By bringing more structure to and enabling collaboration in the investigative process, the method suggested in this thesis is directly aligned with the field of structured analysis which, according to the authors of [12], aims at externalising thought processes "...in a systematic and transparent manner so that they can be shared, built on and easily critiqued by others."

Earlier in the book, the same authors reveal that they do not know about any software that does the intermediate task of grouping nodes into meaningful clusters, "...though algorithms do exist and are used by individual analysts". Furthermore, they do emphasize on the interpretation of what is represented in the graph, a task which relies on the knowledge of the investigator, and therefore

cannot be fully automated by any computer means. On the other hand, semantic technologies can actively assist the investigator in interpreting such data.

Among the necessary steps to conduct network analysis mentioned in the book, the following excerpts have a clear correspondence with the semantic features listed in Table 1.1 (which are necessary to build, enrich and share knowledge bases). Therefore, these features were chosen to be implemented in the prototype, thoroughly discussed in Chapter 4.

1. *Clustering entities:* "Cluster the nodes. Do this by looking for "dense" areas of the chart and relatively "empty" areas" *and* "Cluster the clusters, if you can, using the same method".

2. *Establishing links:* "Identify, combine, or separate nodes within this reporting" *and* "Draw the connections between nodes - connect the dots - on a chart by hand, using a computer drawing tool, or using Network Analysis software".

3. *Inserting tags or comments:* "Label each cluster according to the common denominator among the nodes it contains. In doing this you will identify groups, events, activities, and/or key locations".

4. *Enriching data about an entity:* "Update the chart and supporting documents regularly as new information becomes available".

5. *Rolling back in case of dead end:* "Stop in these cases: when you run out of information, when all of the new links are dead ends, when all of the new links begin to turn in on each other like a spider web, or when you run out of time".

6. *Reproducing investigation steps:* "Each structured analytic technique involves a step-by-step process that externalizes the analyst's thinking in a manner that makes it readily apparent to others, thereby enabling it to be reviewed, discussed, and critiqued piece by piece, or step by step".

7. *Sharing investigative knowledge:* "Many things change when the internal thought process of analysts can be externalized in a transparent manner so that ideas can be shared, built on, and easily critiqued by others".

8. *Exploring the dataset:* "Look for "cliques" - a group of nodes in which every node is connected to every other node, though not to many nodes outside the group" *and* "Look in the empty spaces for nodes or links that connect two clusters. " *and* "Analyze this flow. Does it always go in one direction or in multiple directions? Are the same or different nodes involved? How many different flows are there? What are the pathways?".

9. *Dataset integration:* "Add nodes and links from other sources, constantly checking them against the information you already have".

10. *Establishing provenance:* "Identify at least one reliable source or stream of data to serve as a beginning point".

In addition to the findings discussed in this chapter and observed in the tools being used by the community (e.g. MISP relies heavily on bespoke tags, IBM I2 platform makes extensive use of graphs, etc), the investigation needs listed in Table 1.1 were also validated by two informal interviews with malware experts from the big security companies who agreed that they were relevant to any malware-campaign investigation.

The first interviewee, in addition to giving examples of relationship-rich IOCs (such as strings, debug paths and SSL certificates), mentioned that one of the main issues in current malware-investigation procedures is the lack of tools being developed to industry analysts. The focus is still developing tools to the customers of anti-virus and threat intelligence companies, and not to the internal work of their employees. That results in parallel effort, as analysts develop their own tools and scripts, which will not necessarily be compatible with each other. He also mentioned that analysts make extensive use of tags for describing related artifacts and also Yara, a rule-based tool aimed at searching similarity between different malware.

Finally, the second interviewee recognized the value of rule-based tools in helping industry analysts conducting malware investigations. In his own words:

In terms of investigation hypotheses , there are a tonne to be honest. In fact its kind of an entire field of work – but some examples (each with their own degree of levels of confidence):

- *Two malware connecting to the same domain may belong to the same family or attacker;*
- *Two domains sharing attributes in SSL certs could be related;*
- *Two IP with the same SSH public key can be related to the same controller;*
- *Domains with the same email or personal details in their historical whois data can be linked;*
- *Two malware with similar SSDeep (or better, TLSH) hashes for the file or its sections can be related*
- *Two malware can be sometimes linked by Mutex*

And a whole bunch more – trust me, its literally a discipline unto itself

The opinions shared by both interviewees and the authors of [12] are directly aligned with the scope of this thesis, and are also compatible with the one of the main findings from the literature review: the need for providing a better structure not only for the massive amount of information concerning the investigation of cybercrime, but also for the tools and methods employed by the investigators during analysis. The next chapter will detail how the research hypothesis of this thesis was conceived and matured based on these findings.

Chapter 3

Developing the research hypothesis

Being a member from the Computer Forensics Labs at the Brazilian Federal Police since 2006, the author's constant reflection on how to make cybercrime investigation and forensic analysis processes more efficient was a key incentive for undertaking this project.

The objective of this chapter is to clarify the reader about the thinking process which, starting from that personal motivation and being later corroborated by the literature review (Chapter 2), resulted in the research hypothesis that the investigation needs listed in Table 1.1 (and justified in Section 2.4), once addressed by the correspondent semantic capability in the same Table, could facilitate investigative tasks such as data exploration and insight generation.

Section 3.1 consists on the initial "brainstorming" exercise which led to the evidence model suggested in Section 3.2. Such evidence model, although not being assessed in this thesis, was fundamental to inspire both the development of the prototype (Chapter 4) and the assessment of the investigation method leveraging it (Chapters 5, 6 and 7). Finally, Section 3.3 aims to briefly instruct the reader about some key capabilities on semantic technologies which could provide better support for understanding how the prototype features could be translated into them, and also their relevance to any data exploration task.

3.1 Rethinking digital evidence

The literature review identified that digital forensic frameworks and evidence representation schemes are still largely dedicated to evidence recovery and categorisation. Moreover, the few attempts to connect all this information are mostly based on their technical metadata, not their meaning.

In the context of LEAs, finding relationships regarding the meaning of distinct digital evidence is still entirely performed by the human. Part of it is done by the forensic analysts, who are provided with keywords related to the case being investigated so they can extract, following the best practices for the chain of custody, the largest number of information potentially useful to the investigation.The other part is responsibility of the investigators, who assess the forensic reports from different devices in order to form they conclusion regarding the cybercrime being investigated.

In a traditional investigation setting, the investigator is assigned a criminal notice and is responsible for the first diligences such as collecting more information about the suspects and searching official databases for correlations with distinct cases. Once enough information about the potential culpability is collected, a criminal warrant is issued, allowing the police officers to search and seize any items belonging to the suspect.

At this point, any digital devices seized are forwarded to the forensics lab, together with a request containing questions about the information on the device or specific keywords to be searched for. Criminal investigations following this way of work have two well-defined roles with specific knowledge each, according to Table 3.1.

Information	Investigator	Forensic analyst
Case-specific	X	
General cybercrime	X	X
Technology-specific		X

Table 3.1: Cybercrime investigation roles and knowledge.

However, due to the increasing demand of forensic reports regarding devices with ever larger storage capacity, some recent approaches leave all the work of analysing information to the investigators, being the forensics lab responsible for only providing access to the information respecting chain of custody rules (see Section 2.1 for one example).

While the former approach suffers from scalability issues (after all, investigators outnumber forensic analysts, which cause a bottleneck in the investigation process), the latter suffers from the lack of digital evidence knowledge of the investigators, what could make the results of the investigation less accurate and longer to achieve. Moreover, the huge amount of data to analyse also affects the latter approach as well, and the solution to that is not (only) hiring more analysts, but making the computer able to better assist the human while conducting criminal investigations.

The scalability issue definitely indicates leveraging computer processing to solve it. But which computer technology to use? The issue regarding the knowledge from two related but distinct domains suggests that semantic technologies could be a good start. Prior to assessing it, however, it is necessary to understand how both domains are related. So, as a starting point, the following questions, which refer potentially to any cybercrime investigation, were posed:

1 What is necessary to know?

2 Where the information could be found, and how could it be extracted?

The first question comes from the investigation team, and involves both case-specific and general cybercrime information. Its objective is to scope down what evidence is necessary to prove authorship and materiality of a specific cybercrime case. Some examples would be:

A Is there information about user X in the device? Is it possible to confirm his name? Has it downloaded pictures from a P2P network?

B Is there any malware installed in device Y? If so, is it active, and has it sent information to a remote location?

C Are there any offensive comments towards person X? If so, is it possible to confirm they have been posted from device Y?

D Are there any receipts with beneficiary "Mr. Smith"? If so, is it possible to identify who made the payment?

The second question relies on the empirical experience of the forensic analyst. It involves general cybercrime and technology-specific information. In order to reply to the questions from the investigation team, the forensics analyst should know:

A Where to look for downloaded files records among users from a specific P2P application;

B How to extract embedded files from a piece of malware;

C That comments posted in social media, if not available on the browser's history, might still be recovered using data carving techniques;

D That just searching for "Smith" will retrieve a lot of unrelated hits from the internet cache, although some of them might be relevant.

3.2 Building a semantic-enabled evidence model

Thinking about the questions from the previous section helped to update the objective of traditional criminal investigations with the cybercrime perspective: while the former tries to reveal authorship and materiality (as both are necessary to prove that a crime has happened), a cybercrime investigation tries to demonstrate "how authorship (*people*) is connected to materiality (*subject*) through digital means (*technology*). This insight served as inspiration to define an evidence model which is composed by three main dimensions: people, subject and technology. For instance, an email message (*technology dimension*) with references to two suspects (*people dimension*) might constitute evidence, depending on its content(*subject dimension*). Its meaning and relevance is attributed by the human analyst, who compares the context of the message with the knowledge obtained from the investigation in course.

In addition to the relationship among these dimensions, there are two metrics relevant for any crime and cybercrime investigation: time and (geo)location (in the case of the cybercrime, location could also refer to the device or the cloud location in which evidence was found, in addition to

the physical location (e.g. address, city and country). All these dimensions and metrics could be mapped to the "Five Ws and How"[1] questions.

- People dimension: WHO
- Subject dimension: WHAT
- Technology dimension: HOW
- Timestamps: WHEN
- Different devices/sites: WHERE
- Reason for crime: WHY

Reflecting on such dimensions and the relationships between them should lead to a better specification of which evidence is necessary to prove a cybercrime occurrence. Ultimately, clearer information requirements would help better inform LEAs and internet service providers regarding exceptional access[50].

The *people* dimension comprises any information that can be used to identify a person or an organization. It comprises only real-word information, or information that would exist even if there were no computers. Some examples are names, addresses and relevant dates. Such information is commonly stored in a wealth of digital devices, owned or not by the entity, and with or without their consent.

The main objective of the *people* dimension is to aid the analyst in enriching the profiles of suspects and unveiling their networks. For that, the SuperIdentity [51] model, developed by University of Oxford's Cyber Security Group, is particularly useful: in order to enrich profiles and attribute actions to individuals, it considers the concept identity as a collection of elements from four different domains: biological, psychological, biographical and cyber, spanning both the natural and the cyber world. For instance, a person has one real name (biographical), but might also have a couple of different user-names (cyber). Therefore, the model intends to link such information, attributing a confidence score for the connection.

The *technology* dimension comprises purely technical information, whether abstractions of digital concepts or metadata. The main objective of the *technology* dimension is to define which atomic pieces of digital information might contain or support cybercrime evidence. Some examples are information regarding *connection to P2P network, visited sites* and *contact lists.*

[1]The Five Ws (sometimes referred to as Five Ws and How, 5W1H, or Six Ws)[1] are questions whose answers are considered basic in information gathering or problem solving. They are often mentioned in journalism (cf. news style), research, and police investigations. *Source:* `https://en.wikipedia.org/wiki/Five_Ws`, *accessed in* 07/05/2017.

The concepts of this dimension were inspired by the *Nepomuk*[52] project. It defined ontologies describing user-related data stored in the so-called Semantic Desktop, like files, contacts and messages. Although the project have been long developed by the KDE[53] community, it was recently abandoned due to scalability issues within commodity computers, which is the target platform of the KDE interface. Nevertheless, its scope alignment (after all, most crime evidences are contained in these data structures (files, contacts, messages) and the level of details describing the relationship between classes make the *Nepomuk* group of ontologies an excellent foundation for the technology dimension.

Finally, the *subject* dimension comprises information particular to a domain - in our case, the crime being investigated, such as techniques used for money laundering and keywords commonly used by hackers. Similarly to the *people* dimension, the *subject* dimension comprises only real-word information, or information that would exist even if there were no computers. Figure 3.1 illustrates these dimensions, their intersections and the type of evidence contained within each of them:

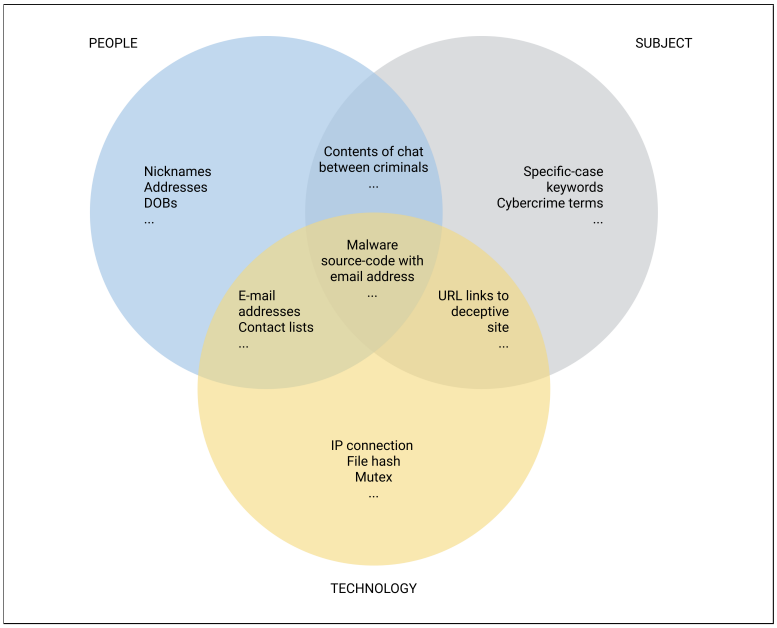

Figure 3.1: Model dimensions.

Knowledge from each of these dimensions could be represented as axioms:

- People: Person hasNickname "Bob" (real-word information, not necessarily related to a crime);
- Technology: EmailAccount hasSentDate Date (purely technical, not necessarily related to a crime);
- Subject: BankingFraudCrime describedByKeyword "carder2018" (information describing a crime but with no reference to a specific criminal);

- People × Technology: Person hasEmailAddress EmailAddress (not necessarily related to a crime offence, but could be used by it);
- People × Subject: “Bob" sameMessageAs “carder2018” (there were references to both “Bob” and “carder2018” found in the same text, which is not necessarily digital - i.e. could be a piece of paper found in the house of a suspect);
- Technology × Subject: “SourceCode” isA PhishingTool (e.g. one script which consumes a payload and a list of emails to generate and send phishing messages);
- People × Technology × Subject: “SourceCode” containsKeyword “Bob” (the *technology* × *subject* evidence from last item contains a reference to “Bob”).

Despite having some open issues (such as where does the *device* dimension would fit in the model) thinking about which evidence belongs to each dimension and being able to make the meaning of the evidence accessible to computers confirm that researching about *how semantic technologies could make cybercrime investigation more efficient* is a topic that, in addition to justifying a phd thesis, could indeed be useful for real-world investigation problems. Therefore, the next section will introduce the reader to some basic concepts about semantic technologies.

3.3 Quick review of semantic technologies

It is probably easier to explain what semantic technologies are by presenting the idea behind the Semantic Web as defined by Tim Berners-Lee: “The Semantic Web is not a separate Web but an extension of the current one in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [54].

Such well-defined meaning is given to information through an ontology, which aims to explicitly define meanings shared by a community [55]. The basic components of an ontology are:

- **Classes**: describe concepts, or the type of things within a domain. For instance, in the case study presented in Chapter 5, `WebServer` and `File` are two classes, and `PayloadFile` is a subclass of the latter, inheriting all its properties;
- **Properties**: relationships between members of classes (*object properties*) or between members of a class and a literal (*data properties*). Some examples from the previously mentioned case study are `connectsTo` (linking members from class *PayloadFile* to members of class *WebServer*) and `name` (describing a characteristic of type *string* about members of the class *File*), respectively;

- **Individuals**: the instances of a class (or the objects described by it). For instance, *evil.org* could be one instance of the class `WebServer`. The individuals are not considered part of the ontology, but together with it form what is called a *knowledge base*.

Classes, properties, individuals and all the other components of a knowledge base are logically stored as triples (*subject predicate object*), describing one statement each. Triples are represented using the RDF, a framework which employs web-based URIs to name the relationship between things as well the things themselves, allowing structured data to be integrated and shared across different applications. Listing 3.1 presents the components of the ontology described above as triples in *Turtle*[56] syntax.

Listing 3.1: Ontology definition in Turtle syntax.

```
# Prefixes for standard namespaces and bespoke ontology
@prefix owl: <http://www.w3.org/2002/07/owl/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema/> .
@prefix onto: <http://ontology.com/> .

# Classes and subclasses
onto:WebServer rdf:type owl:Class .
onto:File rdf:type owl:Class .
onto:PayloadFile rdf:type owl:Class ;
   rdfs:subClassOf onto:File .

# Object property
onto:connectsTo rdf:type owl:ObjectProperty ;
   rdfs:domain onto:PayloadFile;
   rdfs:range onto:WebServer .

# Datatype property
onto:name rdf:type owl:DatatypeProperty ;
   rdfs:domain onto:File ;
   rdfs:range xsd:string .

# Individuals
<http://ssreport.com/1a2b3c4d5e6f> rdf:type onto:PayloadFile .
<http://ssreport.com/1a2b3c4d5e6f> onto:connectsTo
<http://ssreport.com/evil.org> .
<http://ssreport.com/1a2b3c4d5e6f> onto:name "malware.exe"
```

Once in a linked format, triples must be loaded into a triple store (or rdf store), in order to enable semantic querying and materialising new facts over the data. Triple stores are a kind of graph database designed for the storage and retrieval of tuples and which supports data inferencing using rules and set processing[2] using SPARQL. Graph databases are more suited to store and query highly-connected data than relational database systems, facilitating analysis tasks in which relationships and connection patterns between different items is important, see [57] for discussion.

Although native graph databases outperform triple stores in data-processing performance, the latter are better suited for information-modelling and sharing. Moreover, their inherent capability of

[2]Set processing is a SQL technique used to process groups, or sets of rows, at one time, rather than processing each row individually. *Source:*: `https://docs.oracle.com/cd/E80738_01/pt854pbh2/eng/pt/tape/task_UsingSetProcessing-07720a.html`

integrating datasets is relevant, as current threat-intelligence platforms provide limited automated data-integration support[9].

Leveraging a graph technology which favours data integration contributes to investigation pivoting, described by [46] as the fundamental analytic task of *hypothesis-testing* which relies on the ability of analysts to *understand the relationship between elements from distinct data sources.*

Finally, listing 3.1 illustrates one of the key characteristics of the Semantic Web: that classes and properties are defined by Uniform Resource Identifiers (URIs). This ensures that they have an unique definition accessible by anyone on the Web, avoiding concept misunderstanding across different contexts [55].

For instance, one web-master could mark-up the universities described in her site using the definition present in `https://schema.org/Organization`. This would enable any software agent aware of *schema.org* (e.g. a search engine) to recognize "University of Oxford" as an organization, which would then be a "thing" instead of a string. Consequently, this "thing" could now hold the relationship `https://schema.org/employee` with multiple individuals marked-up as `https://schema.org/Person`.

Of course, "The computer doesn't truly "understand" any of this information, but it can now manipulate the terms much more effectively in ways that are useful and meaningful to the human user." [54]. A very good example is linking disparate resources describing the same "thing", which could be very useful for data enrichment while keeping its provenance. This will be further discussed in Section 4.2.2.

Chapter 4

Implementation of the prototype

The interaction between semantic technologies and cybercrime investigation is still very recent, as mentioned in Section 2.3. In order to evaluate whether the latter could benefit from the former, three research questions were defined in the end of Section 1.4, referring to the feasibility of implementing the semantic features from Table 1.1 in a prototype, the learning curve of the users when applying such features, and how useful they could be to the investigation of malware-related cybercrime.

In addition to guiding the development of this thesis, those research questions also informed about the necessary enhancements of the prototype to validate each of them. The following sections of this Chapter will discuss:

- What technologies were chosen to develop the prototype and why;
- How each of the features listed in Table 1.1 were implemented, describing the potential challenges and explaining some design decisions;
- The evolution of the prototype during the course of this thesis, specifying the main updates.

4.1 Technology

There are three main components of the prototype itself: the reasoner[1], which performs all search and materialisation tasks; the GUI to provide the user with the commands, graph and table; and finally the libraries that manipulate RDF data, which process both the user input and the dataset information to be consumed by the reasoner. All these components were connected using Python language.

[1] A semantic reasoner, reasoning engine, rules engine, or simply a reasoner, is a piece of software able to infer logical consequences from a set of asserted facts or axioms. The notion of a semantic reasoner generalizes that of an inference engine, by providing a richer set of mechanisms to work with. The inference rules are commonly specified by means of an ontology language, and often a description logic language. *Source:* `https://en.wikipedia.org/wiki/Semantic_reasoner`, *accessed in* 07/05/2017

4.1.1 Programming language

Python is a scripting language that allows for quick prototyping automated tasks and counts with a wealth of specialized libraries developed by the community. In addition to that, it is the programming language which the author of this thesis has more proficiency with, and was thus chosen to the initial assessment of the idea of the prototype.

During this assessment, two factors contributed decisively to adopting Python as the main language for developing the full prototype:

- Finding out about a reasoner being developed by University of Oxford with some features relevant to the prototype. Such reasoner, despite implemented in C++, counted with a Python "bridge", allowing it to be executed natively from Python code. The reasoner is further discussed in Section 4.1.2;
- The discovery of a data-driven visualisation library implemented in Python capable of meeting all the requirements for future development of the GUI of the prototype. The visualisation library is further discussed in Section 4.1.3.

Other relevant Python libraries leveraged by the prototype and some tasks executed by them are:

- *RDFlib* for processing (e.g. extracting classes, object properties and datatype properties from the ontology file to fill the drop-down boxes);
- *Networkx* for graph processing (e.g. converting the classes and object properties returned by each query into nodes and edges and generating the layout);
- *Pandas* for data transformations (e.g. creating and updating data-frames for each class and respective datatype properties, in order to fill the table).

These and other functionalities were implemented using around 6860 lines of original code written in either Python or Javascript (necessary for specific GUI interactions). All those lines of code are divided in six files, according to Table 4.1.

4.1.2 Reasoner

The reasoner is the part of the prototype responsible for running queries and materialising new information, ultimately reflecting the knowledge of the investigator. Among the existing reasoners, the prototype makes use of *RDFox*, which is a main-memory triple store coupled with a parallel datalog reasoner. In addition to supporting SPARQL (a sql-like query language for RDF, based in graph pattern matching and endorsed by W3C) and handling the "owl:sameAs" object property efficiently, *RDFox* [58] is very fast at computing and updating data materialisations. All these

Table 4.1: Lines of originally-written code from each module of the prototype.

File	num of lines	Reason
dataprep.py	340	Formatting data to feed *RDFox*
seminv_rdf.py	970	Converting rdf data to Python structures
seminv_netwx.py	170	Creating nodes, edges and layout
os_ops.py	130	Auxiliary functions
configs.py	50	Configuration variables
main.py	300	Initialisation
render_bokeh.py	4900	GUI, controller, updates, interaction
TOTAL	6860	

features are important for the prototype, especially the latter: because the investigator might need to assess multiple hypothesis, fast computing of the materialisations involving defining new classes, relationships and tags is recommended. The importance of these features to our approach will become clearer in Section 5.2.

4.1.3 Visualisation library

The data-driven visualisation library leveraged by the prototype is *Bokeh*[59]. Although very recent (its first version dates back to April 2013), its ability to use information stored in Python data structures (whether the dataset itself or input commands captured from the user) to quickly generate or update a rich set of interactive visualisations and other GUI elements (e.g. graph, table, drop-down lists) proved well-suited for the prototype.

4.2 Features

This Section will discuss how the features listed in Table 1.1 are implemented in the prototype. As mentioned before, the overall goal of those features is to provide the investigator with great flexibility while exploring and analysing the data (by leveraging features such as facet queries, the graphical query builder and an interactive graph) and, at the same time, avoiding data-modelling constraints (as the domain logic is embedded within the ontology, which can be changed during analysis according to the own knowledge of the investigator). Finally, the flexibility is also expressed on the capability of sharing queries and rules with other investigators and merging data from distinct datasets.

The current interface is depicted in Figure 4.1, and will be used to explain each of the implemented features:

4.2.1 Feature 8 - Query builder

The query builder is one of the two main features of the prototype, and is located to the right of the graph. As the name suggests, it helps the user in creating SPARQL queries to be submitted to the

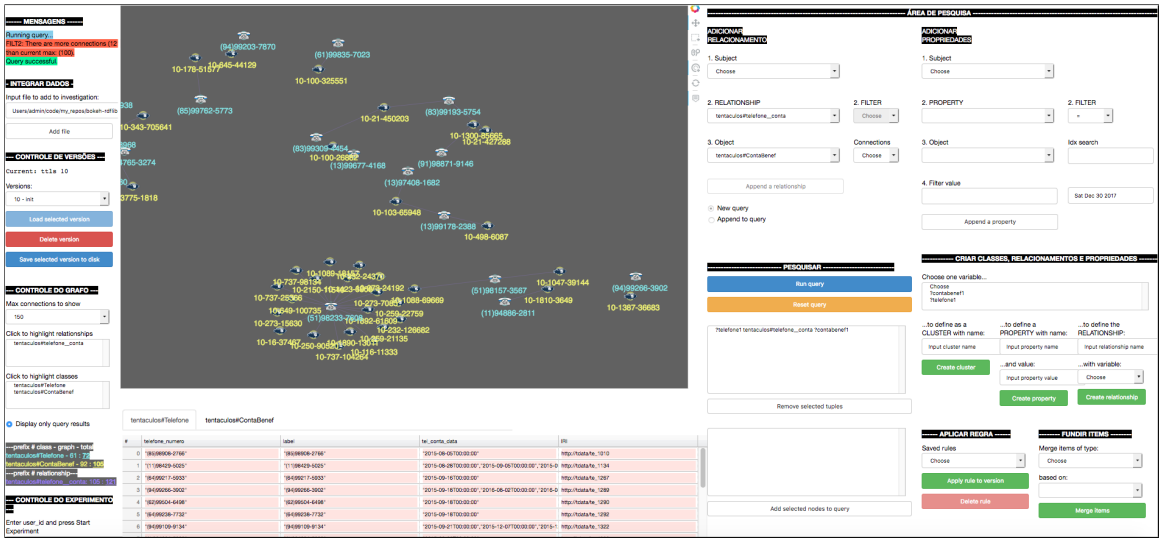

Figure 4.1: Prototype interface.

RDFox reasoner. It is composed of two main columns: one for appending triples describing object properties, and the other for appending datatype property ones.

Appending object properties

This column has three main drop-down boxes: the first one ("1. Subject") is initially loaded with all the classes from the ontology engineered for the investigation. Once a value is selected, the second drop-down ("2. Relationship") box gets loaded with the compatible object properties for the selected class. Finally, the third drop-down box ("3. Object") gets updated with the possible ranges for that object property.

Once the button "Append a relationship" is clicked, the values of the three drop-down boxes are appended to the query as a triple. Now, it is the first drop-down box that gets updated: in addition to the ontology classes, it now lists the two variables used in the first triple, enabling the user to continue appending object properties to the original graph pattern. Figure 4.2 illustrates this action: a second triple, using the object `?contabenef1` from the first triple as its subject, is about to be added to the query. On the other hand, if `tentaculos#ContaBenef` was chosen on the first drop-down box instead of `?contabenef1`, the resulting query would contain two graph patterns:

Listing 4.1: Two graph patterns on the same query.

```
?transferencia1 tentaculos#transferencia__contabenef ?contabenef1
?contabenef2 tentaculos#conta__pessoa ?pessoa1
```

It is worth noticing that despite the fact that only the triples are displayed to the user, the full SPARQL query (i.e. the triples within curly brackets preceded by a select statement) is saved in background, ready to be sent to the *RDFox* reasoner.

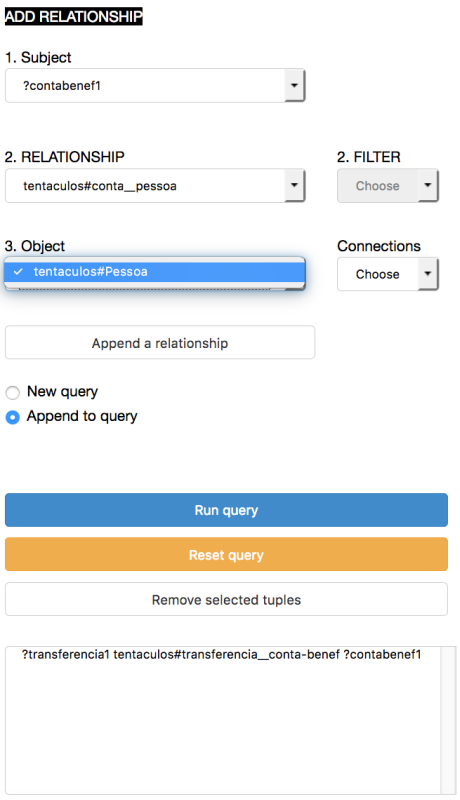

Figure 4.2: Appending object property triples to the query.

Finally, there are two auxiliary drop-down boxes: "Filter", allowing the user to compare two any individuals on the pattern created (e.g. `?contabenef1 = ?contabenef2`) ; and "Connections", which enables the user to select a minimum cardinality for the object property. In other words, if the value "2" was selected in the "Connections" drop-down box in Figure 4.2, the query results would be restricted to instances of type "Beneficiary account" (`?contabenef1`) with at least 2 object properties "account_person"(`tentaculos#conta__pessoa`) connecting each of them to different people (`Pessoa`).

Appending datatype properties

The second column of the query builder allows datatype properties to be appended to the query. It comprises the same drop-down boxes from the first columns (with the exception of the "Connections" one. The first one ("1. Subject") lists the same values from its right-column counterpart. The second drop-down box ("2. Property") updates with all the datatype properties from the class or instance selected in the first drop-down box. Finally, the third drop-down box ("3. Object") comprises all the possible literals for the selected datatype property.

It is worth noticing that the drop-down boxes have facet query capability. In other words, running a query after each triple is appended would update the literals drop-down with the values

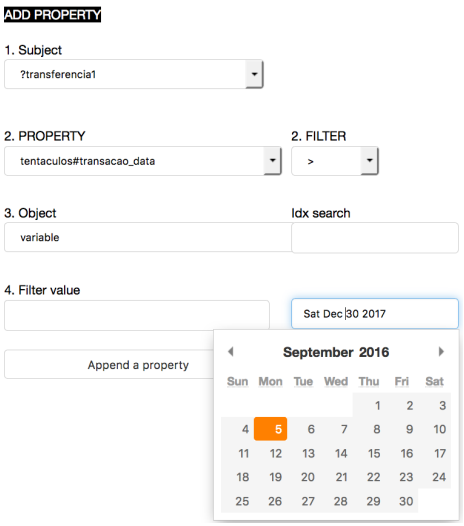

Figure 4.3: Appending datatype property triples to the query.

corresponding to the individuals returned by that query, instead of all literals in the knowledge base for that specific datatype property. This is important to avoid creating complex queries which would return zero results, and also to give insights to the user regarding the data being explored.

Furthermore, the user has other options rather than selecting a value from the that drop-down box. For instance, Figure 4.3 illustrates one more triple being added to the graph pattern displayed in Figure 4.2. In that case, the user is restricting the graph pattern by date: only transfers (`transferencias`) concluded after the transaction date (*transacao_data*) of 5th of September of 2016 would be returned. In addition to *xsd:datetime*[60], datatype properties literals can also be of type *xsd:float*[60]. In that case, typing the value "1000" into the text field "4. Filter value" would restrict the returned transfers to the ones with transaction value (`transacao_valor`) greater than "1000".

Finally, there is the "Idx search" text field, which is one of the new features of version 3 of the prototype (discussed in more detail in Section 4.3.4). It was necessary to implement an indexed search for the datatype property literals due to the large number of possible values stored in the knowledge base (Table 7.2 lists the total literal values on the knowledge base created for the expert validation).

4.2.2 Features 1, 2, 3 and 4 - Facts materialiser

The facts materialiser is the other main feature of the prototype, and it enables the user to create bespoke classes with property restrictions, object properties and datatype properties into the knowledge base. These three possibilities can be interpreted, in an investigation setting, as clusters of individuals, relationships between individuals and pairs attribute-value for each individual respectively. There is also a fourth possibility: to establish identity between any two individuals returned

by the queries. How these capabilities were implemented and why they are important is detailed in the following subsections.

Feature 1 - Materialising classes

Materialising classes can be important in an investigation context to, for instance, group individuals sharing specific properties into a more specialized subgroup, which was not considered at the time of the creation of the base ontology. For instance, the rule in Listing 4.2, once fed to to the *RDFox* reasoner, creates the class `vtinv:new_HT`, grouping all individuals of type `vtinv:File` with at least one object property of type `vtinv:hasJSS-1.0-with` and the datatype properties `cve` and `compression` with literal values *"CVE-2015-5199"* and *"lzma"* respectively.

Listing 4.2: Rule derived from the SPARQL query in Figure 4.4.

```
vtinv:new_HT(?file1) :-
vtinv:compression(?file1, ?compression1),
vtinv:hasJSS-1.0-with(?file1, ?file2),
vtinv:cve(?file1, ?cve1),
FILTER (?cve1 = "CVE-2015-5119"),
FILTER (?compression1 = "lzma")
```

The similar syntax between the SPARQL queries used in the query builder and the *datalog* rules used by *RDFox* to materialise new instances into the data store makes it easy to convert the former into the latter. For instance, the rule in Listing 4.2 was automatically created from the query depicted in Figure 4.4. The user only needs to define a name for the new class (in the case, `new_HT` was chosen) and to select the query variable this new class will be based on. In other words, all the individuals returned by the query and represented by the variable `?file1` will compose the new class.

The next step after the rule is ready is to invoke the reasoner in order to materialize the facts resulting from this rule into the knowledge base. Or, in other words, create the new class `vtinv:new_HT` (which is a subclass of `vtinv#File`, thus inheriting all of its object and datatype properties), comprising all individuals returned by the original query from Figure 4.4. Once materialised, the user can use the recently created class in new queries and rules.

Feature 3 - Materialising datatype properties

Likewise, it is also possible to define a new datatype property for the individuals returned by this query, instead of creating a new class grouping them. That would be recommended in there are multiple values which could be assigned for a specific group of individuals. For instance, the user might prefer to create the property `vtinv#confidence_generator` with value "high" for all `vtinv#Files` returned from the previous query. Later on the investigation, she could define the values "low" or "medium" for `vtinv#Files` returned by other queries. To accomplish that, it would be necessary for the user to choose, in addition to the datatype property name and the variable representing the instances to be materialised, a literal value to the datatype properties to be created. The rule to

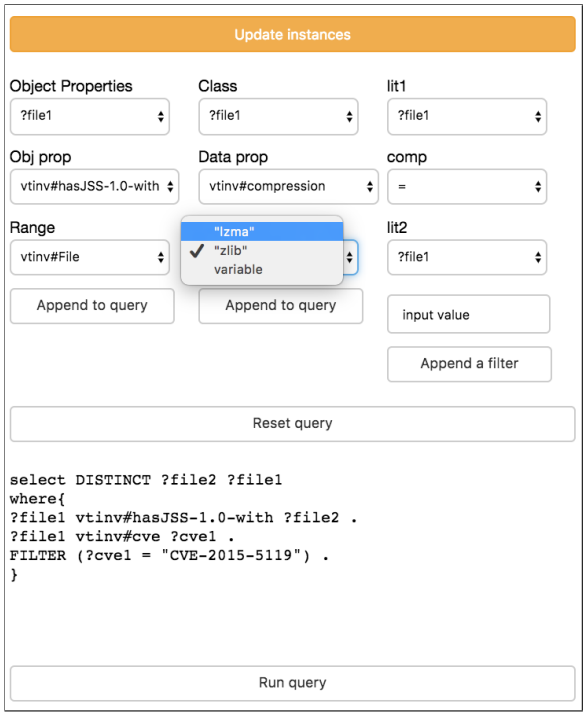

Figure 4.4: Query builder (version 2 of the prototype).

materialise this new datatype property is very similar to the one defining a new class: it is only necessary to replace the first line from the rule in Listing 4.2 with the Listing 4.3:

Listing 4.3: Defining or updating a datatype property.

```
vtinv:confidence_generator(?file1, 'high') :-
```

It is important to notice that the materialisation of such rule will define a new datatype property (`confidence_generator`) with domain `vtinv#File`. As expected, only the individuals returned by the previous query will have the value of this datatype property set to "high".

Feature 2 - Materialising object properties

It is also possible to establish new relationships between any two individuals in the knowledge base. Similarly to the materialisation of classes and datatype properties, the user needs to choose a name for the to-be-created object property, and also pick any two variables returned by the query (instead of one), which will represent the domain and range instances of that object property.

In addition to materialising investigation knowledge (e.g. establishing the relationship " knows" for any two individuals of type Person - `tentaculos#Pessoa`), these bespoke object properties could also optimize the ontology. For instance, they could connect the root and the leaf nodes from graph patterns with more than three nodes in between. That could save some steps in building queries depicting graph patterns constantly searched by the user, ultimately facilitating data exploration. Figure 4.5 shows one example, applied in the dataset for the expert validation: the

original schema exported from database tables did not comprise a relationship connecting people to the states they are from (`Pessoa__Estado`). In order to search for such information, the user would have to build the query depicted in Figure 4.5. Due to the materialisation feature, the user would need to run this query only once, and then create a relationship connecting the variables representing people (`?pessoa1`) and state (`?estado1`). From there on, searching for states where people are from would only require a single triple with the newly-created object property (`Pessoa tentaculos#pessoa__estado Estado`) instead of the three triples depicted in Figure 4.5.

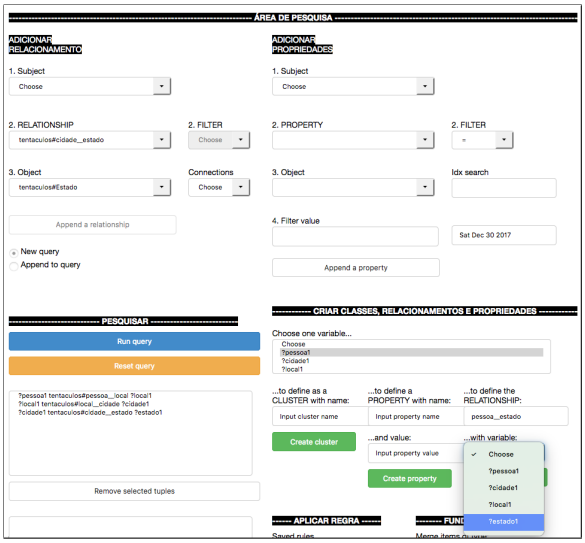

Figure 4.5: Materialising the object property `pessoa__estado` (version 3 of the prototype).

Feature 4 - Enriching

It is also possible to set identity between any two individuals in the knowledge base (in other words, establish that the resources described by both of them actually refer to the same *thing*). This is accomplished by materialising the native object property `owl:sameAs` between them. In a malware investigation, for instance, establishing that things of type `File` which share the same datatype property `md5` value are actually the same `File` is relevant. Figures 4.6 and 4.7 illustrate this situation. On the former, there are two separate individuals of type `File` (in green colour) with the same `md5` but complimentary properties. This is common when investigating data from distinct datasets. The semantic consequence after materialising the object property `owl:sameAs` is that both individuals will now share the conjunction of their respective object and datatype properties (Figure 4.7).

In the prototype, any two individuals linked by the `owl:sameas` property are graphically represented as one bigger circle. This feature was implemented to reduce the visualization clutter, since both individuals still exist within their own namespaces (*`http://ssreport.com`* and *`http://vt.com`*, indicated in Figures 4.6 and 4.7 by the letters 's' and 'v').

Figure 4.6: Before owl:sameas: two resources (in green) referring to the same *file*.

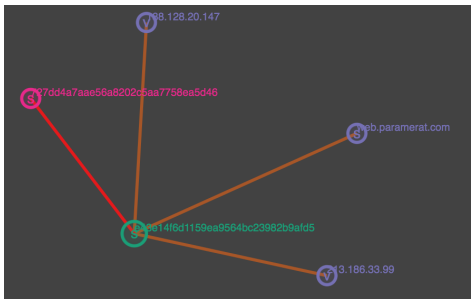

Figure 4.7: After owl:sameas: integrated datatytpe properties (version 2 of the prototype).

4.2.3 Features 5, 6 and 7 - Reproducing investigation steps and rolling back

So far, the features were described in an single-investigation setting, with the analyst defining queries and materialisation rules which are useful to her. However, these same queries and rules could also help other analysts investigating different cases in the same dataset. Because all the rules are automatically saved, it is only a matter of running them again to reach the same dataset state of the investigator who created them, or even adapt them according to the particularities of each case. Such capability is compatible with the reported need for producing and sharing intelligence instead of raw IOCs.

Suppose, however, that the analyst decides that the recently created relationship does not add value to the investigation. In that case, it would only be necessary to load the version prior to this materialisation, and apply new queries or rules from there on. This functionality was implemented in two ways: on the prototype version used in the usability assessment, a new datastore file was saved after each materialisation. Then, to roll back one version of the knowledge base simply meant to run the new queries and materialisations on the previous datastore file. This approach was interesting due to the small size of the knowledge base used in that study.

However, the dataset used on the expert validation study was much bigger, which would cause multiple large datastore files to be saved during one investigation, one per materialisation event. Therefore, a new routine was devised: each investigation starts with two identical datastore files: one is kept as backup, and the second gets updated after each materialisation. That results in storage saving, as there will only be two datastore files at any time. However, the trade-off is the increased effort in rolling back to previous versions. In order for that to happen, the backup datastore file is copied, and all materialisation rules created so far are applied again in the original order.

Defining which approach is best will depend on further real-world testing in order to determine if the feature *rolling back* will be often used and, if so, how long it will take to perform the necessary materialisations. However, implementing this new approach was also convenient to the future

development of sharing and auditing rules, in the case the investigator needs to validate the exact steps taken with a third party. This is specially useful for criminal investigations, as the conclusion of the analysis report could subsidize sending people to prison or freeing them from it.

4.2.4 Features 9 and 10 - Data integration and provenance

The two last semantic capabilities listed in Table 1.1 are related to the merging functionality described in Subsection 4.2.2. Figures 4.6 and 4.7 illustrate two individuals from different datasets which, after a merge operation, are referred to as the same individual. *Keeping track of information provenance* - indicated by the character inside the circles (which represents the first character of the dataset name) and also by the column IRI in the table from Figure 4.1 - is important so the investigator can evaluate information qualities such as confidence. One reported common case is when the investigator needs to compare information from governmental repositories with OSINT.

Data integration, as currently implemented by the prototype, allows the user to add triples (stored in a text file following the *Turtle* syntax, as expected by *RDFox*) to the knowledge base. In addition, the individuals described by the subjects and objects of these triples must be of a type declared by the ontology. Similarly, the predicates of the triples (whether object or datatype properties) must also exist in the ontology, so they can populate the drop-down lists and thus be accessible by the user.

Finally, the `owl:sameAs` object property can also used to define mappings between classes, object and datatype properties from different ontologies (or even to integrate different databases [61]). Although adding `owl:sameAs` declarations mapping the concepts described by new triples with concepts existing in the current ontology has not being tested yet, this can be an important feature if end-users start to add information from already linked datasets. That would eliminate most of the data preparation effort described in Section 5.2.1.

4.2.5 Graph, table and control panel

The graph is the main exploratory tool of the investigator, and it depicts the class and object property instances returned by the user query as nodes and edges respectively. In addition to some native capabilities provided by the Bokeh library (e.g., tools for zooming, selecting and hovering), the graph also comprises some bespoke interaction capabilities such as highlighting the whole tree when tapping a node, choosing query subjects by simply selecting them on the graph, and highlighting specific classes and object properties. The reason why these capabilities were implemented will be discussed in the next section, as it comprises a timeline of the different versions of the prototype which could better illustrate the user needs justifying each capability.

Below the graph there is the table, which displays the datatype properties and the respective literal values from each individual returned by the query. It is divided into tabs grouping individuals

of the same type. In each tab, the columns reflect the datatype properties of each class. Finally, a new table is generated after each materialisation. In other words, defining a new class or datatype property would create a new tab or column, respectively; and updating the literal value of a datatype property for specific individuals would change the value displayed in the related cells.

Finally, to the left of the graph there is the control panel, comprising:

- Control for adding files: used for merging a new graph, stored in text file, into the knowledge base. This operation is executed by RDFlib and, so far, only graphs described by the same ontology have been tested;
- Version control: allows the user to save, delete and navigate between different versions of the knowledge base. A new version is created every time a user materialises a new class, object property or datatype property. The rules defining each of these materialisations are used as input to update or rollback the datastore;
- Visualisation control: enables highlighting specific classes and object properties returned by the query, as selected by the user. In addition, gives the user the option of viewing only the individuals returned by the query, or these individuals plus their direct connections (i.e. object properties not included in the original query);
- Counts control: informs the number of the classes and object properties currently displayed in the graph. It updates according to which option is selected on the Visualisation control.

4.3 Prototype evolution

So far, this chapter has presented how the main features of the prototype were implemented. However, the prototype itself has evolved during the course of this thesis, from a POC to a fully-functional software. There were four versions implemented since the inception of the idea (when some of the items listed in Table 1.1 were tested for feasibility), each one reflecting the requirements of the next evaluating stage. These four versions were described in either accepted papers or a chapters of this thesis, and will be explained next.

4.3.1 Version 0: Feasibility assessment

The main objective of the POC was to assess the feasibility of using semantic technologies to conduct a malware-campaign investigation. Namely, defining and running bespoke queries and rules in a knowledge base and parse the results to validate if they were useful to answer the questioning from an investigation report.

In order to define this knowledge base from structured but not-linked data, there are two important preparation steps which, although not part of the prototype, are fundamental to the success

of any data exploration task using semantic technologies: ontology engineering and conversion to a linked-data format.

For instance, coming up with competency questions to define whether the class *WebServer* (from the investigation report which inspired the case study described in Chapter 5) would have two subclasses (*ExploitServer* and *C2Server*) or not. After reading the report over and over, this was found not relevant to that specific investigation, as opposed to the two subclasses *ExploitFile* and *PayloadFile* of the class *File*, necessary to answer questions such as " Which files were produced by the same generator?".

Technology-wise, the POC relied on the tools *Protege*, *Apache Jena framework*[62], *RDFox* and *Karma*[63]. At this stage, the *RDFox* reasoner started to be considered as a replacement to Apache Jena, for the reasons detailed in 4.1.2.

The initial testing conducted with them have indicated that the answers looked for by the authors of the report could indeed be found using semantic technologies, in a more intuitive way which did not rely at all in writing lines of code to transform the data. These findings were converted into a paper in 2015 presented at IESD[2], a workshop at the ISWC[3]. However, the lack of automation of the POC made reproducing the results and demonstrating its potential to other stakeholders a labour-intensive task, which motivated the implementation of the GUI.

4.3.2 Version 1: GUI (Chapter 5)

The next version of the prototype presented its first GUI, which included the graph, the table and commands to execute the functionalities listed in Table 1.1. Even though it was not optimized for third-parties to operate it, the GUI served its main objective: to demonstrate the idea to researchers from two non-conversant areas (semantic technologies and malware investigation).

Of course, the main visualisation feature of version 1 of the prototype was the graph itself. In addition to the result of the queries, being able to see in a graph the recently-created classes, object properties and datatype properties reflecting the hypotheses tested by an investigator and being able to work with them to deepen the knowledge regarding one case resulted in an article accepted for publication at the IEEE Security & Privacy magazine, further described in Chapter 5.

4.3.3 Version 2: Features usability (Chapter 6)

The focus of version 2 of the prototype was to improve the usability of the prototype so its main functionalities could be assessed by third-parties. That does not mean that this version would provide a fully-optimized and intuitive GUI: after all, doing so would require incremental iterations with the end-users. The main goal of this first iteration was two fold:

[2]Intelligent Exploration of Semantic Data. *Source:* `https://iesd2016.wordpress.com/`, *accessed in* 10/02/2017
[3]International Semantic Web Conference. *Source:*`http://iswc2016.semanticweb.org/`, *accessed in* 10/02/2017

- To confirm that the users would agree with the potential benefit of exploring data using the semantic features listed in Table 1.1 by allowing them to test all the features implemented and analyse the results;
- To capture and fix usability issues in order to improve the prototype before the main validation with expert users.

The improvements of version 2 refer most to the stability of the prototype (i.e. fixing bugs that would cause the execution to terminate), providing better feedback messages to the user during operation, implementing an experiment control logic (which measures time and captures queries and buttons clicked) and other features that would support third-party assessment. The whole experiment in which this version of the prototype was used is detailed in Chapter 6.

4.3.4 Version 3: Scalability (Chapter 7)

One of the objectives of the usability assessment described in Chapter 6 was to identify and fix usability issues that could hamper the final validation study with expert users. Among all the requirements collected, the ones which were effectively implemented are:

Editing queries

The query builder is one of the main features of the prototype. Despite being recognized as relevant by all members of the study, *Creating your own queries* was considered easy only by the members of the technical group (Figure 6.20).

The user has two options when building queries: running them each time a new SPARQL triple is added (thus enabling facet querying functionality) or adding all the triples at once before hitting the button "Run Query". The latter option, which minimizes the time to get to a specific result (as only one query is run) was designed for experienced users who already know what they are looking for. However, complex queries as these could become so restrictive that would often return zero results, and that would demand the user to build a whole new query again.

The caveat of having to build the whole query again was reported by four users of the technical group during the usability assessment (Table 6.6). Therefore, *editing queries* was a priority improvement for the expert validation, and demanded a comprehensive code refactoring. Basically, the necessary changes involved storing the query as a list of triples (displayed as a table) instead of one big string. In addition to improving the back-end control of the components being added by each triple (e.g. variables, filters), this approach allowed the user easily removing a triple from the query. All that is required from her is to select the triple to remove by clicking on the correspondent row of the table, and clicking the button "Remove selected triples".

The query builder also got simpler, with one less column, as Figures 4.4 and 4.5 show: filters are now incorporated into the relationship and property columns. Finally, a new feature for cardinality

restriction was added (the drop-down list "Connections" from Figure 4.5). The need for this one, however, emerged during talks regarding the preparation for the expert validation study.

Improving graph exploration

Improving graph exploration was the top-voted future feature by members of the technical group, which might reflect the potential of such visualization when searching for connected data. Therefore, the following features were added:

- Click and query: in this version it is possible to select the subjects for the new query straight from the graph, being only necessary to click them;
- Highlighting classes and object properties: from a multi-select list menu, it is possible to select different classes and object properties to be highlighted in the graph;
- Tapping items: when tapping one item, the nodes in the whole tree are highlighted, in addition to the direct relationships from the tapped item. All the other nodes and relationships are turned off, making it easier to spot related individuals;
- Support to icons: now the nodes are represented by icons instead of circles, facilitating the analysis of the query results.

Scaling to bigger datasets

The most voted future feature by the non-technical group, *defining ontologies for different datasets*, was partially tackled by this version, as it was necessary to define a new ontology regarding online banking fraud to run the validation with the expert users. Nonetheless, the idea is to allow researchers from different domains to create their own ontology and then explore the resulting linked data with the prototype. This still remains as a future request: even though it will be not implemented by the prototype (as there is better specialized software to do so, like *Protege*), a tutorial comprising procedures, external resources and auxiliary scripts about creating a basic ontology and using it to convert structured data to a linked-data format is expected.

After defining the ontology for the dataset to be used in the expert validation study and converting it to linked data format, an unforeseen issue arose: the prototype was not able to process this much larger dataset. Although not one of the top priorities from the final questionnaire detailed in Chapter 6, *making the prototype work with bigger datasets* automatically became the major necessary improvement for this version. Until the prior version, the focus was on making sure the functionalities implemented were returning correct results. As no performance penalties in operating the prototype had been observed so far, *enhancing data processing* was not a milestone in the development planning.

Even though the experiments described in Chapters 5 and 6 used real-world data, it was based on an single report, and the original dataset would fit in a spreadsheet with less than 50 KB in size. For the expert validation, though, the original dataset was much larger: it refers to three-year online banking fraud investigation data sent from a public bank in Brazil. The dump from the database totalled around 200 MB. Thus, a major refactoring of the code took place, ultimately causing a delay in running the validation with the expert users.

The necessary improvements regarded both the front-end and the back-end of the prototype, and some trade-offs had to be considered. For instance:

- If a query returns nodes with a high number of connections, should all of them be displayed in the graph? What is the balance between completeness and graph clutter?
- The datastore file for each version is not being saved any more, as it would consume a large portion of storage. However, the alternative to that makes navigating through different versions slower.

The next chapter will detail how the prototype was used to successfully answer the first research question.

Chapter 5

Case study

The first assessment of the prototype comprises an exploratory case study intended to answer by demonstration the first research question discussed in Section 1.4: "Could the semantic features in Table 1.1 be used to perform the corresponding tasks in a real investigation scenario?" For that, a specific malware investigation report was chosen (for reasons described in the following section) and the investigative steps described in that report were reproduced using the prototype presented in Chapter 4.

For evaluation purposes, a qualitative metric was chosen not only due to the lack of accredited benchmarks for cybercrime investigation tools but most importantly for not having identified any other tool which offers similar capabilities as this prototype does. For instance, enabling the investigator to automatically create multiple relationships between instances from a large dataset. The tool that gets the most closer to that, and has been used by many LEAs [64] is IBM's I2 Analyst Notebook [65]. Despite being a comprehensive tool comprising many analysis possibilities, the I2 Analyst Notebook was not developed to allow frequent nor batch transformation of data (e.g. on it, relationships must be drawn manually for each pair of instances [66]).

Therefore, the "Goal-Question-Metric" approach [33] was chosen to define whether the prototype was feasible and suitable for investigating malware campaigns:

1.
   - Goal: Feasibility.
   - Question: Could the semantic capabilities of the prototype be used to arrive at the same conclusions of the authors of the report without making use of any other manual resources such as taking notes or tabulating data?
   - Metric: The number of investigative tasks in the report which could be successfully reproduced by the prototype.

2.
   - Goal: Suitability.
   - Question: Could the prototype be used to analyse further data regarding malware campaigns not originally described in the report?

- Metric: The effort in number of steps from merging more data to the dataset to acknowledging how it changes the original conclusions.

The results from this first experiment proved that is was indeed feasible to reproduce all the analysis tasks (creating the same clusters and relationships described in the report) in a timely and easier way, which led to the same conclusions as the authors, by only using the prototype. That was only possible due to the capabilities of creating meaningful clusters and relationships in "batch-mode" and "on the fly" during the analysis process, which are not known to exist in any other software, according to [12] and also to the literature review.

Moreover, the method described in this chapter was also proven suitable for investigating general malware campaigns, in the sense that it allowed to extend the conclusion of the authors by adding and analysing more recent data to the investigation, which was not available at the time the report was written.

The following sections will explain the reasons for choosing the report used in this case study, describe all the investigation steps taken by the authors of the report and present the results for both reproducing those same steps of the authors and extending their conclusions by only using the prototype.

5.1 The *Italian Connection* report

A significant cyber-security event happened in 2015, in which data from the company Hacking Team[1] was leaked to the internet. Among the exposed files, there were some zero-day[2] vulnerabilities which were promptly harnessed by other hackers.

Following this breach, researchers at Shadow Server Foundation published a report [67] aiming to revealing relationships between supposedly independent groups based on similarities across IOCs such as payload[3] and command-and-control (C2) infrastructure. This report will henceforth be referred as *ItCo*, in reference to its original name "The Italian Connection".

Differently from most reports describing malware campaigns, the *ItCo* report clearly states the methodology used (which investigative step was taken at each stage, and why), explains the rationale behind each assertion they make (e.g. "We define independent operators as actors that maintain distinct infrastructure without any technical overlaps such as ip history.") and concludes by presenting

[1] Hacking Team is an Italian information technology company that sells offensive intrusion and surveillance capabilities to governments, law enforcement agencies and corporations. *Source:* `http://en.wikipedia.org/wiki/Hacking_Team`, *accessed in* 24/04/2016.

[2] A zero day vulnerability refers to a hole in software that is unknown to the vendor. This security hole is then exploited by hackers before the vendor becomes aware and hurries to fix it. *Source:* `http://www.pctools.com/security-news/zero-day-vulnerability/`, *acessed in* 24/04/2016.

[3] In computer security, payload refers to the part of malware which performs a malicious action. *Source:* `https://en.wikipedia.org/wiki/Payload_(computing)`, *accessed in* 25/04/2016.

an analysis of competing hypotheses [12] regarding whether different exploits might have a unique origin. Its quality was also asserted by at least one more cyber-security analyst [68].

The dataset used by the authors of the *ItCo* report comprises a total of 52 Adobe Flash files exploiting either CVE[4]-2015-5119 or CVE-2015-5122, which are the vulnerabilities related to Hacking Team leaks. These samples were obtained by crawling websites known for distributing malware and also searching online repositories such as *VirusTotal* and *Shadow Server*. For each sample, IOCs such as domains, ips, hashes and compression method were extracted by static and dynamic sandbox[5] analysis.

The first step of the investigation was to distinguish the exploit files produced by a common *generator tool*[6] from the ones obtained via *source-code sharing*. According to the authors' knowledge, exploit files produced by the same generator tool share specific features. For instance, they used the following values to define one of the clusters:

- Created on the same date of 7/7/2015;
- Targeted the same vulnerability of CVE-2015-5119;
- Compressed via the LZMA algorithm;
- Contained an embedded payload;
- Had identical *ActionScript* classes.

All exploits produced by one *generator tool* would necessarily have all embedded *Action Script* classes identical. However, some other exploits, despite sharing most features with each other, would have one or more differing classes. In these cases, the authors of the *ItCo* report would classify such exploits as *source-code sharing*.

The authors of the *ItCo* report did not mention if an automated method was used to compare the classes from each file. Thus, the approach used in the experiment was to first automatically extract all the classes from each file and produce their MD5 hashes. Then, to calculate the Jaccard Similarity Score (JSS) for every two files, using their internal classes MD5 hashes as the set of features. The pairs with a score of 1 were attributed the relationship `hasJSS-1.0-with`, and those with a score between 0.9 and 0.99, `hasJSS-0.9-with`.

By assessing the different clusters created (two *same-generator* clusters and three *shared source-code* ones) and which of their members would potentially have a unique origin (i.e. the same actor),

[4]CVE is a dictionary of publicly known information security vulnerabilities and exposures. *Source:* `https://cve.mitre.org/`, *accessed in* 27/04/206.

[5]A sandbox is a security mechanism for separating running programs. It is often used to execute untested code, or untrusted programs. *Source:* `https://en.wikipedia.org/wiki/Sandbox_(computer_security)`, *accessed in* 27/04/2016.

[6]Generator tools, or exploit kits, enable "...an operator to quickly and easy bind a payload or remote download url to shellcode in the flash exploit file via a handful of mouse clicks or a simple command." [67]

the authors established competing hypotheses about the exploits supply-chain. These were mostly related to the malware-development skills of distinct actors, which groups would be "collaborating" among themselves and the speed with which their exploits were deployed into the wild.

The *ItCo* report is an excellent opportunity to validate the proposed approach, presented before in a position paper [69]. At that time it was difficult to implement due to the lack of specific domain knowledge (i.e., the rationale used by malware investigators) and the difficulty of obtaining suitable datasets for the case study (now made available by the authors of the report as a spreadsheet containing all the IOCs).

5.2 Semantic investigation

This section will demonstrate how the *ItCo* report was fully reproduced by harnessing some of the capabilities mentioned in Table 1.1 in the prototype presented in Chapter 4. Initially, it will explain the steps taken to convert the structured data provided by the authors into a linked format, which is necessary so it can be loaded into the prototype. Then, the investigative steps taken by the authors will be reproduced using the prototype, and details about how the leveraged materialisation features could make the data exploration work more efficient will be discussed (e.g. representing related-malware as distinct classes with property restrictions instead of exploring them in a spreadsheet). Finally, the *merging* and *dataset integration* features will be proposed as a extension of the report, in order to prove how they could quickly add extra knowledge to the investigation.

5.2.1 Converting structured data to linked data

First, it is necessary to define the ontology, which will be the basis for converting data to a linked format. It is possible to either extend the concepts from an existing ontology or create a new one. A new ontology was created for the case study as there was not a widely accepted ontology for malware investigation at that time, but only taxonomies which do not follow semantic-technology standards, as Section 2.3 shows.

The *datatype properties* of this ontology derived from the column headers of the IOCs spreadsheet are shown in Table 5.1. The first four rows also represent *classes*, as `md5` and `domain` were chosen to compose the URI of the entities `File` and `Webserver`, respectively.

Figure 5.1 illustrates the resulting ontology in *vowl* [70] notation: circles are *classes* and green rectangles are *datatype properties*. The blue rectangles represent *object properties*, or relationships between *classes* reflecting the knowledge disclosed by the authors in the report. Note there are no individuals represented in this figure.

Once the ontology is ready, it is necessary to map the data (in this case, the information from the IOCs spreadsheet) onto it. This was accomplished using the tool *Karma*, which provides an intuitive interface to create the mapping model and later allows for batch processing.

Table 5.1: Mapping IOCs to semantic concepts.

Column headers	**Classes and datatype properties**	**Sample values**
swf md5	File (Exploit File) and md5	e33cf5b9f...71f4380dd7eb1
payload md5	File (Payload File) and md5	5a22e5aee...77f1351265a00
exploit site	WebServer (ExploitServer) and domain	news.turkceil.tk/movie.swf
C2	WebServer (C2Server) and domain	amxil.opmuert.org
payload in swf	embeddedPayload	yes
create date of swf	createDate	7/14/15
swf cve	cve	CVE-2015-5122
swf compression	compression	lzma
payload family	family	PlugX

The resulting linked data resembles the individuals depicted in Listing 3.1: *ssreport.com* is the bespoke namespace defined for the data, and *vtinv* is the ontology comprising, among others, the property `connectsTo`. *1a2b3c4d5e6f* and *evil.org* are the identifiers for one `File` and one `Webserver` within the namespace, respectively.

Finally, the knowledge base (comprising both the ontology and the data) was loaded into the prototype. The initial number of individuals is shown in Table 5.2, column 0.

Table 5.2: Evolution of the number of individuals.

Iteration	0	1	2	3	4	5	6	7
File	74	**43**	43	**70**	**43**	**198**	**161**	**137**
new_HT	-	**11**	11	11	11	11	11	**31**
new_002	-	**12**	12	12	12	12	12	12
new_exp1	-	**6**	6	6	6	6	6	**10**
new_exp2	-	**2**	2	2	2	2	2	2
WebServer	65	65	65	**81**	**71**	**109**	**103**	103
hasJSS-1.0	230	230	230	230	230	**1174**	**970**	970
hasJSS-0.9	150	150	150	150	150	150	150	150
connects	42	42	42	**62**	**50**	**96**	**91**	91
distributes	36	36	36	36	36	36	36	36
embeds	37	37	37	37	37	**73**	73	73
sameC2as	-	-	**14**	14	14	14	14	**38**
sameC2ESas	-	-	**2**	**-**	-	-	-	-

Bold denotes a change in value.
The symbol "-" means the concept is not present in that iteration.

The number of *Files* in each iteration includes both `ExploitFiles` and their embedded `PayloadFiles` (obtained from the columns *swf md5* and *payload md5* of the IOCs spreadsheet). Therefore, there is no conflict with the initial count of exploit files given in Section 5.1 (52).

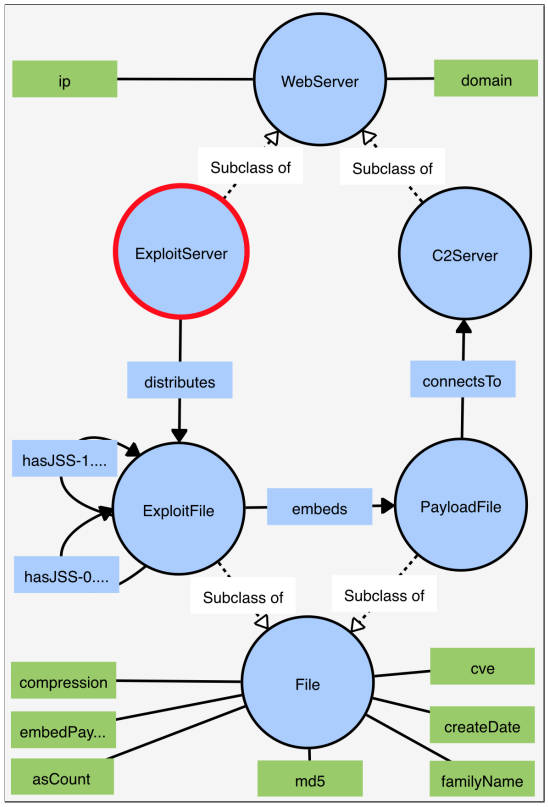

Figure 5.1: The *vtinv* ontology, designed specifically for this case study.

5.2.2 Facet querying and defining classes

As mentioned before, the initial step of the authors of the *ItCo* report was to cluster files potentially created by the generator. After exploring the data, they came to the cluster definition given in Section 5.1.

Although the report does not mention their exploratory process, it was possible to simulate it using facet querying: first, `Files` which hold the relationship `hasJSS-1.0-with` with another `File` were queried. The analysis of the results made it clear that `Compression` and `CVE` would be good features for clustering, and thus should be appended to the query as new filter triples, as illustrated by Listing 5.1.

This is a basic use of the facet querying, and the fact that the analysis refers to previously processed data certainly made it easier to spot good cluster features. Hence, a more complex query will be demonstrated below in Subsection 5.2.3, and Subsection 5.2.4 will demonstrate the approach within a larger dataset.

Running the query from Listing 5.1 on the knowledge base yielded the same results as Table 1 from the *ItCo* report (which enumerates the members of the *HT_ exploit* cluster). If deemed relevant for the investigation, the individuals returned by this query could easily be grouped into a new class, which would then be available for further querying.

Listing 5.1: Searching for files potentially from the same generator tool

```
select DISTINCT ?file1 where
?file1 vtinv:cve ?cve1 .
?file1 vtinv:compression ?compression1 .
?file1 vtinv:hasJSS-1.0-with ?file2 .
FILTER ?cve1 == "CVE-2015-5119".
FILTER ?compression1 == "lzma" .
```

Going back to this facet-querying exploration process, it was possible to identify and define one more *same generator* class, comprising files exploiting *CVE-2015-5122* and compressed with the *lzma* algorithm. These coincide with cluster flash_exploit_002[7], described in Table 4 of the *ItCo* report.

In a similar fashion, two classes of *shared-code* exploits were created: *new_exp1*[8] (Table 7 from *ItCo* report) and *new_exp2* (Table 10 from *ItCo* report). The updated number of individuals at this point in the investigation is given in Table 5.2, column (iteration) 1.

5.2.3 Establishing links

The authors of the *ItCo* report define independent actors as the ones who "...maintain distinct infrastructure without any technical overlaps such as ip history".

Thus, two `ExploitFiles` would potentially share the same actor if (1) they are being distributed by the same `ExploitSite` or (2) their embedded `PayloadFiles` connect to the same `C2Server`[9].

In order to find out about (1), a simple and direct query would be enough : *Find all* `ExploitFiles` *that are* `distributed` *by the same*[10] `ExploitServer`. On the other hand, finding about (2) would require a "*join*-like" operation which is better handled by graph-based technologies. After all, it would involve traversing two relationships (`ExploitFile` *embeds* `PayloadFile` and `PayloadFile` *connectsTo* `C2Server` – see Figure 5.1) for all `ExploitFiles` in the knowledge base to find out which of them share the same `C2Server`.

The native linking structure of the triples makes it easier to define this *sparql* query as depicted in Listing 5.2:

Listing 5.2: Querying for different ExploitFiles (?file1 and ?file3) with matching C2Servers (?webserver1).

```
select DISTINCT ?file1 where
?file1 vtinv:embeds ?file2 .
?file2 vtinv:connectsTo ?webserver1 .
?file3 vtinv:embeds ?file4 .
?file4 vtinv:connectsTo ?webserver1 .
FILTER (?file1 != ?file3) .
```

[7] Contains one less file, not mentioned in the written report but present in the spreadsheet.

[8] Contains one less file. This was expected, since it has compression value none.

[9] Command-and-control (C&C) servers are used to remotely send often malicious commands to a botnet, or a compromised network of computers. *Source:* `https://www.trendmicro.com/vinfo/us/security/definition/command-and-control-(c-c)-server`, *accessed in 12/04/2017*

[10] Domain resolution will not be considered as it is not present in the original data. However, a more precise *Webserver* definition, considering its assigned IP on a particular day, could also be modelled.

Table 5.3: Correspondences depicted in Figure 5.2.

Actor	***ItCo table***	**ExploitFile**	***ItCo cluster***	**WebServer**
APT18	3 6	079a440bee0f86d8a59ebc5c4b523a07 726bd0bd6cca8d481cf6165c95528caa	HT 002	223.25.233.248
UNK1	6 6 3	b65076f4cb6e74429dd02fcacda0bec3 8a8e9bbf1ca2a926f0a5d06217eeea55 f46019f795bd721262dc69988d7e53bc	002 002 HT	nfitsub.com
APT20	8 12	c101d289d36558c6fbe388d32bd32ab4 195bdc84f114c282e61f206dc88cd26d	EXP1 EXP2	win7.myz.info
DNSCalc/APT12	15 15	edcd313791506c623d8a2a88b9b0e84c 83388058055d325a2fa5288182a41e89	MOVIE MOVIE	213.186.164.211 202.183.129.155
UNK10	6 6	451c52652ddb28e9071078f214a327a7 e33cf5b9f3991a8ee4e71f4380dd7eb1	002 002	amxil.opmuert.org

This search returned 11 different `ExploitFiles`. If this result is considered relevant, the related query could be transformed into a rule for defining the new object property `sameC2as`, similarly to the procedure described in Section 5.2.2. Feeding this rule to the reasoner resulted in the materialized knowledge depicted in Figure 5.2. The yellow edges represent the recently created instances of the *sameC2as* object propertuproperty, connecting the `ExploitFiles` returned in the original query.

These files belong to different clusters (new_HT, new_002, new_exp2 and new_exp1) derived from previous class materializations. In addition to listing their *md5* hashes, Table 5.3 also informs that the conclusions of this case study regarding `ExploitFiles` belonging to the same actor agree with ones from the *ItCo* report: column *ItCo table* indicates the original tables comprising these files and their matching `C2Server`.

Creating bespoke relationships allows to promptly reuse them as necessary. For instance, the definition of *same actor* could be restricted by considering (1) *and* (2) instead of (1) *or* (2). The new `sameC2ESas` rule, building upon `sameC2as`, would then be:

Listing 5.3: Defining a more restrict definition of *same actor*.

```
vtinv:sameC2ESas(?file1, ?file3) :-
vtinv:distributes(?webserver0, ?file1),
vtinv:distributes(?webserver0, ?file3),
vtinv:sameC2as(?file1, ?file3) .
```

Once materialised, this last rule added two instances of the object property `sameC2ESas` to the knowledge base, linking two distinct `ExploitFiles`, as described by column (iteration) 2 of Table 5.2. However, it was noticed thah both `ExploitFiles` embedded the same `PayloadFile`. Because this revelation was not considered relevant, the investigation was rolled back by loading the previous state of the knowledge base (i.e. before the materialization of the object property `vtinv:sameC2ESas`).

Table 5.3 lists the correspondences between the *ItCo* report tables and the results from this case study in regard to `ExploitFiles` supposedly from the same actor.

At this stage using the *Seminv* method has reached the same conclusions of the authors of the *ItCo* report in relation to which `ExploitFiles` would come from the same actors. To discover if the

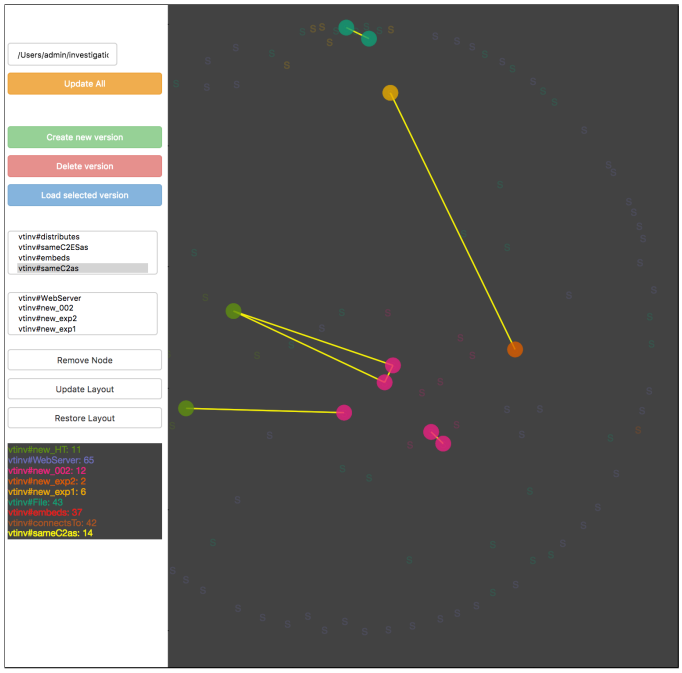

Figure 5.2: Members from different clusters linked by the bespoke relationship `sameC2as`.

Seminv method could add additional insights using the *merging* and *data integration* capabilities listed in Table 1.1 and described in Sections 4.2.2 and 4.2.4, further data enrichment was undertaken.

5.2.4 Enriching existing entities

So far the current semantic investigation has reached the same conclusions of the authors of the *ItCo* report in relation to which `ExploitFiles` would come from the same actors. As an extended validation, the original dataset will be enriched with both `ExploitFiles` and `PayloadFiles` subsequent to the publication of the *ItCo* report. This will make use of other capabilities mentioned in Table 1.1, and the goal was to check if any new relationships would emerge. There are two possibilities for data enrichment, which are not mutually exclusive: enriching data about existing entities or adding new entities.

The existing `PayloadFiles` within this case study were enriched with network data from *VirusTotal*. After searching this repository using their md5 hashes, it was possible to download a total of 27 file reports.

In the same way as described in Section 5.2.1, first it is necessary to define a mapping between the current ontology and the *VirusTotal* network reports. For demonstration purposes, only the object property `connectsTo`, and the datatype properties `ip`, `domain` and `md5` were mapped from the network reports.

The resulting linked-data extract of the network reports comprised 20 `Files` (out of the initial 27) holding the relationship `connectsTo` with 16 `WebServers`. After defining the namespace `http:`

//vt.com/, the resulting graph was merged to the current knowledge base and loaded it into the prototype, in order to extend the investigation.

As expected, the counts of `File`, `WebServer` and `connectsTo` have increased, as shown in Table 5.2, column (iteration) 3. Even though all of these "new" `File` instances refer to pre-existing `PayloadFiles`, they are still distinct resources (e.g. `http://ssreport.com/e43e14...b9afd5` and `http://vt.com/file-e43e14...b9afd5`), which might hold identical or complimentary information about the same *thing*: the file with md5 hash *e43e14...b9afd5*.

As explained in Section 4.2.2, the object property `owl:sameas` was defined for every two or more files holding the same md5 value. Column (iteration) 4 of Table 5.2 shows that, after the materialization of the `owl:sameas` property, the quantity of files decreased to the same number as iteration 2 (i.e. before the addition of the network reports in iteration 3). This was expected, since only reports for files pre-existing in the knowledge base were downloaded.

For exploration purposes, the `owl:sameas` property was also established for any two `WebServers` with matching `domain` or `ip`. Differently from `Files`, the difference between versions 2 and 4 indicates that there are six new `WebServers` in the knowledge base, out of the sixteen added in iteration 3.

5.2.5 Adding new entities

Following the procedure described in Section 5.1, *VirusTotal* was searched for the tags *CVE-2015-5119* and *CVE-2015-5122* in April 2016. It was possible to download 155 new files, comprising both exploits and payloads.

After extracting the *ActionScript* classes from the Adobe Flash exploit files and calculating their JSS, the data was converted to linked format and merged into the knowledge base. Column (iteration) 5 of Table 5.2 gives the new individuals count, and column (iteration) 6 reflects the counts resulting after applying the `owl:sameas` property to both `Files` and `WebServers`.

Then, the rules defining the four clusters and the relationship `sameC2as`, saved automatically during the investigation, were applied to the enriched knowledge base. After the last materialisation (column 7 of Table 5.2), cluster `new_HT` was automatically populated with twenty new members and cluster `new_EXP1` with four new ones.

Moreover, it was easy to spot three new sets of `ExploitFiles` holding the property `sameC2as` among themselves, as indicated in Figure 5.3. They are all members of cluster `new_HT`, endorsing the hypothesis in page 17 of the ItCo report: "The model of a single quartermaster developing and sharing generators would explain the identical nature of the malicious ActionScript classes in the HT_Exploit and flash_exploit_002 clusters."

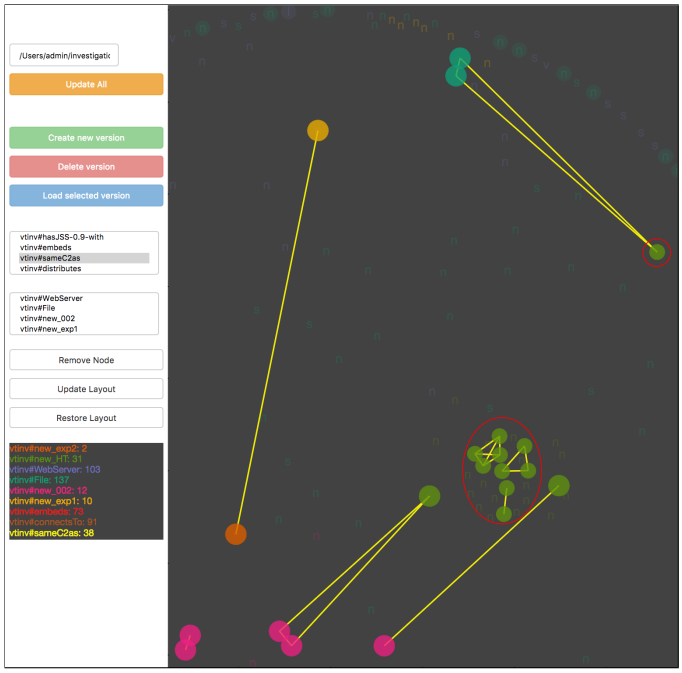

Figure 5.3: Result of iteration 7 revealed ten `ExploifFiles` holding the relationship `sameC2as`.

5.3 Conclusion

This case study positively answered the first research question from Section 1.4. After all, it demonstrated how some of the features listed in Table 1.1 were used to reproduce a real-world investigation, and that the conclusions achieved by applying them were compatible with the ones from the authors of the report this case study was based on. Because the authors did not mention nor replied to the questioning about the use of any tools in support of their investigation procedures (i.e. how did they build the clusters and search for relationships between them), it is supposed that they had performed "manual" analysis of the data (e.g. using bespoke scripts and spreadsheet-like tools). And the overall aim of the prototype is exactly to make such data exploration simple, flexible and fast enough so analysts with no background in programming or data science but experienced in investigations can focus on the interpretation of the results, avoiding much of the time-consuming, error-prone work in manipulating and correlating data.

Moreover, the enrichment features showed how it was possible to extend the analysis performed and achieve updated results after adding data to the investigation. The main benefit of the semantic approach, in addition to the expressiveness of the queries and rules, is the easiness to verify whether this new data would fit into one of the clusters or relationships created, as their defining rules were automatically saved and could be applied again any time. Furthermore, being a rule-based analysis tool means that batch processes could be set to search data and warn the human analysis once new matches are found. This would be specially useful in analysing datasets which update constantly.

From the features listed in Table 1.1, only "3. inserting tags or comments", "5. rolling back in case of dead end" and "7. sharing investigative knowledge" were not demonstrated in this case study. Because the former is used actively in malware-investigation (as mentioned in Section 2.4), it might have helped the authors to organize the malware before grouping them into different tables, even though no pre-processing tasks were disclosed in the report. Finally, features 5 and 7 were not demonstrated as the *ItCo* report already comprised the final results of the investigation. Nonetheless, they could have been useful if the authors needed to discuss their findings with other malware experts before producing the final report experts, or to try different cluster definitions until reaching the one informed in Section 5.1.

Finally, it must be noticed that there was a considerable preparation step in this case study, mainly because it was necessary to first convert the original dataset to a linked format. However, once a mapping file for a specific data source was produced, the rest of the process could be automated. Interestingly enough, the increasing adoption of JSON-LD [49] (JSON for Linked Data) within the web community (search engines like Google are recommending it for marking up data [71]) could motivate online data providers to publish their JSON data in JSON-LD. After all, the compatibility between both formats means that only small modifications would be necessary to convert the former into the latter.

That would certainly decrease the burden of the preparation step, and could consequently foster the development of novel semantic approaches to the exploration of the available data sources on the web. Moreover, the active research regarding the scalability of current reasoners and benchmarks for linked data processing [57] could foster the development of novel semantic approaches to the exploration of the available data sources on the web.

Chapter 6

Usability assessment

This chapter describes a usability assessment of the *Seminv* prototype, designed to check whether the user can successfully operate the main features in the context of a synthetic case study, and to identify any issues with the general interface design ahead of the main validation with the expert users. In addition, this assessment will also serve to estimating the learning curve of the prototype.

The assessment comprises four main phases: introduction questionnaire, training, tasks and final questionnaire, which are described in the next section. There were no personal identifiable information collected and all the procedures and documents presented to the participant were approved by the Central University Research Ethics Committee, reference R50632/RE001.

6.1 Method overview

For this usability assessment, ten participants were recruited among students and researchers from the Cyber Security Analytics Group at University of Oxford. The restriction to this department was because although the experiment does not require working knowledge about malware campaigns, a basic understanding of concepts related to cyber security such as "file payload" and "command and control servers" is recommended in order to create the necessary queries.

The usability referred to in this assessment is more related to the technology itself (i.e. whether other people can use and evaluate it) rather than the graphical interface (i.e. whether it is fully optimized and easy to navigate). After all, achieving a simple but powerful interface requires multiple interactions with the end user. Thus, the evaluation criteria for this assessment was not only measuring how many participants could finish all the tasks. Due to the novelty of the approach proposed, it was very important to figure out whether the users could understand and apply the techniques available and how such technology could improve their common daily tasks.

This experiment employed both quantitative (e.g. experiment tasks and logging) and qualitative (e.g. user observation and discursive questionnaire) methods, commonly used for assessing user-adaptive systems and which "...derived both from the evaluation of human–computer interaction systems and from information retrieval and information filtering systems" [72]. Being an information

filter and retrieval system which provides different perspectives of the data according to the user input, some of the assessment methods discussed in [72] proved suitable for evaluating the current experiment:

- Collection of users opinions using pre and post-experiment questionnaires;
- Experiment tasks: necessary to estimate if the users could operate the prototype, and collect suggestions for improvement before the main validation;
- Observation and usage monitoring:
    1. Logging: number of clicks and time taken to complete the tasks;
    2. User observation: notes regarding the execution of tasks for each participant were taken and later compiled into a spreadsheet. In addition, the screen interaction together with the audio were also recorded;
    3. Think aloud protocol: capturing potential difficulties faced by the participants during the experiment and any unforeseen issues.

The experiment setting resembles a real investigation, in which it is hardly possible to arrive at a conclusion or build insights with only one step. Therefore, the tasks follow a natural order, in which the result of an individual task depends on the correct accomplishment of the previous one. Finishing all the tasks was also important to make sure the participants understood the connection of the different cognitive activities as part of a bigger process, so they could rate their experience afterwards. The think aloud protocol proved specially useful in revealing whether the users needed help in executing a particular task: whenever they had any issue, they could ask for a "minor" advice. If even so they could not finish the task, then they were guided through it. The difficulty in completing each task, which was measured using three possible values ("solved alone", "minor advice given" or "guidance needed"), was important to estimate the learning curve of the participants, considering that some tasks required similar actions from them.

6.1.1 Initial questionnaire

The objective of the initial questionnaire is to identify any background or demographic factor which could aid the analysis of the experiment, by correlating them with the performance of the tasks or the responses to the final questionnaire. There are:

- four questions about demographics: age, gender, working sector and duration of employment in years. Even though all participants are from the Cyber Security Analytics group, their previous work occupation could reflect on their ability in operating the prototype;

- three questions about specific background: whether the participants had knowledge about malware investigations and semantic web technologies, as well as which aspect of cyber-security they were more conversant: social or technical. These questions are important to identify potential distinct performances of the participants in understanding and completing the tasks;
- two questions about querying: whether they had already written a SQL-query of their own, and how often did they have to search structured datasets as part of their work. These two questions are relevant because building queries is a major factor of the prototype. Moreover, the question addressed SQL and not SPARQL queries due to the former being much more popular and resembling the latter;
- one question about visualisation, aiming to identify the most important visualisation types for their work. This is important to determine if any participant is already used to the visualisation type leveraged by the prototype (graph-based), which could cause a better performance in some tasks. Also, the prototype makes use of text and table-based ones, in a minor scale;
- one question asking the participants to value generic features during dataset exploration (e.g. "tool automation" and "abstraction of data"). These replies will be compared to the ones from question 9 of the final questionnaire, which asks about the importance of specific implemented features. Both groups of answers are related and, together, they will indicate future developments of the prototype.

6.1.2 Training

The goal of the training stage is to get the participants engaged with the experiment and to provide them with the knowledge necessary to complete the tasks. This process was divided in three parts:

1. Description of the experiment ontology with a brief explanation of linked data and the following semantic concepts: ontology classes, object properties and data properties; triples; sparql queries and namespaces;
2. Watching five videos, with a total duration of 10m42s, demonstrating the main features and how to operate the prototype;
3. Free exploration period, in which the participant could try bespoke actions with the prototype and reproduce the examples from the videos.

Once the participants felt comfortable and confident with operating it, they could proceed to the tasks phase.

6.1.3 Tasks

The experiment tasks were based on the case study discussed previously in Chapter 5. There were a total of ten specific tasks exploring the main features of the prototype: querying, materialisation, enrichment, graph exploration, knowledge reuse and creating new versions. They follow a logical-temporal sequence which, although resembling a real investigation, do not require any additional knowledge from what was introduced to the participants during the training phase.

Task 1

- **Question**: How many ExploitFiles have a relationship has-JSS-1.0.with another ExploitFile?
- **Main theme**: Query.
- **Reason**: In order to start any investigation, an initial query is necessary. This is a simple one, which can be easily transcribed from the statement to the query.
- **How to solve it**: Using only the relationship column of the query builder, append the triple `?exploitfile1 vtinv#hasJSS-1.0-with ?exploitfile2` and hit run query.
- **Expected outcome**: Make the user realize that running a query updates the graph and the counts panel. The user should answer "28", which is the number of `ExploitFiles` holding the relationship `has-JSS-1.0.with` with another `ExploitFile`.

Task 2

- **Question**: Among the ExploitFiles returned in Task 1, how many also have the properties cve = CVE-2015-5119 AND compression = LZMA AND ascount = 7?
- **Reason**: The query builder offers the capability of appending each triple separately, based on the assessment of the previous query. This could be useful to the participants, allowing them to reason upon smaller datasets each time. In this case, considering only the subset of 28 `ExploitFiles` from Task 1.
- **Main theme**: Query.
- **How to solve it**: Using only the properties column of the query builder, append three triples: `?exploitfile1 vtinv#asCount "7", ?exploitfile1 vtinv#compression "lzma"` and `?exploitfile1 vtinv#cve "CVE-2015-5119"`, without resetting the previous query.
- **Expected outcome**: Check whether the participants will reset the previous query or not, as it is not necessary, and also how many of them will choose the same variable for all triples (either `?exploitfile1` or `?exploitfile2`) , thus defining only one graph pattern. That would represent a basic understanding of linked data, as explained in Section 4.2.1. The answer for this question is "11".

Task 3

- **Question**: Define a cluster named HT, comprising the results of the last query. Which version are you in now? How many members on cluster HT?
- **Main theme**: Materialization, creating a new version.
- **Reason**: One of the key capabilities of the prototype is allowing the participants to easily define their own clusters by creating subclasses with bespoke property restrictions from a specific existing class. Such cluster would reflect their own knowledge towards a subset of the data, as it is the participants who choose its defining properties and values. In this case, `HT` is a subclass of `ExploitFiles`, and comprises only the `ExploitFiles` which were possibly originated from the same *generator kit*, as explained in Chapter 5.
- **How to solve it**: In order to define a cluster comprising the 11 `ExploitFiles` returned in Task 2, the participant must use the materialisation feature: select the same variable used in the triples (`?exploitfile1`), insert the name "HT" and click on "Create cluster".
- **Expected outcome**: The participant should realize that there is a new version of the dataset now ("version 12"), with the 11 returned `ExploitFile` from Task 2 grouped in the recently created cluster (or class) `HT`.

Task 4

- **Question**: How many web servers are distributing files from cluster HT? Is there any file being distributed by two web servers? If so, which (inform only the 4 initial digits)?
- **Main themes**: Query, knowledge reuse, graph exploration.
- **Reason**: Assess whether the user will realize two things: that their recently created cluster is now available for using in new queries and comparing to other data, and that some graph-exploration tools could indeed facilitate knowledge discovery.
- **How to solve it**: Similarly to Task 1, the participants must build and run a new query, this time using the recently created cluster `HT`. There is only one triple necessary for that: `?webserver1 vtinv#distributes ?ht1`. Now, in order to identify any file being distributed by two web servers, the graph must be analysed. Because all the relationships from the returned entities are displayed (as shown in Figure 6.1), the participant should make use of the relationship-highlighting feature. Selecting only `vtinv#distributes` will disable the other relationships, making it easier to spot which file, among the nine returned, is being distributed by two web servers, as depicted in Figure 6.2.

- **Expected outcome**: The participant should realize that there is an extra step necessary after running the query for the first question. Although it would be possible to manually search the nodes returned in the graph depicted in Figure 6.1 (as there are not many of them), highlighting the relationship would be recommended. Another solution would be to run a different query, such as the one in Listing 6.1. This would be recommended if the graph would remain cluttered even after the highlighting step (Figure 6.2), and would return a more specific result, as shown in Figure 6.3.

Listing 6.1: Alternative way to complete Task 4.

```
?webserver1 vtinv#distributes ?file1 .
?webserver2 vtinv#distributes ?file2 .
FILTER (?webserver1 != ?webserver2) .
FILTER (?file1 = ?file2) .
```

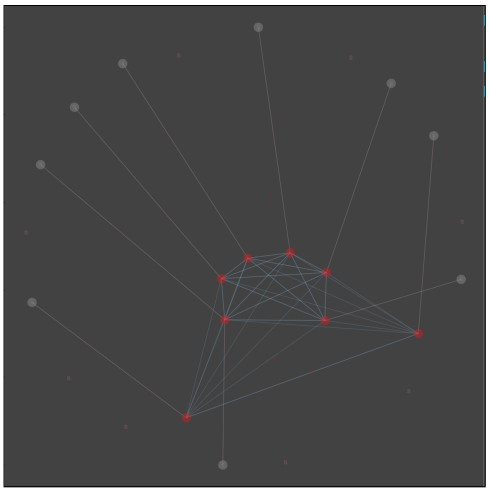

Figure 6.1: Task 4 - before highlighting the relationship.

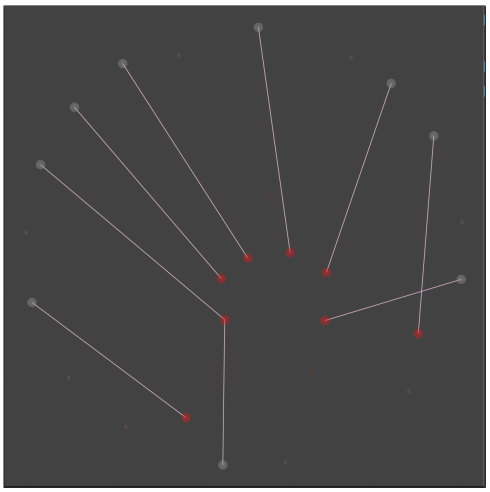

Figure 6.2: Task 4 - after highlighting the relationship.

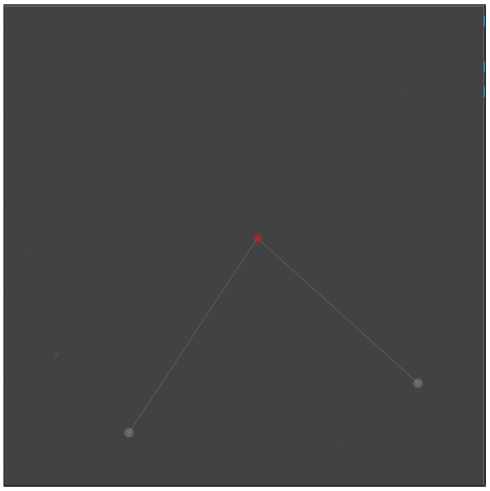

Figure 6.3: Task 4 - solution using alternative query.

Task 5

- **Question**: Create a query for every two different `ExploitFiles` embedding `PayloadFiles` which `connectsTo` the same `Webserver`. Define the relationship `sameActorAs` for these `ExploitFiles`. Which version are you in now? How many relationships were created?
- **Main themes**: Query, materialization, new version.
- **Reason**: To demonstrate another key feature of using semantic technologies to investigate datasets: the ability to perform graph processing (querying nodes, relationships and properties as demonstrated in tasks 1 and 2, for instance) and set processing[1] at the same time. In this case, it will be necessary to search for and compare every two graph patterns in the dataset according to some specific criteria.
- **How to solve it**: The first step is very similar to tasks 1 and 2: creating a SPARQL query representing the graph pattern to be searched for. This is done by translating the query statement to the following triples: `?exploitfile1 vtinv#embeds` *`?payloadfile1`* and *`?payloadfile1`* `vtinv#connectsTo ?webserver1`. Then, it is necessary to add the second pattern, equivalent to the first one, but using different variables: `?exploitfile2 vtinv#embeds` *`?payloadfile2`* and *`?payloadfile2`* `vtinv#connectsTo ?webserver2`. Finally, add the criteria to compare every two patterns in the knowledge base, by appending the following *filter* triples: `FILTER (?exploitfile1 != ?exploitfile2)` and `FILTER (?webserver1 = ?webserver2)`. Figure 6.4 represents the patterns that will be created from the triples displayed in Listing 6.2.

[1]Set processing is a SQL technique used to process groups, or sets of rows, at one time, rather than processing each row individually. *Source:*: `https://docs.oracle.com/cd/E80738_01/pt854pbh2/eng/pt/tape/task_UsingSetProcessing-07720a.html`, *accessed in* 21/11/2017

The second part of the task is very similar to Task 3: the only difference is that the participants must select two target variables this time(`?exploitfile1` and `?exploitfile2`) instead of one, as they will be creating a relationship. Figure 6.5 illustrates one of the 14 to-be-created relationships (in light brown), each connecting two `ExploitFiles` (in dark brown).

- **Expected outcome**: Being able to define these two patterns and the filters using the query builder would already demonstrate an understanding in both the technology and the prototype. However, the participant must pay attention to selecting the same variable (in italics) in order to link the triples in each pattern. Otherwise, there would be four graphs instead of two. Finally, it is necessary to create a relationship between every two patterns returned.

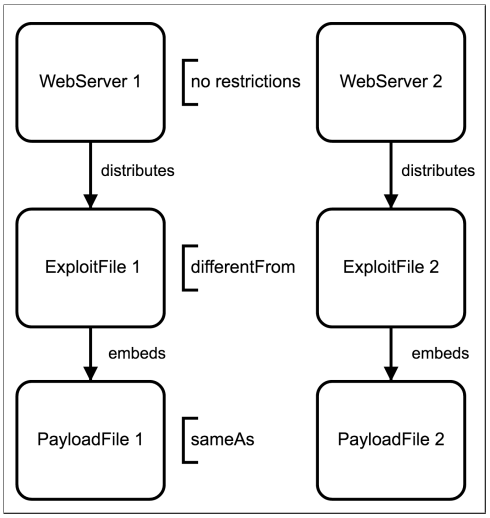

Figure 6.4: Querying two graph patterns.

Listing 6.2: Query from Task 5.

```
?exploitfile1 vtinv#embeds ?payloadfile1 .
?payloadfile1 vtinv#connectsTo ?webserver1 .
?exploitfile2 vtinv#embeds ?payloadfile2 .
?payloadfile2 vtinv#connectsTo ?webserver2 .
FILTER (?webserver1 = ?webserver2) .
FILTER (?exploitfile1 != ?exploitfile2) .
```

Task 6

- **Question**: Among all the ExploitFiles holding the relationship sameActorAs with another ExploitFile, how many are from cluster HT?
- **Main themes**: Graph exploration, query, knowledge reuse.
- **Reason**: Similarly to Task 4, the aim of the present task is to demonstrate that the participants could reuse the recently-created knowledge in a new query. In this case, they would be able

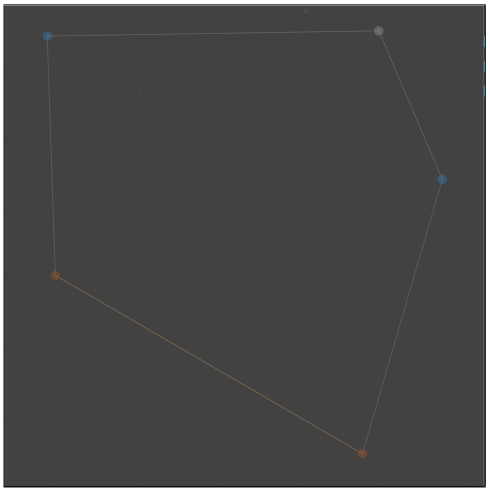

Figure 6.5: One of the user-created relationships `sameActorAs`.

to check the intersection between two pieces of knowledge created by them, helping to assess hypotheses and possibly generating new insights.

- **How to solve it**: Since `HT` is a subclass of `ExploitFile`, the query necessary to complete this task needs just one triple: `?exploitfile1 vtinv#sameActorAs ?ht1`. Notice that both the predicate (`vtinv#sameActorAs`) and the object (`?ht1`)of this triple refer to the relationship and the cluster (class) created by the participant, respectively.

- **Expected outcome**: Running this query would return five `ExploitFiles`, out of which two from the cluster `HT`. Another triple that would also solve this task is `?exploitfile1 vtinv#sameActorAs ?exploitfile2`. In this case, a total of 11 `ExploitFiles` would return, as there are six other `ExploitFiles` (which are not from the cluster `HT`) holding the relationship `vtinv#sameActorAs` among themselves.

Task 7

- **Question**: Add the file: "more_info.ttl" to the investigation. Which version are you now? Are there any duplicate files (i.e. two files with the same md5)? If so, how did you find about them?

- **Main themes**: Integration, query, new version.

- **Reason**: This task demonstrates the capability of integrating a second dataset in the knowledge base, after mapping it to the current ontology (as discussed in Section 5.2.1). However, merging both datasets could result in "duplicate" entities, or items from distinct datasets holding complimentary information about the same "thing", as Table 6.1 illustrates. In order to find them, it is necessary to perform set and graph processing again. Therefore, the goal of this

Table 6.1: Distinct datasets with complimentary information about the same file.

	Dataset 1	Dataset 2
Namespace	ss_report	new_cuckoo
MD5	1a2b...3c	1a2b...3c
distributedBy	evil.com	-
embeds	-	4d5e...6f

task is two-fold: demonstrate the merging capability, and check whether users have understood this key feature of the prototype.

- **How to solve it**: The first step consists simply in inserting the file name containing the new dataset into the appropriate filed and clicking "Add file". Once the new version is generated, the participant should create the query displayed in Listing 6.3, which will search for every two `Files` with matching MD5 hashes.
- **Expected outcome**: It is expected that the participant will notice that the number of `Files` and `Webservers` both in the graph and in the counts panel have increased after integrating the new dataset. Also, that the participant should acknowledge the existence of duplicates by either running the query or sorting the table `File` by the column `MD5`.

Listing 6.3: Triples necessary to search for files with the same MD5 hash value.

```
?file1 vtinv#md5 ?md51 .
?file2 vtinv#md5 ?md52 .
FILTER (?md51 = ?md52) .
FILTER (?file1 != ?file2) .
```

Task 8

- **Question**: Merge all the "duplicate" files based on their MD5 hash value. Which version are you in now? How many ExploitFiles are there now?
- **Main theme**: Enrichment, new version.
- **Reason**: This task demonstrates another important feature: enriching datasets. If any "duplicate" files are identified from the previous task, the participants could merge them. Even though this task could be automated every time data is added to the knowledge base, one of the experts interviewed beforehand had alerted that because it is seldom necessary to assess datasets with low confidentiality, the user might not want to automatically merge all the "duplicate" entities.
- **How to solve it**: In the merging section, the participant should select the target class (`File`) and property (`MD5`), and click on the button"merge". Notice that any pair class/property could be chosen: this will depend on the domain being investigated and the hypotheses being tested by the investigator.

- **Expected outcome**: In this new version (15), the participants should realize that the number of `Files` have decreased to 34, but not the number of `Webservers`. Despite there might be some duplicate entities of the latter, the merging rule only targeted the former. In addition, they should notice that merged entities are now represented as bigger circles (to facilitate graph exploration), and that they comprise properties originated from distinct datasets (indicated by different letters inside the circle, as described in Chapter 4.

Task 9

- **Question**: Apply again the rules for cluster `HT` and relationship `vtinv#sameActor`. What are the new figures for them?
- **Main themes**: Knowledge reuse, new version.
- **Reason**: Once more data is loaded and merged into the knowledge base, it is possible to apply any of the previous bespoke knowledge rules, to check if they will match any of the added information. If so, new members from cluster (class) `HT` and relationship `vtinv#sameActorAs` would materialise.
- **How to solve it**: Rules applied to the knowledge base are automatically saved into the prototype. Therefore, the participants only need to select which ones will be applied (from the "saved rules" dropdown box) and click "Apply rule to version".
- **Expected outcome**: The new version will have updated entities counts for both materialisations executed: members from cluster `HT` will increase from 12 to 28, and the number of `vtinv#sameActorAs` relationships will increase from 14 to 34.

Task 10

- **Question**: Among all the `ExploitFiles` holding the relationship `vtinv#sameActorAs` with another `ExploitFile`, how many are from cluster `HT` now?
- **Main themes**: Graph exploration, query.
- **Reason**: The question from this task is the same from Task 6. The aim is to verify whether new insights will arise from applying a previous hypothesis to a dataset enriched with new information.
- **How to solve it**: The participants only need to run the same query from Task 6 again, either by recreating or loading it (in the case they have saved the query back in Task 6).
- **Expected outcome**: After running the query again, the participant should notice that the number of `ExploitFiles` from cluster `HT` which hold a relationship `vtinv#sameActorAs` with another `ExploitFile` have increased from two to eleven.

6.1.4 Final questionnaire

The main objective of the final questionnaire was to capture the impressions of the participants when operating the prototype: whether they had any technical (usability bugs) or conceptual (lack of understanding of the technology) issues. In addition, how well they would value the proposed features when investigating datasets, as they have tried all of them during the experiment. In order to avoid question bias, the questionnaire was based on open, discursive questions. There were seven of such questions, and three multiple-choice ones.

Discursive questions

1. *Describe in your own words what you think of SemInv*: This question aims to assessing the fitness of the underlying technology to the problem, which is one of the reasons it was explained in detail beforehand.

2. *What was your overall experience with SemInv like and why?*: This question deals with the experiment itself: how usable the prototype is in performing the tasks, and whether the participant felt the prototype could address them satisfactorily or not.

3. *What can SemInv improve?*: Considered together with the analysis of the tasks in which the participants had most problems, the replies to this question will inform improvements for the next version of the prototype.

4. *What should SemInv keep the same?*: This goal of this question is to check whether any of the implemented features will be mentioned here, before asking for them explicitly in questions nine and ten.

5. *Do you think SemInv could be useful for investigating other datasets? If so, which?*: The objective of this question is twofold: in addition to identifying potential real-world problems that the prototype could be applied to, it also aims to getting inspired by different domains for future improvements of the prototype.

6. *Did you find building queries challenging? If so, why?*: This question, analysed together with the performance of the users in most tasks, will compare the level of difficulty between technical and non-technical participants in creating queries, and will inform for future developments of the query builder.

7. *Is there anything in this experience or experiment setup you found particularly problematic and would like to highlight (introductory videos, tasks, etc)? If so, what?*: fuThis question aims exclusively to improving the experiment procedures before the main validation with the malware investigation experts.

Multiple-choice questions

Question 8 presented a table listing potential future improvements for the prototype, and asked the participant to value each of them, from the most to the least relevant. The list comprises five suggestions: *decreasing the learning curve, improving graph exploration, defining ontologies for different datasets, exchanging rules and queries with other users and scaling to bigger datasets.* There is also an open choice field, in which the participant could add any not-listed improvement.

Finally, questions 9 and 10 asked the participant, respectively, how *important* each of the main implemented features are, and how *easy* was to operate them via the prototype. These are the same features proposed in Table 1.1.

- Creating bespoke queries;
- Materialising (creating) clusters, relationships and tags;
- Merging items from distinct datasets;
- Exploring the dataset using a graph;
- Working with different versions of the dataset;
- Reusing queries and rules.

6.2 Results and analysis

6.2.1 Initial questionnaire

Demographics

Out of the ten participants who replied to the invitation email, one could not attend the agreed time slot. All nine attendees (five male and four female) are currently working in academia: one has been involved between 10 and 15 years, three have been involved between 5 and 10 years, and the last five have been involved in academia for less than five years. In terms of age and gender, there are:

- six participants with age between 30 and 39 years;
- two participants with age between 18 and 29 years;
- one participant with age between 40 and 49.

Background

From now on, the participants will be referred to with an "@" followed by a specific letter, in the case it is necessary to identify and relate their replies and comments.

According to question 6, all participants have a background (work or education) in cybersecurity, which was a criteria for the recruitment. As a sub-question, five participants (@c, @j,@w, @h and @q) expressed their primary focus was on the technical aspects of cybersecurity, while the other four of them (@m, @f, @a, @p) stated they were more involved with the social aspects. Therefore, they will be divided into the technical and non-technical group in the analysis of the experiment.

Also, none had previous knowledge about semantic web-technologies, and only one(@j) claimed to have a background in malware investigation or threat intelligence.

Querying

Regarding the two questions about queries, only six participants (@c, @f, @a, @j, @w, @h) had already written queries on their own. Among them, two (@f and @a) were from the non-technical group.

The second question asked how often did the participants explore structured datasets as part of their work. Six participants replied "rarely". @c and @a replied "often', and @j, the only one with previous background in malware investigation, replied "always".

The replies from each participant to the background and querying questions were compiled in Table 6.2, which aims to aid the discussion and analysis of the rest of this chapter.

Table 6.2: Replies to background and querying questions, per participant.

Replies / Participant	**c**	**j**	**w**	**h**	**q**	**m**	**f**	**a**	**p**
Primary focus on technical aspects	x	x	x	x	x				
Primary focus on social aspects						x	x	x	x
Previous knowledge about malware investigation		x							
Previous knowledge about semantic technologies									
Written SQL before?	x	x	x	x			x	x	
Explore datasets as part of work?[a]	O	A	R	R	R	R	R	O	R

[a] R: *rarely*, O: *often*, A: *always*

Visualisation

There was one question asking the participants to rate how important the below visualisation types are for they work. Figure 6.6 illustrates the replies by members of the technical and non-tecnical

group.

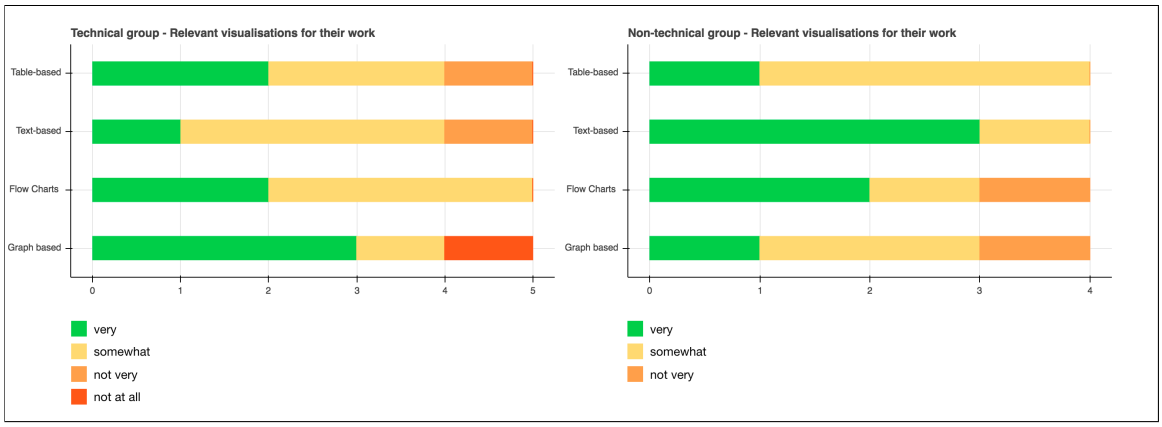

Figure 6.6: Relevant visualisations for the participants's work.

The majority of the participants replied that all of these visualisation types are either "very" or "somewhat" important for their own work (31 out of 36 total replies). According to Table 6.3, participants @m and @w rated "graph-based visualisation" as "not very" and "not at all" important respectively. Thus, their performance on tasks involving graph exploration will be compared to the other participants , to verify any correlation to their low rating to this question.

Table 6.3: Replies to importance of visualisation types, per participant.

Visualisation feature value [a]	**c**	**j**	**w**	**h**	**q**	**m**	**f**	**a**	**p**
Graph based (e.g. showing connectivity or relationships between nodes in a network)	4	4	1	4	3	2	3	4	3
Flow Charts (e.g. representing the flow of data)	3	4	3	4	3	2	3	4	4
Text-based (e.g. using colour coding to highlight text differences)	3	3	2	4	3	4	3	4	4
Table-based (e.g. manipulating information in spreadsheets)	4	2	3	4	3	4	3	3	3

[a] 4: *very*, 3: *somewhat*, 2: *not very*, 1: *not at all*.

Dataset exploration

Regarding the question about useful features for visual exploration of datasets, Figure 6.7 indicates that "Abstraction of data to show the bigger picture", "Detailed data to show fine information" and "Emphasis on pertinent information" were the most voted ones. All of them relate to having different perspectives on the data. In the prototype, this general feature is implemented through the materialisation capability, as they allow the user to refine the graph view using with bespoke clusters, relationships and tags derived from their own knowledge."Scaling to bigger datasets", "Adding your own knowledge to the dataset" and "Tool automation" were the least voted features. In terms of feature relevance by members of both groups, "Scaling to bigger datasets" was the feature with

greater variance, as column “Dif” indicates: participants with a technical background gave much more importance to it.

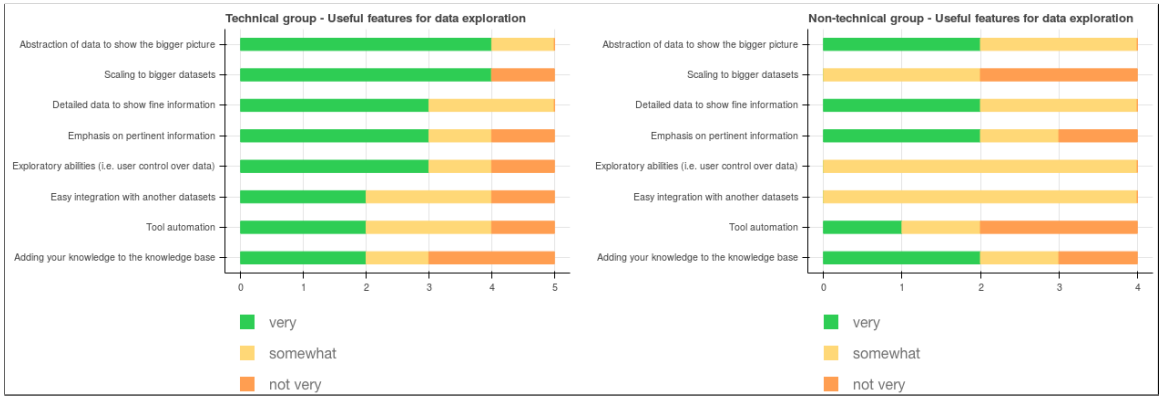

Figure 6.7: Useful features for data exploration.

6.2.2 Tasks

Considering all tasks from all participants (as shown in Table 6.4), there were 57 of them completed without any help, 16 completed with a minor advice, and 15 in which guidance to completion was necessary. In total there were 88 tasks assessed, instead of 90: due to a program crash during execution of Task 6, participant @a could not complete two of them. This crash was caused by an unhandled exception, already listed as a bug, which caused the data necessary to complete tasks 9 and 10 to be lost.

The individual analysis of the tasks below comprises three plots: one displaying the time taken to complete the tasks (in seconds), one illustrating the number of buttons clicked, and one correlating these two metrics. The data points in the scatter plot have different colour codes:

- *Green*: the participant has completed the task by herself;
- *Yellow*: the participant only needed a minor advice to complete it;
- *Red*: the participant needed guidance throughout the task.

Task 1

All participants have completed Task 1 without any advice or guidance. They all used four clicks, which was the minimum to create the query that would return the correct graph, indicating that building and running the query was quite straightforward. The differences in timing, though, are due to some users taking longer than others to find the number of resulting nodes returned by the query: @q took longer to identify that the answer was in the *counter box* , which @h found it rather quickly.

Table 6.4: Task assessment.

Task	Completed alone	Advice given	Guidance needed
1	9	0	0
2	8	0	1
3	8	1	0
4	5	1	3
5	3	3	3
6	6	2	1
7	3	2	4
8	9	0	0
9	3	5	0
10	4	2	2
Total	57	16	15

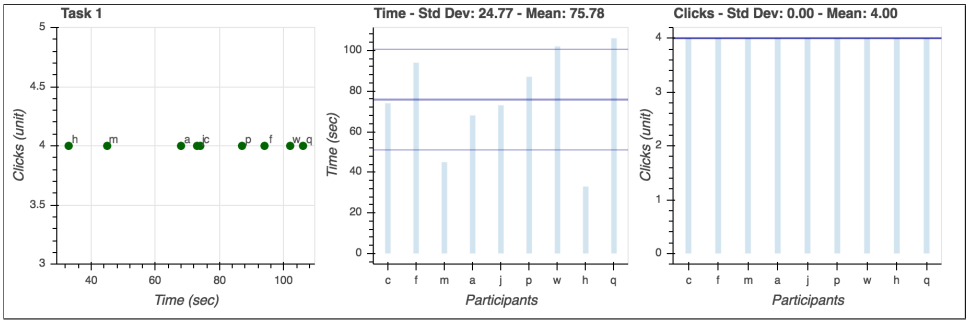

Figure 6.8: Clicks and timing for Task 1.

Task 2

Most participants completed Task 2 without any help. The exception was @w, which was selecting new variables for each triple added to the query, what would cause it to return the wrong results. The high values of time taken and buttons clicked are because this participant had to create a new query from scratch. Participants @q, @c, @m, @f and @j correctly completed the tasks with the minimum required clicks.

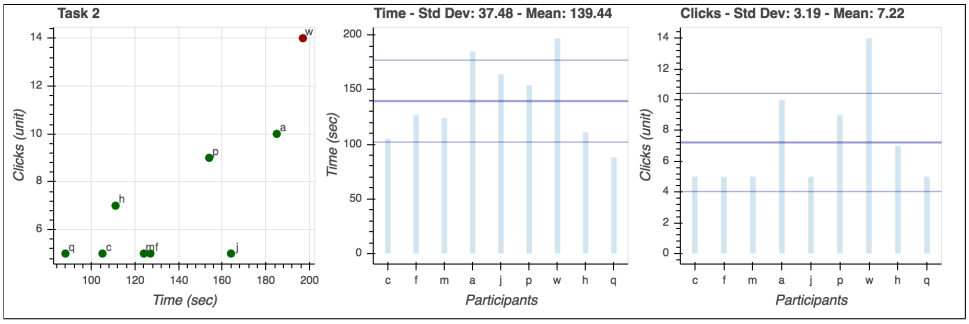

Figure 6.9: Clicks and timing for Task 2.

Task 3

Task 3 also yielded good results. There was only one minor advice given: although @w correctly converted the results of the query into a cluster, this participant had problems in identifying that the recently created cluster would now show in the *counter box* as a new class. On the other hand, participant @p took a longer time exploring the interface, but realized the answer was in the *counter box* by herself.

Finally, the longest time observed was from participant @j. She took longer than all others because she spent some time reviewing the query definition before clicking on "Create cluster".

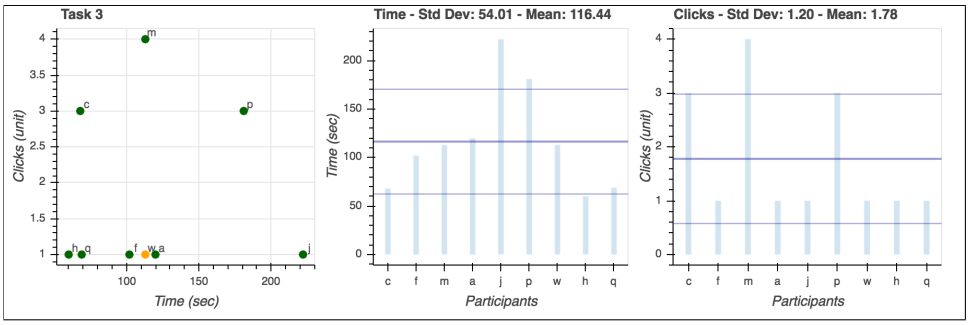

Figure 6.10: Clicks and timing for Task 3.

Task 4

Task 4 was more challenging than the previous three ones: as we can see in Figure 6.11, four participants needed advice or guidance to complete it.

Although taking much longer than the others, participant @c was determined to complete this task on her own. The high number of clicks was due to an intense use of the query builder. She was trying to search for `Files` first, and then filter the ones from cluster `HT`. For that, she tried to rebuild the same query that created cluster `HT`. At this point, the advice "Maybe you could reuse the cluster" was given. But even then, she did not realize that the class `HT` was available in the current version of the dataset. Eventually, she figured out the answer manually, by checking every single node in the resulting graph.

Participants @p, @m and @a did understand they could use the recently created cluster in a new query, so they correctly finished the first step of the task: building the query `?webserver1 #distributes ?ht1`. However, there were problems on the second step: how to tell if any of that `files` were being distributed by two `webservers`. @p and @m were trying to accomplish that by further adding triples, but they failed. @a had no idea what to do after the query. So, they were pointed to the expected solution: highlighting the relationship `distributes` on the resulting graph would instantly reveal the answer, as shown in Figure 6.2. Participant @a also mentioned that the

fonts were too small for her, what might have impacted her performance so far. This issue was solved instantly.

Finally, participant @w was the only one who took the alternative route: she created the query displayed in Listing 6.1, obtaining the graph shown in Figure 6.3.

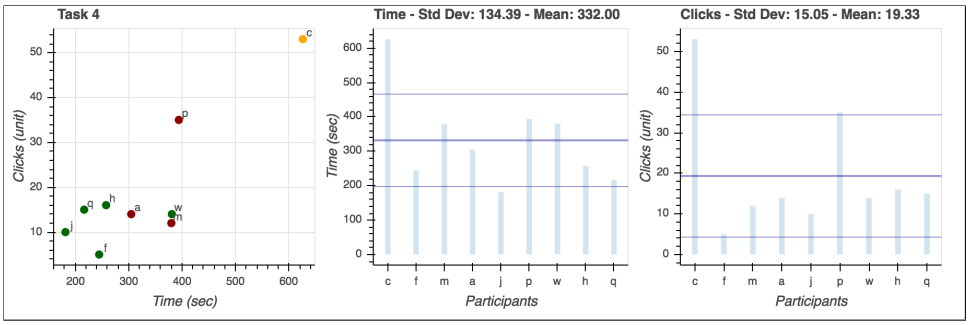

Figure 6.11: Clicks and timing for Task 4.

Task 5

Task 5 was the most difficult one, as it involved both set and graph processing. The abstract idea of comparing every two graph patterns in a dataset could explain that only three participants, all from the technical group, could complete it without any help. The main issue observed was exactly to append the triples that would form the second pattern.

Participants @p, @w and @q confused `ExploitFiles` with `PayloadFiles`, which led to a wrong triple appended to the query (`?payloadfile1 vtinv#embeds ?exploitfile1` instead of the opposite). However, after pointing this out as a minor advice, they could finish the task correctly.

Regarding @m and @f: although they could correctly create the first pattern, they did not understand why it was necessary to define a second one. This is more a conceptual problem than an usability one. Even though a figure similar to Figure 6.4 was explained to all participants during the training phase, it could still be difficult to grasp the concept of set and graph processing in one experiment session.

Finally, participant @a still had trouble in creating the first pattern, as she was worngly defining a new variable for `PayloadFile` on the second triple, instead of reusing the one from the first triple. This led to the wrong first pattern `?exploitfile1 vtinv#embeds` *`?payloadfile1`* `.` *`?payloadfile2`* `vtinv#connectsTo ?webserver1`.

Task 6

The analysis of this task comprised only eight participants, as @a could not complete it: she found a bug during the previous task which caused the program to crash.

Two participants (@f and @q) did not remember that they could use the recently created relationship in new queries. It took @f 236 seconds and @q 181 seconds until the advice "your recently

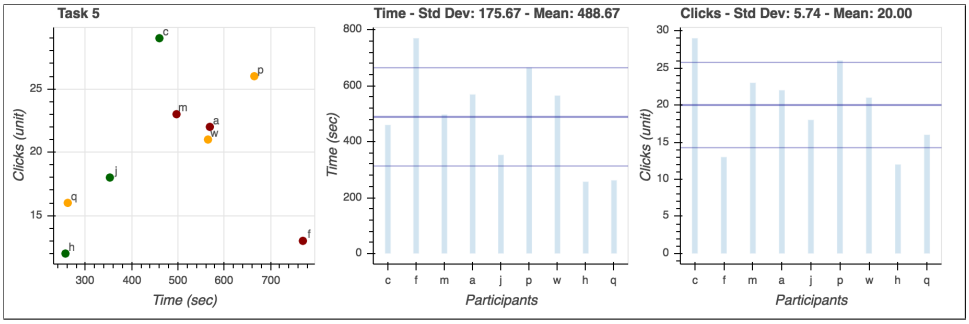

Figure 6.12: Clicks and timing for Task 5.

created relationship and cluster are available to use" was given. From there on, they could solve the task rather quickly.

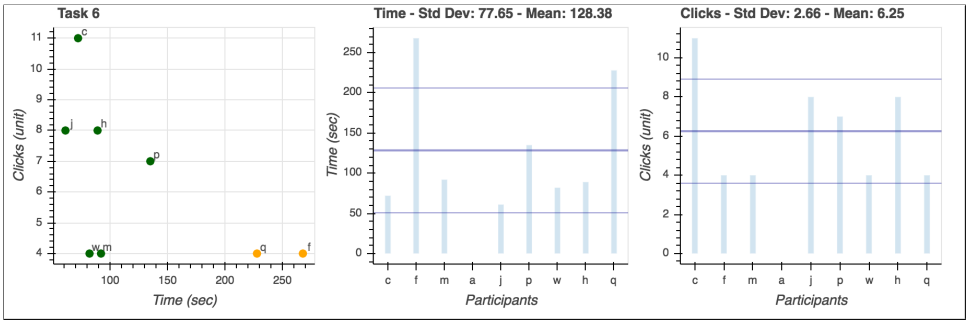

Figure 6.13: Clicks and timing for Task 6.

Task 7

Participants did not perform well in Task 7, with only three of them completing it without any advice or guidance. The objective was to find out if there were any "duplicate" files after loading more data into the knowledge base.

Participants @q and @w just needed a minor advice: that md5 was actually a property, and not a relationship. However, @m, @a, @p and @f have not included a second variable `?file2` in the query, what is necessary to compare every two patterns in the knowledge base. This is the same rationale from Task 5, but applied to a different objective, reinforcing that although participants from the non-technical group could easily create one pattern (as shown by the results from tasks 1 and 2), they did not understand the concept of defining two patterns (as they also had problems in Task 5). Finally, the low number of clicks number from @q is due to the fact that she focused in exploring the content of drop-down boxes (which was not captured for any tasks for technical reasons), instead of clicking other buttons. Once the advice was given, it was quite straightforward for her to finish it.

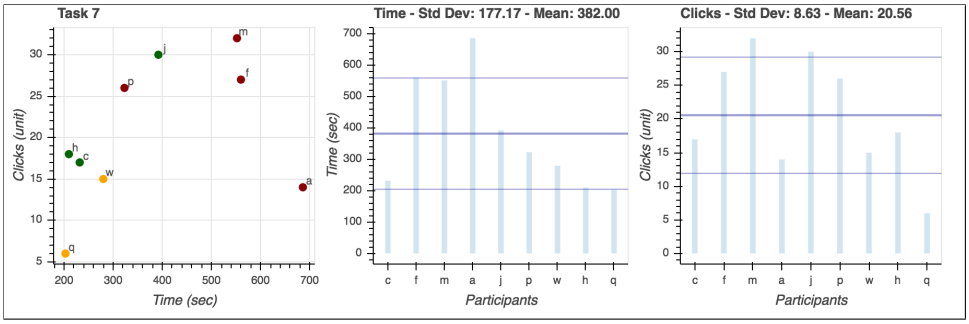

Figure 6.14: Clicks and timing for Task 7.

Task 8

Task 8 was was a follow-up from Task 7, and it was relatively simple, involving only one action to take (merge). The discrepancy in times reflect how long did the participants took either to find the button and drop-down boxes on the screen or to remember that merging could be done automatically. All the participants noticed the decrease in the number of `Files`, and were told about the difference on the size of the merged nodes and also that they now comprised properties from distinct datasets.

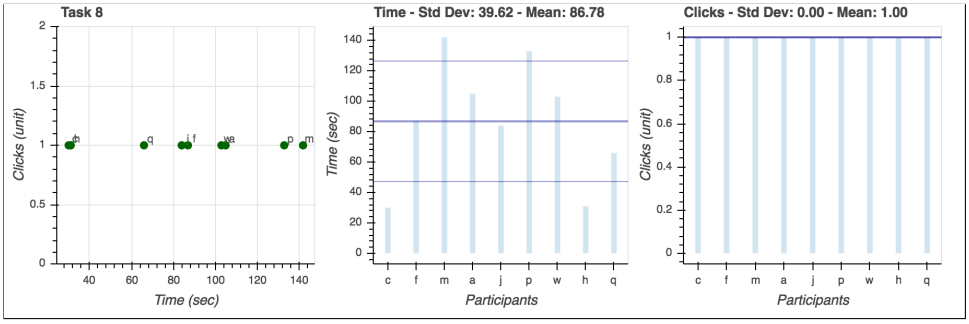

Figure 6.15: Clicks and timing for Task 8.

Task 9

Participant @a could not finish this tasks as the necessary rules to complete it were lost due to the program crash happened during Task 6. Participants @q, @c, @w, @m and @j did not remember that the rules for defining the cluster `HT` and the relationship `sameActorAs` were automatically created and saved from the queries in tasks 3 and 5. Some of them were trying to recreate the cluster `HT` from scratch, and others were trying to build new queries using `HT`, but not using `sameActorAs`. Maybe this feature was not properly addressed during the training phase, as more than half of the participants had issues in a task considered easy.

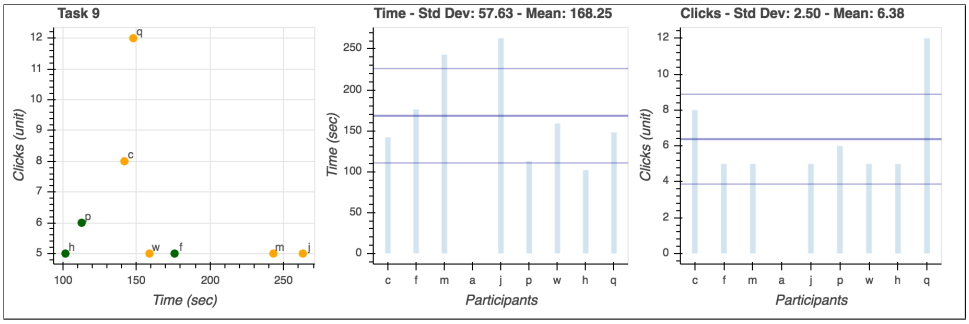

Figure 6.16: Clicks and timing for Task 9.

Task 10

Participant @a could not complete Task 10 as well, as it is a follow-up from Task 9. Participants @p and @m did the query `?exploitFile1 #sameActorAs ?exploitFile2`, but then were looking for a property named HT, when it is actually a class. There were two options: count `HT` members from the resulting graph (as they are a subclass of `ExploitFile`) or adding a different triple: `?ht1 #sameActorAs ?exploitFile2`.

Participant @f got a bit confused on whether to choose `?ExploitFile` or `?HT` as the subject and object of the triple, when both possibilities would lead to the right answer (with the former, though, it would be necessary to count items on the graph). Finally, @w simply replied the total of `HT` members. When advised about it, she was quick to create the query that would result in the right answer.

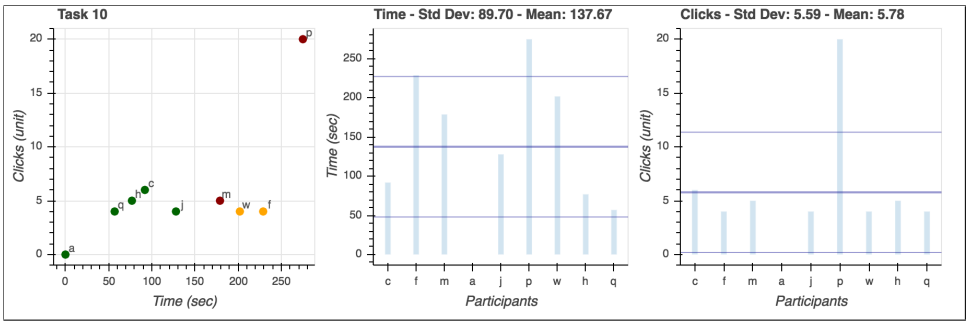

Figure 6.17: Clicks and timing for Task 10.

Analysis per participant

Figure 6.18 illustrates how many issues each participant had in completing the tasks. The green color indicates the number of tasks completed without any help; yellow indicates how many tasks were completed with only a minor advice; and red shows how many were completed with guidance.

It is possible to notice that participant @h completed all the tasks without any help. Participants @j, @c and @q also performed well, with one, two and four minor advices given respectively. None of

these four participants needed to be guided through any tasks, and they all belong to the technical group. The fifth participant of this group is @w, who needed to be guided through Task 2. This was considered an easy task: Figure 6.9 shows that only @w had problems in it. Going back to the analysis of this task and the others in which @w had issues with, it seems that this participant had more problems with the interface rather than with the technology. Even though there was an increased number of advices given to this participant, only the first one was query related. This confirms that the important concept of query-building was properly understood by members of the technical group.

The other four participants needed guidance more frequently, indicating that the prototype would need improvements if it is to be used by non-technical people. The next section presents the results of the final questionnaire, listing some of these improvements.

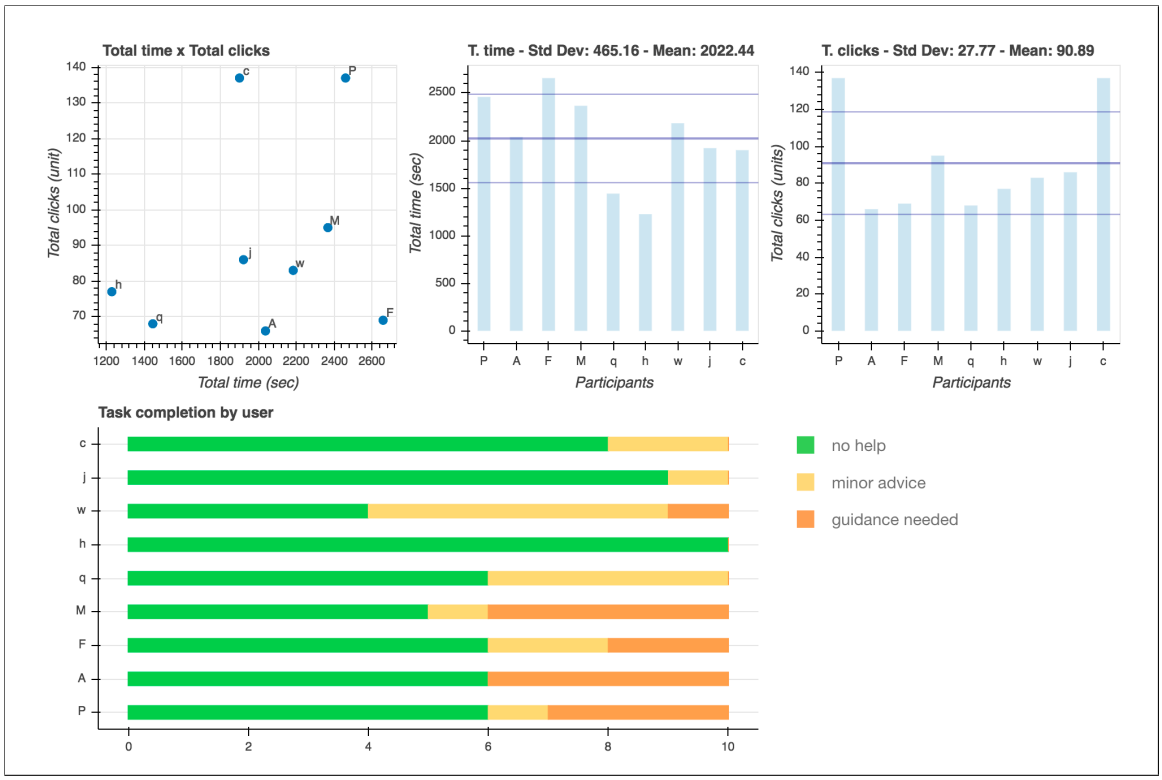

Figure 6.18: Task completion per user.

6.2.3 Final questionnaire

For the final questionnaire assessment, a thematic analysis [73] of all the discursive questions enumerated in 6.1.4 was performed. Due to the small number of respondents for each question, it was not necessary to define codes, but only to identify the main themes.

Questions 1 and 2

Even though questions 1 and 2 intended to explore different aspects of the prototype (the former addressing the concept of the prototype, and the latter addressing the actual experience with it), it seems that the replies to both of them addressed only the experience with it (e.g. mentions of learning curve), possibly due to imprecise wording. Therefore they were analysed together, and four main themes emerged from the responses:

1. **Good/Useful**: All nine participants described either the prototype or the experience with it positively, using words such as "good", "useful", "effective", "nice", "interesting", and "powerful".

2. **Learning curve**: five participants (@f, @c, @q, @p, @j) considered there was a learning curve to operating the prototype. Two of them stated this terms explicitly, while two others mentioned "...as long as the analyst is familiar with the language of the relationships etc" and "...once you understand the basic concepts of building queries and relations.".

3. **Easy to use**: five participants considered the prototype "easy to use", four of whom mentioned the learning curve as well.

4. **Not user friendly**: the responses from three participants were grouped into the theme "Not user friendly". Interestingly enough, none of them mentioned "learning curve", what might indicate that both terms refer to the same concept.

Table 6.5 presents the themes per participant response. Even though all nine participants thought the prototype was "Good/Useful", eight participants also mentioned either "Learning curve" or "Not user friendly, confirming that there is, indeed, a learning process associated with the prototype. Participant @w was the only one from the technical to mention "Not user friendly". The fact that "graph-based" visualisations are not relevant for her work duties (according to Table 6.3 could explain this divergence from other participants in the same group. Similarly, participant @m from the non-technical group did not perform well in the tasks.

Table 6.5: Experience with the prototype per participant.

	@c	**@j**	**@w**	**@h**	**@q**	**@m**	**@f**	**@a**	**@p**
Good/Useful	x	x	x	x	x	x	x	x	x
Learning curve	x	x			x		x		x
Easy to use	x	x		x	x				x
Not user friendly			x			x		x	

Finally, the two most significative responses from participants from the non-technical and technical group are, respectively :

- "Great concept but not super user-friendly yet. I like the power it gives me to interact with data but am not good at driving the interface."

- "I thought *seminv* was very detailed and interesting to use. At first I felt like indeed I needed practice to learn all the tool features, but after some time this became much easier and fairly natural to use."

Question 3: What can SemInv improve?

Three themes emerged from the analysis of the responses about potential improvements for *seminv*: *User Experience (UX), font size/colour and queries*. Although the second could be grouped within the first one, there were explicit mentions to the former, making it deserve a theme of its own.

1. **UX**: five participants mentioned that the User Experience could be improved, whether by moving, hiding and explaining different modules/actions, or enabling different versions comparison. Some expressions used were "...more user friendly" and "become more intuitive".

2. **Font size / colour**: three participants found the font size too small or the colours of the graph nodes and edges confusing.

3. **Queries**: four participants reported issues related to queries: three of them missed an "editing" option in order to avoid recreating all the query in the case a mistake in one triple is made, and one mentioned the ordering of the queries.

Table 6.6: Necessary improvements per participant.

	@c	**@j**	**@w**	**@h**	**@q**	**@m**	**@f**	**@a**	**@p**
UX			x			x	x	x	x
Font size/colour	x				x			x	
Queries	x	x	x	x					

Out of the five participants who mentioned UX, four are from the non-technical group. On the other hand, all participants who thought the query builder could be improved are from the technical group, clearly indicating distinct concerns between both groups. Due to being one of the most voted improvements and also a fundamental feature of the prototype, revamping the query builder will be prioritized before the main validation. Besides, more intuitive query building could definitely improve UX.

Question 4: What should SemInv keep the same?

This question had many different replies, making it hard to categorize them. There were two mentions to *query language* and another two for *hovering/graph*. Moreover, there were single mentions to: "reapplying queries and rules", "adding more data", "simplicity", "functionality" and "amount of choice"

Finally, three participants mentioned "everything else" apart from their own answers to question 3 (which referred to improvements on the query builder and the font size of the titles). Interestingly enough, these three participants were the among the ones with the better overall performance.

Question 5: Do you think Seminv could be useful for investigating other datasets? If so, which?

Six participants responded positively to this question; two replied "Probably" and one "Not sure". However, very few added which would be the potential datasets: one mentioned "...statistical analysis such as data in social science.", and another, "...investigate malicious users online. Social networks and how they are linked together.".

Question 6: Did you find building queries challenging? If so, why?

Six participants replied "No" to this question: all the five ones from the technical group plus @p, who commented "It is quite straightforward as long as you become familiar with the commands and language.", possibly indicating that the underlying concepts were properly understood by her. Among the five participants from the technical group, two of them found the ordering of the triples confusing and one mentioned the "reset issue" of the query builder. Finally, the three participants who replied "Yes" to this question are from the non-technical. According to their answers, they had problems understanding either what the triples *meant* or their role in forming the queries.

Question 7: Is there anything in this experience or experiment setup you found particularly problematic and would like to highlight (introductory videos, tasks, etc)? If so, what?

Four participants replied "Yes" to this question: one had problems with the wording of the tasks and three others would appreciate a better training to the tool and its underlying technology before the experiment. The most significative reply to this question was "Either simplify the UI to intuitive commands, or spend more time explaining how the database works and why you need to use commands the way you do (property vs. relationships. Adding to the query. Query vs cluster etc)".

Question 8: Please rate the most important future features of Seminv

This question asked the participant to rate how important she thinks some potential future features of the prototype are, from most important (1) to fifth most important(5). Table 6.7 shows that "Defining ontologies for different datasets" and "Improving graph exploration" were the rated features for the non-technical and technical group, respectively. The averages (At, As) and the standard deviation (St, Ss) for the ratings of both groups were calculated as well. The results show that both groups value the features differently:

- For the participants with technical background "Improving graph exploration" was the best rated feature. This might indicate that they perceived it as an important but underdeveloped

feature. Future developments of the prototype will definitely address it, as the main purpose of the current version was to check whether specific semantic technologies could be mapped to common investigation needs;

- Participants from the non-technical group considered "Defining ontologies for different datasets" more important, possibly indicating a will to use this prototype in more familiar domains. Finally, "Decreasing the learning curve" was the second most-valued feature for the latter, what is reflected in the replies from questions 3 and 10.

Table 6.7: Ratings for potential future features.

Future feature	c	j	w	h[b]	q	**At**	**St**	m	f[b]	a[b]	p	**As**	**Ss**
Improving graph exploration	2	1	2	1	5	**2.2**	**1.6**	5	2	1	3	**2.75**	**1.7**
Scaling to bigger datasets	3	2	3	3	2	**2.6**	**0.5**	4	2	1	5	**3**	**1.8**
Defining ontologies for different datasets	5[a]	4	1	1	3	**2.8**	**1.8**	2	2	1	1	**1.5**	**0.6**
Decreasing the learning curve	1	5	4	3	4	**3.4**	**1.5**	1	1	2	4	**2**	**1.4**
Exchanging rules and queries with other users	5[a]	3	5	3	1	**3.4**	**1.7**	3	2	2	2	**2.25**	**0.5**

[a] indicates that the participant left it blank, thus interpreted as not important and attributed the value "5".
[b] indicates that these participants misunderstood the question, having not rated the features from 1 (most important) to 5 (5th most important).

Question 9: How important do you think the below functionalities are?

The two most important features according to participants with either technical or non-technical background are "Creating your own queries" and "Materialisation of clusters, relationships and tags". This potentially indicates that they did understand and appreciate these two functionalities of the prototype, which were the driving force behind its development.

In addition, it is worth noticing that all features were considered either "very important" (40 replies, in green) or "somewhat important" (14 replies, in yellow), as Figure 6.19 shows. Overall, participants from the technical group tended to consider the features more important than the participants from the non-technical group, what might be explained by a better understanding of the features by the former.

Question 10: How easy to use do you think the below functionalities are?

In terms of usability of these same features, the results were not as positive as their recognized relevance. Figure 6.20 indicates that the number of replies "Very easy to use" (25 total replies, in green) was very similar to the number of replies "Somewhat easy to use" (22 total replies, in yellow). Lastly, there were seven responses "Not very easy to use", in red.

However, analysing both technical and non-technical groups separately, it is clear that participants from the latter found the features much harder to execute than the ones from the former

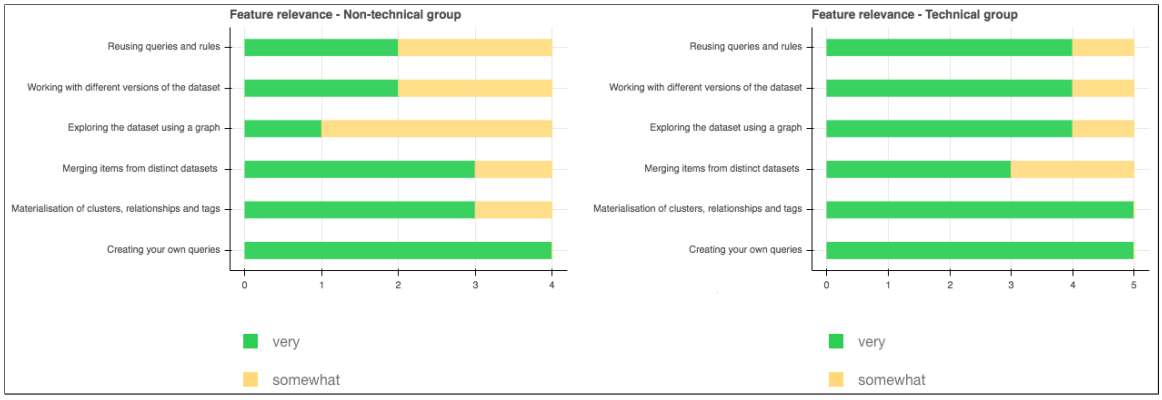

Figure 6.19: Features by relevance.

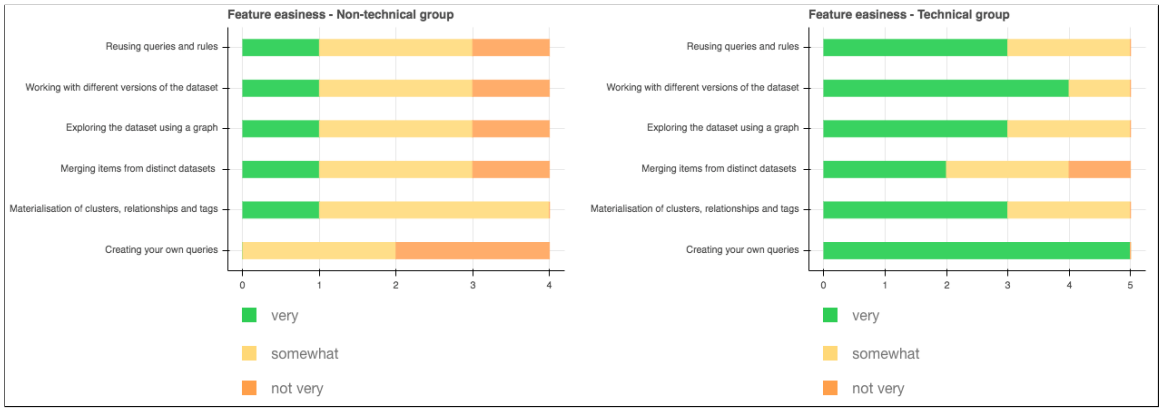

Figure 6.20: Features by easiness.

group, as Figure 6.20 indicates. Interestingly enough, "Creating your own queries" was the feature with the most difference between both groups: it was the easiest from the view point of the members from the technical group, and the hardest according to the members of the non-technical group.

Having all participants from the technical group considering that "Creating your own queries" was the easiest is very encouraging, as this feature is both one of the two most important and also the one who demands more work to be executed. After all, it is the starting point for exploring the dataset and depends on the knowledge of the user. This unanimity is compatible with the replies to question 6, in which no participant from the technical group found the task of building queries challenging.

With the exception of "Exploring the dataset using a graph", all the other features are highly automated already, and demand little effort from the user. One possible explanation that they were not considered as easy as "Creating your own queries" is that they were not utilized as much as this one during the experiment.

Finally, participant "@m" had three replies "Not very easy to use", and participants @p, @a, @f and @q one each. With the exception of the the latter, all the other participants are from

the non-technical group. These results are important and will be addressed in the future, as this prototype is intended to be used in non-technical domains as well. However, as the main validation will only involve participants with technical background, the analysis of the responses to question 10 are considered satisfactorily.

6.3 Conclusion

Completing all the tasks was not trivial to the participants, and there are two factors which might explain that:

- The wide range of actions the user could take by using a non-optimized interface.This is the first version of the GUI of the prototype, which was only intended to validate if specific semantic technologies could be applied to investigating a malware dataset;
- A 1-hour training might not have been enough to train the participants in semantic technologies, and some might have got overwhelmed by learning it in addition to some aspects of malware investigation.

Yet, most participants recognized the power and flexibility they had in defining their own queries and creating new clusters and relationships according to their will, which are the main features of the prototype. In addition, it was nice to notice that six participants did not find building queries challenging, as the query builder is a fundamental part of the prototype. Nonetheless, the query builder still needs improvements in order to better guide the participant in creating and visualising the queries.

For instance, although only four participants complained about the need of restarting the query from scratch if any mistake was made, it was possible to observe that this issue also bothered other participants during the tasks phase. Moreover, two participants from the non-technical group mentioned that it was hard to visualize and understand the raw query before running it. Therefore, the query builder will be revamped before the final validation experiment with the expert users: it will be made simpler (with less controls) and will allow for the user to edit queries.

On the other hand, refactoring the GUI will be left for future work. Even though fonts and colors are easy to change within the prototype, achieving a good interface design (e.g. choosing the right combination of font size, spacing, color palletes and quantity of items displayed in a single screen) is not trivial, and requires both a comprehensive study of visual analytics and multiple interactions with the end user. Thus, the final validation will also serve as the first interaction interview with them.

In terms of experiment preparation the following issues were observed and will be improved for the main validation:

- Some participants wanted to get the query right from the beginning, instead of running one triple, checking the results and appending more triples afterwards. A better explanation of the benefits of gradually appending triples will be reinforced during the training phase;
- There were three participants who made more questions than the others during the training phase. Among them, the two with technical background had the best performance in completing the tasks (@h and @j). Thus, a different approach for training will be applied in the final validation experiment;
- A high reliance on queries to solve all parts of a task rather than exploring the graph was observed. That could have been caused by either the wording of the questions or a lack of graph manipulation during the training, and will be considered during the preparation of the main validation.

Chapter 7

Expert testing

The final piece of this work aims to answer the third research question: whether the suggested approach would be efficient and effective in a real-word investigation setting. Although Chapter 6 also employed real-world data, it was very limited in terms of size, and the participants assessing the prototype were not actual cybercrime investigators. Nonetheless, these factors did not affect the main goal of that assessment: to verify if the user of the prototype could operate and understand the value of its main features (listed in Table 1.1) and to identify any major issues with the general interface design ahead of the main validation discussed in this chapter.

The current experiment will be based on Projeto Tentaculos (Project Tentacles), a project from the Brazilian Federal Police (which will be referred to as *PF* for the rest of this chapter). This permanent project is aimed at investigating online banking fraud leveraging different malware. Having been active for almost ten years, this project fits perfectly as a real-world validation scenario for this thesis. After all, it comprises a mature domain data model and a considerable dataset with hundreds of megabytes of already sanitized data. Moreover, there are around 20 investigators specialized in analysing online banking fraud data (and many other who had training in investigating such crime using the IBM I2 framework) who are used to searching for highly-connected data between different frauds.

7.1 Background of the Tentacles Project - Brazilian Federal Police

Project Tentacles was created inside SRCC (*Serviço de Repressão a Crimes Cibernéticos* - Cybercrime Repression Service), and today counts with a permanent task force to maintain it: *GPA* (*Grupo Permanente de Análise* - Analysis permanent group). Back in 2009, PF started a cooperation with a public bank with the objective to avoid opening a formal criminal investigation for every *notitia criminis* (crime notice) reported. Instead, all the information regarding online banking fraud and card skimming would be inserted previously in a database (*Base Nacional de Fraudes Bancárias e Eletrônicas* - Electronic Banking Fraud National Database). The mission of the *GPA*, in addition

to maintaining this database, is to search it for relationships which could support a robust and comprehensive prosecution (e.g. when the money sums transferred by individuals which were found to be related surpass certain threshold). A formal criminal investigation would only start after enough information is gathered, thus optimizing the scarce investigation resources[74].

The IBM I2 framework[65], comprising IBase and the Analyst Notebook, is the official storage and analysis tool used by *GPA*. Although being a powerful solution, initial inquiring with the head of *GPA* revealed that most investigators do not consider it an user-friendly tool, with some of them using it to simply draw graphs displaying the results of the "manual" investigation they had performed.

Other issues that emerged during this initial briefing were:

- Cases being investigated grow up too much because of successive node expansions, causing the investigator to loose focus when the graph gets too big;
- Police agents in temporary missions at the *GPA* need training in both the investigation domain and in the tool. While the former is indispensable, making the latter easier could improve human resources retention in the group;
- Because of the low adoption of I2 by investigators in different states, the current methodology within the *GPA* is to run automatic scripts searching for relationships between different frauds and then submitting the "partially-ready" investigation results to each state. The main drawback from this approach is that often the results are too big and complex due to the interstate nature of online banking fraud crimes.

Therefore, it is already a consensus at *GPA* that the best approach is to make the investigation procedure easy enough so the investigators from each state would be capable of and persuaded to searching for these relationships between different cases by themselves, however following the central coordination of *GPA*. The semantic features of the prototype fit exactly the need of facilitating the analysis processes by allowing the investigators to:

1. Search and manipulate data according to their hypotheses (by leveraging the querying and materialisation features, respectively);
2. Easily assess data from different sources with their investigation knowledge (by leveraging the integration and merging features, respectively);
3. Share the obtained knowledge with their colleagues (by exchanging bespoke rules).

7.1.1 Dataset

The dataset used in this experiment comprises real fraud data as reported by the partner bank, encompassing the years 2015, 2016 and 2017. Such data is stored in a *mysql* database of around 200 MB, and its schema comprises a table for each relevant item considered in the investigation, as listed in Table 7.1.

Table Name	Table Rows
Agencia (Branch)	5762
Banco (Bank)	19
Boleto (Payment slip)	62775
Conta (Account)	73957
Convenio (Partnership)	4700
Estado (State)	27
Id (Machine Id)	33477
ID_processo (ID_process)	32848
IP (IP)	36766
IP_processo (IP_process)	637004
Local (Address)	43525
Pessoa (Person)	37610
Pessoa_conta (Person_Account)	30517
Pessoa_local (Person_Address)	39090
Processo (Process)	74477
Telefone (Phone)	3763
Telefone_conta (Phone_account)	35862
Transacao (Transaction)	75584
Veiculo (Vehicle)	1606
TOTAL	1229369

Table 7.1: Total rows per table of the original dataset used in the experiment.

This dataset was converted to linked format using the tool *Karma*. The total classes, object and datatype properties obtained are listed in Table 7.2. Tables 7.3, 7.4 and 7.5 show their individualized counts.

After converting the original dataset to a linked format, the next step was loading the resulting turtle file, comprising around 3 million triples, to the prototype. However, the loading time was extremely long and, once finished, the prototype was barely responsive. This issue was completely unexpected, as the prototype performed well during the case study (which dataset comprised merely 240 KB). While that was effective for a proof-of-concept validation, real-world usage of the prototype

	Definitions	Individuals
Classes	21	435325
Object propeties	50	1533170
Datatype properties	30	1083845
Total	101	3052340

Table 7.2: Total counts for classes, object properties, datatype properties and their individuals.

Classes	Totals
Agencia (Branch)	5762
Banco (Bank)	19
Boleto (Payment slip)	62591
Cidade (City)	461
Conta (Account)	7428
ContaBenef (Benefitiary acc.)	11142
ContaVitima (Victim acc.)	34662
Convenio (Partnership)	4736
CredPrePago (Phone top-up)	2188
Estado (State)	27
IDMaq (MachineID)	33306
IP (IP)	36500
Local (Address)	43798
PagBoleto (Payment)	56733
Pessoa (Person)	37576
ProBan (Bank case)	74561
Telefone (Phone)	3763
Transacao (Transaction)	75780
Transferencia (Money transfer)	16859
Veiculo (Vehicle)	3212
Total	435325

Table 7.3: Total counts for classes.

definitely demanded better scaling.

Thus, a complete refactoring of the code of the prototype was necessary before planning and preparing the assessment with the expert users. More details about the code refactoring can be found in Section 4.3.4.

7.1.2 Understanding the investigative domain

Once the prototype refactoring was finished, the next step was to refine the ontology to be used in the experiment. For that, one investigator was assigned by the head of *GPA* to help with the design of the experiment and also to perform some initial testing with the prototype before inviting the other investigators to try the tasks. For instance, during one of the discussions, it was decided how to model the class `state` within the experiment. There are 27 regional branches of the Brazilian Federal Police, one per state, and investigators from each branch focus on the frauds which happened in their jurisdiction. However, due to some intrinsic characteristics of cyber-enabled crimes (namely the ability of committing offences remotely), it is usual for a criminal organization to operate in multiple states. The relevance of the `state` in every investigation demanded it to be modelled as a class, and object properties connecting it directly to both `person` and `account` classes were created as well (in the Tentacles schema from the *mysql* dump, these two latter are not related to state directly, but to `location` and `bank branch`, respectively).

Object property	Total
agencia__banco (branch__bank)	5762
agencia__local (branch__address)	5762
boleto__benef (slip__beneficiary)	13456
boleto__convenio (slip__partnership)	31013
boleto__sacado (slip__victim)	8537
boleto__veiculo (slip__vehicle)	5571
cidade__estado (city__state)	4061
conexao__proban (connection__bank_case)	73375
conta__agencia (account__branch)	74028
conta__estado (account__state)	69581
conta__ip (account__ip)	15805
conta__telefone (account__telephone)	35715
convenio__estado (partnership__state)	4736
cred-pre-pago__telefone (top-up__phone)	2188
ip__provedor (ip__isp)	36500
local__cidade (address__city)	43797
pagBoleto__boleto (payment__slip)	56733
pessoa__conta (person__account)	31138
pessoa__estado (person__state)	35049
pessoa__local (person__address)	39990
pessoa__veiculo (person__vehicle)	1606
telefone__estado (phone__state)	3763
transacao__conta-vitima (transaction__victim-acc)	75780
transacao__proban (transaction__bank_case_	75780
transferencia__conta-benef (transfer__benefitiary-acc)	16859
Total	766585
Total with inverses	1533170

Table 7.4: Total counts for object properties.

Another difference from the Tentacles schema is the creation of the `transaction` class. In the I2 schema, it is represented as a relationship with properties between two `account` nodes. Even though it might cause some confusion during the experiment, that was the simplest way found to represent such relevant information in an ontology.

Finally, other enhancements were discussed and implemented with the goal of making operating the prototype more familiar to the investigators:

- Object property names would follow the pattern DOMAIN_RANGE (e.g. account_state);
- Data property names would follow the pattern DOMAIN_NAME (e.g. account_number);
- All object properties would have their inverse counterpart;
- All nodes would be represented using icons;
- It would be possible to query an item by selecting it in the graph;

Datatype property	Total
agencia_nome (branch_name)	3528
agencia_numero (branch_number)	5762
banco_numero (bank_number)	19
boleto_numero (slip_number)	62591
cidade_nome (city_name)	4059
codigo_Conta (code_account)	74028
conexao_data (connection_date)	102567
conexao_id (connection_id)	69806
conta_numero (account_number)	74028
conta_tipo (account_type)	74028
convenio_cgc (partnership_id)	4735
convenio_nome (partnership_name)	4735
cpf_cnpj (social_security_number)	37574
estado_nome (state_name)	27
estado_sigla (state_abr)	27
local_bairro (address_neighbourhood)	9238
local_cep (address_postcode)	42749
local_logradrouro (address_street)	43796
pessoa_nome (person_name)	37573
pessoa_tipo (person_type)	37575
proBan_data (bank_case_date)	74557
proBan_numero (bank_case_number)	74561
tel_conta_data (phone_account_date)	5719
telefone_numero (phone_number)	3763
transacao_data (transaction_date)	80156
transacao_tipo (transaction_type)	75780
transacao_valor (transaction_value)	77968
veiculo_ano (vehicle_year)	1365
veiculo_placa (vehicle_plate)	166
veiculo_renavam (vehicle_id)	1365
TOTAL	1083845

Table 7.5: Total counts for datatype properties.

- It would be possible to specify the cardinality of object properties (i.e. allow the user to select the minimum number of object properties of the same type sharing a unique domain);
- Index search for datatype property literals would be implemented.

7.2 Method overview

Initially, the idea was to set up a new investigation, based on selected previous cases, and ask the investigators to reproduce the tasks they would normally perform using the prototype. In the end, their experience would be assessed via a semi-structured interview. Despite being an ideal scenario, doing the experiment this way could risk having the investigators using only the query capabilities of the prototype, as they might not be used to materialise their own knowledge into the investigation, thus not assessing one of the two main features of the prototype.

Therefore, the final validation comprised a set of tasks which allowed the participants to fully assess the capabilities of the prototype (similarly to what was done in Chapter 6). This time, however, the tasks were inspired by common investigation activities of the participants (e.g. searching for accounts receiving multiple transfers). That was the reason why the previous informal discussions with one of the investigators (as detailed in Section 7.1.2) was necessary. Finally, the semi-structured interview after completing the tasks was designed to capture if the participants agree that the semantic features assessed by the tasks could indeed make their routine investigations more efficient and effective.

There was only one requirement for being invited to the experiment: having worked for Project Tentacles for more than one year. There were in total nine eligible investigators who were in Brasilia in the period of the experiment: one was indicated to help shaping the experiment, and eight were recruited to run it. This a more homogeneous set of participants than the ones from Chapter 6, as all of the former have proven experience in investigating the subject of the experiment.

The research methods used were basically the same ones described in Section 6.1 (collection of users opinions, experiment tasks, logging and the think aloud protocol), with the exception of the final questionnaire, which was divided into a semi-structured interview and a questionnaire. Semi-structured interviews allow the researcher to, from a initial set of questions, seek clarification and explore paths that might emerge from the replies of the interviewee which were not considered initially. The questionnaire is thus considered an interview guide, providing the interviewee with "...a great deal of leeway in how to reply"[75]. The reason for adopting such technique was two fold:

- to obtain more complete answers to the discursive questions (if comparing to the responses gathered from the questionnaire described in Chapter 6);
- as opposed to Chapter 6, the participants of the current experiment are experts in the scenario being explored (online banking fraud). Keeping the evaluation using only closed questions, with no discussion, could lead to missing relevant insights.

During the experiment, the interaction of the participants with the prototype was captured (buttons clicked and queries created), as well as the timing of each task. Finally, no personal identifiable information was collected and the audio of the experiment was only recorded to dispel potential doubts arising in the post analysis of the tasks. The methodology for this experiment was approved by the University of Oxford Computer Science Department Research Ethics Committee, reference CS_C1A_18_030.

7.2.1 Technology and prototype demonstration

During this phase, the participants were gathered in a room foro for a quick seminar about semantic technologies. The presentation comprised: uses of semantic technologies on the internet today

(Google search using the knowledge graph); the "triple data model" used to define the classes, object properties, datatype properties and individuals within a knowledge base and also to perform SPARQL queries; and a demonstration of all the prototype features. Differently from Chapter 6, such clarification was given ahead of the individual sessions to avoid extending them too much, what could make the participants tired during the tasks phase or eager to finish the experiment at once.

7.2.2 Initial questionnaire

Upon the arrival on their scheduled session, the participants were welcomed and given the instructions for the experiment. Next, the participants proceeded to filling the consent form and the initial questionnaire, the latter comprising:

- Four questions about demographics: what are their age, gender, graduation degree and for how long have they been investigating online banking fraud. Having more experience in such investigations or having their graduation degree in a technical area might make it easier for them to understand semantic technologies or operate the prototype;
- Three questions about the investigation tools: if the participants had taken any course on such tools; which tools and databases they normally use; and finally what are the main benefits and limitations of current procedures and tools in their opinion. The goal of these questions is to get an informed response about the level of satisfaction of the participants with the available tool and procedures;
- Two questions related to the technology used in the prototype: if they had ever written a SQL-query before (as the query builder is a fundamental part of the prototype and SPARQL queries are somewhat related to SQL ones), and if they have ever heard about or used semantic technologies (what could explain different performances during the experiment);
- Four questions regarding their daily practice and opinion about investigation procedures: if they usually take notes when conducting an investigation and what kind of notes would those be; how important it is to search for information regarding past cases when investigating new ones; if it is important to share the specific knowledge acquired in a case with fellow investigators; and, if so, how does that information-sharing happen? The goal of this question is to check, in the final questionnaire, if the participants recognize that some semantic features of the prototype could serve those needs.

7.2.3 Training

After filling the initial questionnaire, both the audio recording and screen capture were turned on, and the training phase started. Initially the participants were asked if they had any doubts regarding the technology and prototype presentation which happened days before, and were then let free to

explore the prototype. During it, they had chance to identify the main elements of the interface, to try different actions and to ask any questions. Once the participants felt comfortable and confident with operating the prototype, they could proceed to the tasks phase.

7.2.4 Tasks

The experiment tasks for the expert assessment study were based on the day-by-day investigations conducted within Project Tentacles. There were a total of nine specific tasks exploring the main features of the prototype: querying, materialisation, integration, merging, graph exploration, knowledge reuse and creating new versions.

Task 1

- **Question**:How many beneficiary accounts have received more than 10 transfers?
- **Main theme**: Query (cardinality).
- **Reason**: Explore the cardinality feature of the prototype, which can be relevant to many investigations.
- **How to solve it**: Using only the relationship column of the query builder, append the triple `?contaBenef1 tentaculos#conta__transacao ?transacao1`, having chosen the value "10" in the "Connections" dropdown box and hitting the button "Run query".
- **Expected outcome**: For the participant to identify that, in addition to the graph displayed, the "Counts" control also gets updated with the two totals: items in the graph, and items returned by the query.

Task 2

- **Question**: Who are the holders of such accounts? Which states are they from?
- **Main theme**: Query (append).
- **Reason**: The procedure of appending triples is equivalent to tailoring a query to specific details, which is useful for any data exploration task.
- **How to solve it**: By appending two more triples to the query from the previous task: `?contaBenef1 tentaculos#conta__pessoa ?pessoa1` and `?pessoa1 tentaculos#pessoa__estado ?estado1`.
- **Expected outcome**: Make sure the participants understand that, if they wanted information which is not in the graph yet, a new triple referring to it must be added. Also, that it is necessary to select the variable used in the previous triple to create the pattern to be queried.

Finally, make them notice that as the pattern gets more complex (with more added triples), the number of results can decrease.

Task 3

- **Question**: Create a relationship "scammer_state" connecting these account holders to their home states. How many scammers exist in the state of Minas Gerais?
- **Main theme**: Materialisation (object property), knowledge reuse, query (facets).
- **Reason**: This task aims to demonstrate how the users can materialise important findings obtained via data exploration into the knowledge base. In addition, that this knowledge (in the case, the relationship insight) is ready to be used in new queries.
- **How to solve it**: If the participant completed Task 2 accordingly, it is just a matter of filling the name for the new relationship "scammer_state" (`tentaculos#fraudador__estado`) in the appropriate textbox in the materialisation area and selecting the variables representing people (`?pessoa1`) and state (`?state1`).

  For the second part, the participant needs to create a new query, appending two triples: one using the recently created relationship (`?pessoa1 tentaculos#fraudador__estado ?estado1`) and the other from the properties column, restricting the state by name (`?estado1 tentaculos #estado_nome ''Minas Gerais''`).
- **Expected outcome**: That the participants notice that, after the materialisation took place, the recently created relationship is ready to be used, making it unnecessary to build the whole query again. Also, if they run the query after appending each triple, faceted results for state name will be returned.

Task 4

- **Question**: Regarding the scammers from Minas Gerais, whose accounts have received bank transfers with transaction date later than 01/01/2016?
- **Main theme**: Query(append, restrict by datatype property).
- **Reason**: Restricting results by date is relevant in most investigations. Making it easy to do so using SPARQL queries gives great flexibility if the users wants to restrict more than one individual in the same graph pattern.
- **How to solve it**: The participants need to add the triples `?pessoa1 tentaculos#pessoa__conta ?conta1` and `?conta1 tentaculos#conta__transacao ?transferencia1`, as the data type property mentioned in the question belongs to the object from the latter triple (?transferencia1). Then, it is just a matter of selecting the required value in the date field.

- **Expected outcome**: That the participants will notice that the class comprising the datatype property `tentaculos#transacao_data` is not in the graph yet (`tentaculos#Transferencia`). Thus, they will need to add triples to the query until the graph pattern contains the specific class which the datatype property restriction will be applied to.

Task 5

- **Question**: Create a new class named "suspicious_ips" to group all the IPs this account used to commit the frauds, with the exception of the IP starting with "200".
- **Main theme**: Materialisation (class), graph interaction (click-and-query).
- **Reason**: Giving the participants the capability to easily group items into clusters (by tapping nodes in the graph) could facilitate testing hypothesis as the investigation proceeds.
- **How to solve it**: By selecting all the 7 `tentaculos#IP` instances from the graph which do not start with "200" and clicking "Add selected nodes to the query". From there on, the variables representing each of these IPs will be available to be "grouped" in a new class, being enough to select them, choose a name for the new class, and click "Create cluster".
- **Expected outcome**: For the participant to notice that even though the class `tentaculos#IP` is not present in the query, some instances of it are displayed in the graph already: the ones that have a direct relationship (object property) with the the account being investigated. So, the participant only needs to select those instances straight from the graph to obtain the variables which will be used to create the new class.

Task 6

- **Question**: How many relationships are there connecting people to their bank accounts? What are the total instances of people and bank accounts? Now, add to the investigation new data regarding account holders from a distinct bank (the data is in the file *correntistas_light.ttl*).
- **Main theme**: Data integration, query.
- **Reason**: Apart from consolidating knowledge on how to run queries, demonstrate how easy it is to add new data to the investigation.
- **How to solve it**: Similarly to Task 1, the participant needs to append one triple (`?pessoa1 tentaculos#pessoa__conta ?conta1`), click on "Run query" and check the how many instances of `tentaculos#Pessoa`, `tentaculos#Conta` and `tentaculos#pessoa__` are displayed in the "Counts control". The answer is 5753, 5797 and 5815 respectively. Then, copy the path of the file containing the new data to the respective text box in the prototype, and click on "Add data".

- **Expected outcome**: That the participants notice that, despite the apparent easiness of adding new data to the knowledge base, it is necessary for this data to be represented in a linked format and mapped to the ontology being used in the investigation.

Task 7

- **Question**: What are the new totals for people and bank account? What is the namespace of the new data added? Now merge instances of type person (`tentaculos#Pessoa`) based on their social security ID (`tentaculos#pessoa_cpf`)and also instances of type account (`tentaculos#ContaBenef`) based on the account id (`tentaculos#conta_numero`).

- **Main theme**: Merging.

- **Reason**: Enrichment is important to any data exploration process. Doing so using semantic technologies (in the case, leveraging the object property "owl:sameas") is not difficult, and would bring an interesting result to the user: if there are different individuals in the knowledge base which refer to the same thing (in the case, people and bank account), applying the "merge" operation would enrich both of them with the conjunction of their complimentary object and datatype properties.

- **How to solve it**: First, it is necessary to run the same query from Task 6 and check the new figures in the "Counts control" for the instances of type `tentaculos#Pessoa`, `tentaculos#Conta` and `tentaculos#pessoa__conta`, which are 5795, 5837, 5857. Then, checking the new instances in the table will reveal their namespace. Finally, in the materialisation area there is a command for merging instances. The participant needs to run it twice: once for merging `tentaculos#Pessoa` based on their `tentaculos#pessoa_cpf`, and once for merging `tentaculos#ContaBenef` based on their texttttentaculos#conta_numero. It is only a matter of selecting these values in the appropriate dropdown boxes and clicking in "Merge items".

- **Expected outcome**: That the participant will notice that around 80 individuals were added to the knowledge base, and that they have a different namespace, what indicates that they came from a different dataset.

Task 8

- **Question**: Run again the rule created in Task 3. What is the first name of the new scammers identified in the state of Minas Gerais?

- **Main theme**: Knowledge reuse, versions.

- **Reason**: To make the participant aware that, after adding new data to the knowledge base, the results of the previous materialisations might not be updated any more. In other words,

because the addition of data happened after the materialisation rule was applied, there might be individuals in the new data which would be affected by that rule. Also, to demonstrate how easy it is to reapply rules, also meaning that such rules could be distributed to other investigators.

- **How to solve it**: In the materialisation area, selecting the rule **tentaculos#scammer_state** and clicking "Apply rule".

- **Expected outcome**: That the participant notices that there are more scammers in Minas Gerais now: seven instead of four. And that just happened because of the two previous steps: among the data inserted in Task 6, there was new information about the account holders of some of the accounts returned in Task 3. After merging them in Task 7 and running the same query from task3, it is possible to notice the updated results.

Task 9

- **Question**: Create a property "from_case" with the literal value "5_2018" for all the seven scammers discovered in the last task.

- **Main theme**: Materialisation(datatype property).

- **Reason**: Similarly to creating clusters, demonstrate that creating or updating datatype properties is also easy, but could serve different purposes, as the user can choose any value for the property to be created. One example would be organising the leads of a case, when it is necessary to add information about specific individuals obtained via surveillance (i.e. information which is not in any database).

- **How to solve it**: The participant only needs to select the variable `?pessoa1` (which represents the seven scammers returned in the previous tasks) in the materialisation area, insert the new name and value for the datatype property to be created, and clickl "Create property".

- **Expected outcome**: That the participants will notice that they could assign the same property with different values to other individuals in the knowledge base.

Tasks assessment

Similarly to Chapter 6, the assessment of these nine tasks involved qualitative and quantitative data:

- During the experiment, notes regarding the execution of tasks for each participant were taken and later compiled into a spreadsheet. In addition, the screen interaction together with the audio were also recorded;

- The interaction within each task was captured by the prototype: number of clicks and which buttons were clicked (except drop-down boxes, for technical reasons), time taken, queries created, nodes and rows highlighted.

After analysing the notes and watching again the screen recording, it should possible to identify the tasks in which the participants had most problems. Whenever they had any issue, they could ask for a "minor" advice. If even so they could not finish the task, then they were guided through it. That was necessary because some tasks depended on the result of the previous ones.

Finishing all the tasks was also important to make sure the participants understood the connection of different activities as part of a bigger investigation process, so they could rate their experience afterwards. Therefore, all participants did complete all the tasks, and the individual assessment was divided in three possibilities: "solved alone", 'minor advice given" or "guidance needed" for each task.

7.2.5 Interview

The last phase of the experiment comprised a semi-structured interview and a mini-questionnaire. Differently from Chapter 6, in which there was only the final questionnaire, applying a semi-structured interview had two goals: to confirm that the participants understood operating the prototype, and to try to get more complete answers regarding thee overall fitness of the proposed solution to investigating online banking fraud. In addition, how well they would rate each of the proposed semantic features, as they have tried all of them during the experiment. Finally, the interview also tries to capture whether they had any technical (usability bugs) or conceptual (lack of understanding of the technology) issues. There were a total of ten questions: seven discursive ones from the interview, and three multiple-choice ones from the questionnaire.

Discursive questions

1. *Please, comment how was your overall experience with SemInv*: During the analysis of the final questionnaire from Chapter 6, it was noticed that participants were giving similar answers to the first two questions, which were later found to be related. Thus, for the interview of the expert assessment study there is only one question trying to capture which words comes to the mind of the participants when asked about their experience with the prototype.

2. *What can SemInv improve?*: Considered together with the analysis of the tasks in which the participants had most problems, the replies to this question will inform improvements for the next version of the prototype.

3. *Which features should be kept the same?*: The goal of this question is to check whether any of the implemented features will be mentioned here before asking for them explicitly in questions eight and nine. This would be a good indicator that they have appreciated specific features.

4. *Did you find building queries challenging? If so, why?*: This question, analysed together with the performance in tasks which demanded a query to be built, will clarify if any potential issues reported by the user derive from the GUI or from a lack of understanding of the technology.

5. *Is there anything in this experience or experiment setup you found particularly problematic and would like to highlight (introductory videos, tasks, etc)? If so, what?*: This question tries to identify if there were any external issues during the experiment which could have affected the performance of the participants.

6. *Do you think that the materialisation features could help to organise the leads of an investigation?*: This inquiry was one of the main motivations for the research presented in this Thesis, and seeks to confirm if the participants agree with it.

7. *Do you think SemInv could be useful for investigating other crimes apart from online banking fraud? If so, which?*: The objective of this question is twofold: in addition to identifying potential real-world problems that the prototype could be applied to, it also confirms that the participant had understood that the prototype is generic: to be applied in different domains, it would only be necessary to create a new ontology and convert the dataset being explored to a linked data format.

Multiple-choice questions

Questions 8 and 9 asked the participant, respectively, how *important* each of the main implemented features are for investigation purposes, and how *easy* was to operate them via the prototype. These are the same features proposed in Table 1.1:

- Creating bespoke queries;
- Materialising (creating) clusters;
- Materialising (creating) relationships;
- Materialising (creating) tags;
- Merging items from distinct datasets;
- Exploring the dataset using a graph;
- Working with different versions of the dataset;
- Reusing queries and rules.

Finally, question 10 presented a list outlining potential future improvements for the prototype, and asked the participant to value each of them, from the most to the least relevant. The list comprises six suggestions plus an open choice field, in which the participant could add any non-listed improvement.

- Decreasing the learning curve;
- Improving graph exploration;
- Integrate data with other datasets;
- Make the prototype faster;
- Exchanging rules and queries with other users;
- Turn the GUI simpler;
- Others (please specify).

7.3 Results and analysis

The results of the experiment will be described separately, following the three main phases:

7.3.1 Initial questionnaire

Demographics

All eight participants were male police officers used to conducting investigation in online banking fraud. In terms of age, there were:

- four participants with age between 30 and 39 years;
- three participants with age between 40 and 49 years;
- one participant with age between 50 and 59 years;

Regarding the graduation courses of the participants:

- Three of them graduated in Law;
- One of them graduated in Psychology;
- One of them graduated in Architecture;
- Three of them have graduated in computer-related degrees: (two bachelors in data processing and computer science and one with specialisation in computer networks).

Background

From now on, the participants will be referred to with an "@" followed by a specific letter, in the case it is necessary to identify and relate their replies and comments.

Table 7.6: Initial questionnaire replies.

Replies per participant	@n	@f	@g	@b	@j	@m	@q	@u
Years investigating online banking fraud?	8+	2-4	2-4	2-4	2-4	8+	4-6	4-6
Graduated in computer-related course?	x					x	x	
Already done a course on investigative tools?	x	x	x		x	x	x	x
Heard about semantic technologies before?	x	x	x			x		
Take notes during the course of an investigation?	x	x	x		x	x	x	x
Already wrote a SQL-query? before?	x	x				x	x	

Four participants have between two and four years investigating online banking fraud. Two of them reported 8+ years, as they were among the creators of the Tentacles Project. The last two participants have between 4 and 6 years of experience, and have special roles in the group: in addition to investigation tasks, one is the database administrator and also developer of new solutions within Project Tentacles, and the other provides training for other investigators.

Seven of them have already done courses on investigative tools, particularly the I2 framework (comprising IBase and the Analyst Notebook), as this was the software acquired by the Brazilian Federal Police for investigation purposes. These trainings explore the dataset from Project Tentacles, so the participants learn about the online banking fraud investigation domain as well. One of participants (@b) reported never having done a course on investigative tools before.

According to question 6, four participants have already written a SQL query before: the three ones with a computer science-related degree plus one who recently started helping to develop the tools used in the Tentacles Project.

Regarding the question about semantic technologies, four participants have already heard about it, including the two creators of the Tentacles Project, who might have long being offered and assessing different tools . However, no participant had reported to have ever used such technologies, similarly to the participants from the usability assessment described in Chapter 6. Therefore, some level of difficulty in understanding and applying the semantic features during the experiment is expected.

Investigation practices

According to Question 8, all but one participant use the I2 framework for their daily investigations. Although being one of the creators of Project Tentacles and thus having already done a course on I2 in the past, the current duties of participant @m are more related to reverse-engineering malware: in Question 8, he reported using reverse engineering tools and malware analysers such as IDA Pro[1] and Cuckoo[76]. Furthermore, five of them reported using official databases (although it is believed that one of the two who did not mention using them forgot to do so), and three participants added that they also search OSINT.

Question 9 asked whether the participants take notes when conducting investigations and, if so, what kind of information do they annotate? Seven of them replied yes (as shown by Table 7.6). Name, address, phone number and money sums were the top voted ones, with three, two, two and two mentions respectively. Participants @q, @j and @u replied more generically: "Qualificação dos suspeitos" (*suspect qualification*), "Info dos suspeitos bem como das fraudes" (*"info" about the suspects as well as about the frauds*) and "Dados cadastrais" (*register data*).

In terms of the benefits of current tools and procedures, the most cited features were grouped in the following themes.

- Visualisation: three replies ("Fácil visualisação", "Interface", "Parte gráfica", meaning *easy visualisation, interface, graphical part*) respectively;
- Usability: three replies ("Facilidade do programa ", "Manuseio ", "Facilidade em descobrir vínculos ", meaning *Easiness of the program, handling* and *Easiness in finding links* respectively);
- Available data: three replies ("Quantidade de dados ", " dados fornecidos" and "Organização do banco de dados", meaning *Easiness of the program, handling* and *Database organisation*respectively).

Referring to the limitations, there were only two themes which emerged from the replies:

- Customization: two replies ("Customização" and "Falta de possibilidade de adaptação da estrutura da base ", meaning *customisation* and *no possibility of adapting the structure of the database*, respectively);
- Learning curve: two replies ("Inicio do aprendizado não é fácil" and "Complexidade na operação das ferramentas", meaning *the start of the learning is not easy* and *complexity regarding tool operation*, respectively).

[1]The Interactive Disassembler (IDA) is a disassembler for computer software which generates assembly language source code from machine-executable code. It supports a variety of executable formats for different processors and operating systems. It also can be used as a debugger for Windows PE, Mac OS X Mach-O, and Linux ELF executables. *Source*: https://en.wikipedia.org/wiki/Interactive_Disassembler. *Accessed in:* 22/08/2018

The following issues were all cited once: “Falta de interatividade entre ferramentas e bancos de dados disponíveis ”, “Compartilhamento de análises dificilmente são aproveitadas por outros ”, “Liberdade do usuário em criar relacionamentos ” and “Necessidade de procedimentos manuais para expansão da investigação”, meaning *Lack of interactivity between tools and available databases, analysis sharing are hardly leveraged by others*, *flexibility of the use in creating relationships* and *need for manual procedures for expanding investigations.*

Some of these mentions might have been inspired by the prototype and technology demonstration which happened before filling the questionnaire. Nonetheless, that does not nullify the observation that, although there is a perceived demand for such features, they are either not available in current investigation procedures or might be difficult to be applied. Finally, the potential antagonism regarding the mentioned benefit *Usability* and limitation *Learning curve* might be explained because Question 9 was later found to be too generic. Post-experiment discussion with the participants revealed that, in order to reply to this question, they had considered both the the I2 framework and the official databases, which have being recently renewed in terms of integration and usability.

In reply to Question 11, five participants argued that checking information from previous cases when investigating new cases is very important, and three replied totally important. As expected, there were no negative replies to this question, confirming that any feature which could facilitate future data retrieval is relevant. The prototype can assist this task by allowing the users to add their own “mark-ups” to the data (namely in the form of tags).

Similarly to Question 11, all participants replied positively to Question 12: when asked about the importance of sharing with colleagues the knowledge obtained in one investigation, half considered it very important, and the other half rated it as totally important. Although the prototype is not completely ready to allow such sharing (as the distribution of rules is not implemented yet, but only their generation and storing for future use), it is expected that the participants will be able to understand and assess this capability by reusing a rule created by themselves in a merged dataset during the tasks phase.

Finally, Question 13 asked how the participants share the knowledge obtained with other investigators. The most cited sharing mean was *email*, with a total of five replies. *Files (doc, xls)* and *databases* were mentioned twice and three times respectively. The following vectors were cited once: *meetings, instant message, videos, courses* and *yara*[2] *rules.* This latter citation, in addition to the great popularity of *Yara* in the malware domain, indicate that rule sharing could indeed be an effective way of sharing knowledge.

[2]YARA is the name of a tool primarily used in malware research and detection.It provides a rule-based approach to create descriptions of malware families based on textual or binary patterns. A description is essentially a Yara rule name, where these rules consist of sets of strings and a boolean expression. *Source*: https://en.wikipedia.org/wiki/YARA *Accessed* in: 23/08/2018

7.3.2 Tasks

Considering all tasks from all participants (as shown in Table 7.7), there were 39 of them completed without any help, 28 completed with a minor advice, and five in which guidance to completion was necessary.

Table 7.7: Task assessment.

Task	Completed alone	Advice given	Guidance needed
1	7	1	0
2	2	5	1
3	2	3	3
4	1	6	1
5	4	4	0
6	8	0	0
7	8	0	0
8	1	7	0
9	6	2	0
Total	39	28	5

The individual analysis of the tasks below comprises three plots: one displaying the time taken to complete the tasks (in seconds), one illustrating the number of buttons clicked, and one correlating these two metrics. The data points in the scatter plot have different colour codes:

- *Green*: the participant has completed the task by herself;
- *Yellow*: the participant only needed a minor advice to complete it;
- *Red*: the participant needed guidance throughout the task.

Task 1

Most participants performed well in this task, having finished it without any guidance or advice. Although having chosen the right dropdown list, participant @b did not see that the relationship `tentaculos#contaBenef__transferencia` was there. There were also suggestions to increase the font size and remove the prefix from the ontology concepts, which could make information easier to visualise. Finally, participant @g took longer than everyone else because, even having completed the task successfully, spent some time staring at the returned graph (apparently checking if any mistake was made). Observing the next tasks performed by him, it is possible to notice that @g normally took longer than the others: he was one of the most interested participant in the experiment.

Task 2

Task 2 was more difficult than Task 1. Although all participants realised they had to add more triples to the query, most of them had issues. Participants @f and @j did not choose the variable

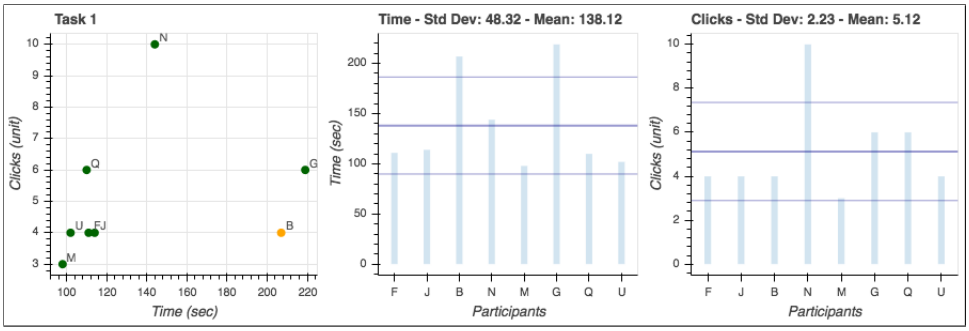

Figure 7.1: Clicks and timing for Task 1.

used in the first triple (`?contaBenef1`), resulting in creating two separate patterns. On the other hand, participants @m successfully added the triple `?contaBenef1 tentaculos#conta__pessoa ?pessoa1`, but did not remeber that *state* was a class as well: he tried to create the remaining triple using the properties column. Despite seeming to be an issue regarding how the ontology was designed, maybe having all object and datatype properties on the same dropdown list could make the process of building queries easier. Once @m was given the advice that "maybe *state* was not modelled that way", he could finish the task satisfactorily.

Participant @g had the same issues as @m. However, the high increase in clicks and time observed is due to his own exploration of the prototype (e.g. testing the highlight feature and counting nodes from the graph). Participant @n added only one triple (`?pessoa1 tentaculos#pessoa__estado ?estado1`) and thought that was enough. Once told it was not, he could add the remaining triple succesfully, choosing the right variables. Finally, participant @b had problems understanding the role of the triples in building queries, and thus had to be guided until adding `?contaBenef1 tentaculos#conta__pessoa ?pessoa1`. After that, he succeeded in adding the remaining triple and choosing the right variable by himself.

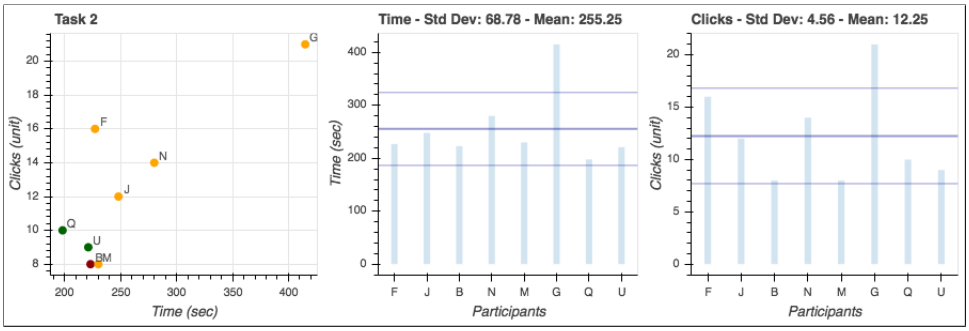

Figure 7.2: Clicks and timing for Task 2.

Task 3

Task 3 was the one who demanded more people to be guided through it. Participants @n and @g had confused "create a relationship" with "search for a relationship": the former involves materialisation, and the latter querying. The confusion might have arisen due to the frequent use of the expression "create a query". Participant @b did not remember that he could create a relationship from the results of a query, and asked for help in the very beginning of the task.

On the other hand, participant @f promptly created the new relationship. However, on the second part of the question, when it was necessary to restrict the scammers from the state "Minas Gerais", he used the wrong variable again: `tentaculos#Estado` instead of `?estado1`. On his remarks after finishing the tasks, he mentioned that better emphasis should be given to such variables. But another user gave an even better idea, discussed in Section 7.4. Lastly, participant @j commited minor mistakes: the former selected the variable `?contaBenef1` instead of `?pessoa1` in the materialisation area (which might have been to some regarding how *account* was modelled), and the latter chose the object property `tentaculos#pessoa__estado` instead of the recently created `tentaculos#fraudador__estado` in the second part of the question.

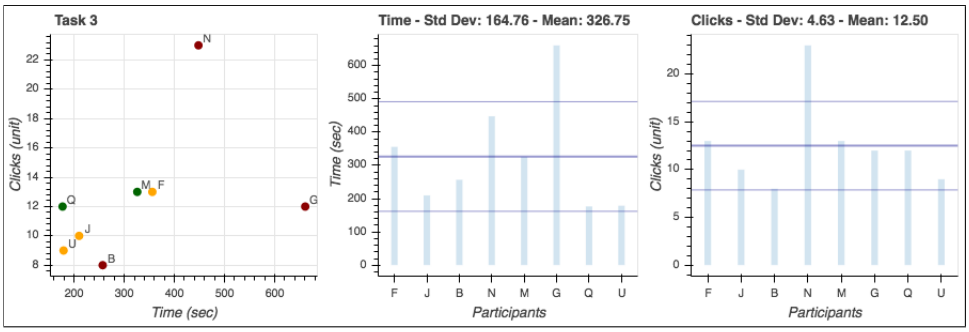

Figure 7.3: Clicks and timing for Task 3.

Task 4

This task was a follow up from Task 3, and demanded the participants to add more triples to the query until reaching the class `tentaculos#Transferencia`, so they could restrict it by date. Participant @b was still confused on how to add such triples (e.g. was searching for the datatype property `tentaculos#data_transacao` in the subject dropdown list) and needed to be guided through all the task. All the others except @m needed advice regarding the experiment schema, as "money transfer" is not represented as a node in their current investigation tool, but as a relationship with properties between two accounts. One possible reason that @m did not need advice in this task is because he has long been away from Tentacles Project, therefore its schema is not so deeply-ingrained in his cognitive reasoning as it might be to other officers who are conducting investigations routinely.

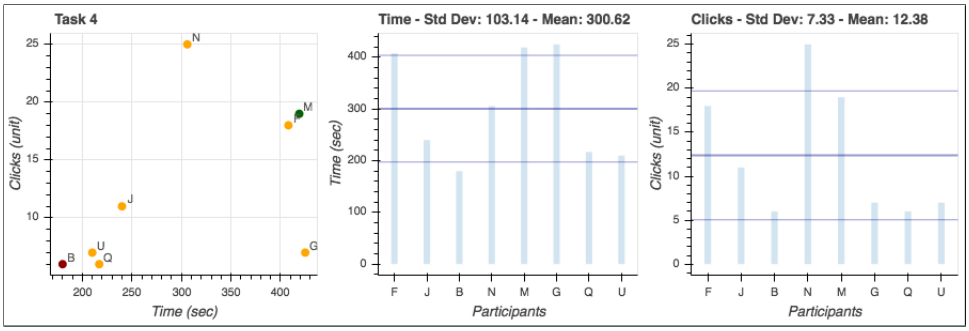

Figure 7.4: Clicks and timing for Task 4.

Task 5

In this task, participants @m, @f and @u had no trouble in spotting the nodes referring to `tentaculos#IP` in the graph and tapped all of them in order to make them available for querying and materialisation. That was the expected answer.

Otherwise, participants @g, @b and @j went straight to the materialisation area and, having not found the variable for `tentaculos#IP` there, got confused on how to move on in the task. The three of them were given the same advice of looking at the query and check if there was anything missing there. Participant @n realised that by himself and did added the necessary triple. However, he needed advice on how to exclude the IP starting with "200" from the query. He tried to restrict it using a datatype property but did not succeed. This attempt to solving the task (adding the triple `?ip1 != http://tdata/200.125.63.13` instead of tapping the specific nodes from the graph) was unexpected, but did work as well: participant @q had the same idea as @n but could finish the task successfully.

Finally, the large number of clicks observed for all users (as Figure 7.5 shows) is explained by the task itself (as it was necessary to tap nodes in the graph, and tapping outside the node would erase all the selected nodes and force the participant to repeat this procedure), and also to an unexpected but harmless behaviour of the prototype: both highlighting a specific class in the graph and selecting items in the table were counted as multiple clicks.

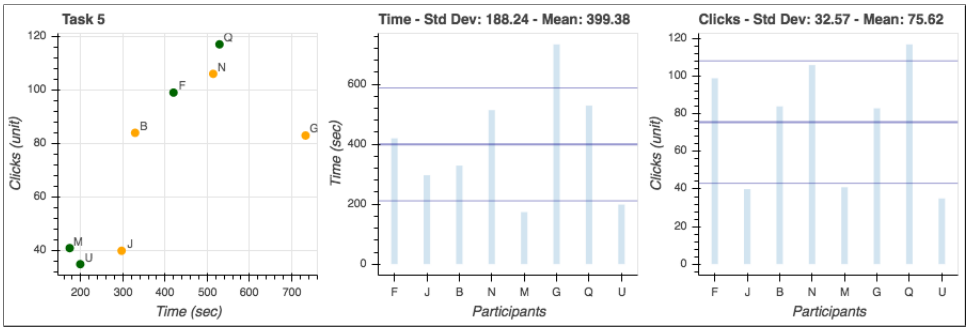

Figure 7.5: Clicks and timing for Task 5.

Task 6

Task 6 was relatively simple, and it was expected that no participant would have any problems with it. After all, it involved only running a single-triple query, checking the total individuals and relationships returned, and clicking one button, which adds more data to the knowledge base. It is thus more of a demonstration task. The difference in timings from participant @m to the others is due to the fact that it was not necessary to show him that the counts had increased: he anticipated that, and moved to the next task.

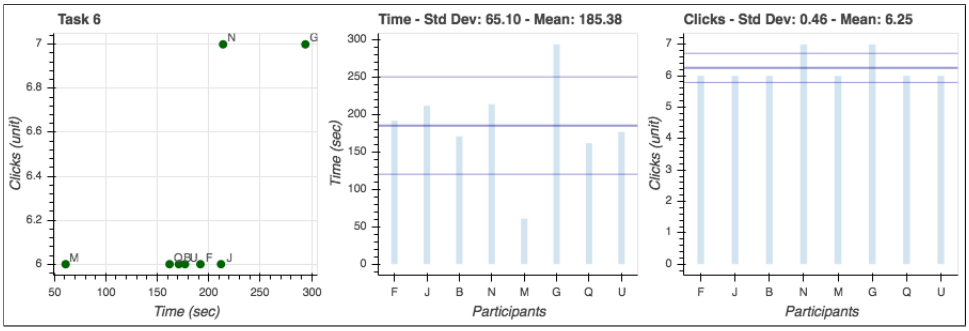

Figure 7.6: Clicks and timing for Task 6.

Task 7

Similarly to Task 6, Task 7 was also a demonstration task involving only simple steps: running the same query from Task 6 (so the participants would notice the increase in the number of individuals and relationships, as a result of adding more data) and then merging individuals from two classes. This last step is also easy to perform, requiring the user to select two values in drop-down lists and clicking a button.

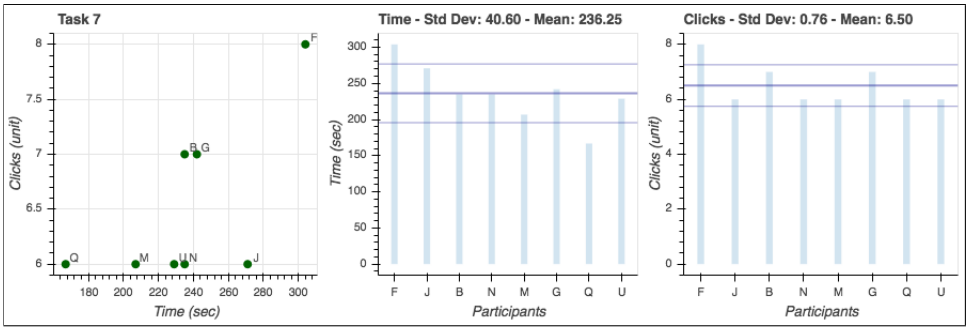

Figure 7.7: Clicks and timing for Task 7.

Task 8

The main issue of Task 8 was the confusion between the *rule* and *query* concepts. When asked to run again the rule created in Task 3, all participants but @m tried to build the same query from Task

3 again. Participant @m was the only one who remembered that the query created was converted into a rule for materialising the relationship `tentaculos#scammer__state` and was thus available to be executed again. Although suggesting that the difference between a rule and a query should be reinforced, this issue is not problem of the prototype. Once the participants were reminded of this feature, they showed appreciation in having their rules automatically saved and available to use again.

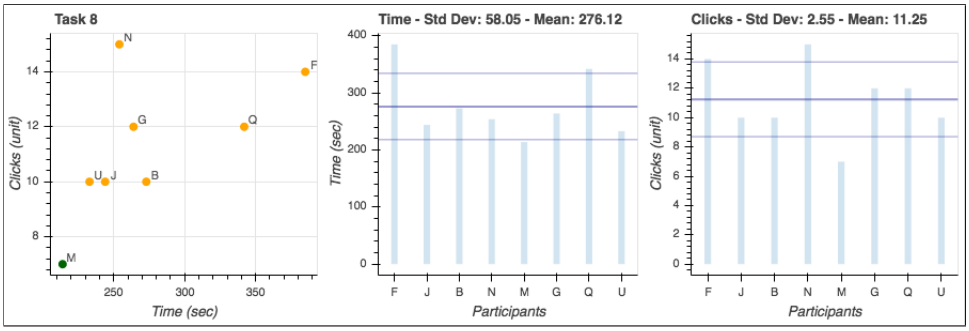

Figure 7.8: Clicks and timing for Task 8.

Task 9

Participants @f and @b were the only ones who were given advice in Task 9. More specifically, they needed to be reminded that the variable `?pessoa1` in the materialisation area already represented the people to be included in a case: @f wanted to do a new query, and @b did not remember he needed to select that variable. All the other participants had no problems in finishing this task.

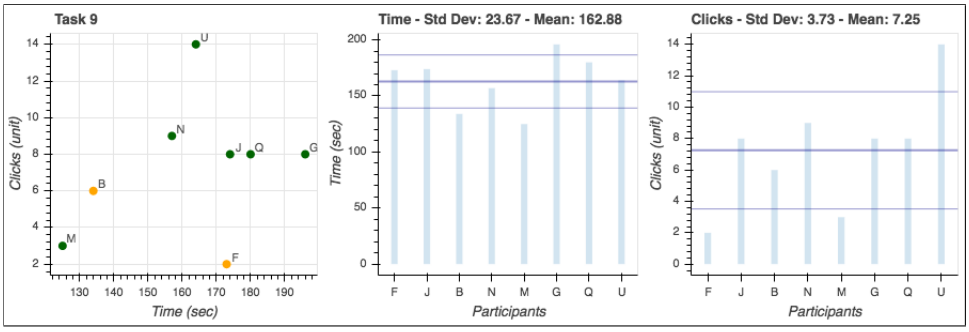

Figure 7.9: Clicks and timing for Task 9.

Analysis per participant

Figure 7.10 illustrates how many issues each participant had in completing the tasks. The green color indicates the number of tasks completed without any help; yellow indicates how many tasks were completed with only a minor advice; and red shows how many were completed with guidance.

Two participants (@q and @m, both with computer science-related degrees) performed better than the others. Participant @u was the third best: being an instructor from Tentacles Project (and thus having more experience with operating I2) might have gave him some advantage. Meanwhile, three of the participants (@g, @n and @b) needed guidance in at least one task. The latter was the one who performed worse: in addition to not have done any course on investigation tools, he was clearly anxious to finish all the tasks quickly, not seeming to give much importance in understanding their results.

The low number of tasks in which the participants needed guidance indicate a good understanding of the prototype features by the participants. However, the number of tasks completed without any help is similar to the number of tasks in which advice was necessary, suggesting that the GUI could be further optimized. The feedback from the interview, discussed in the next section, confirmed this assumption.

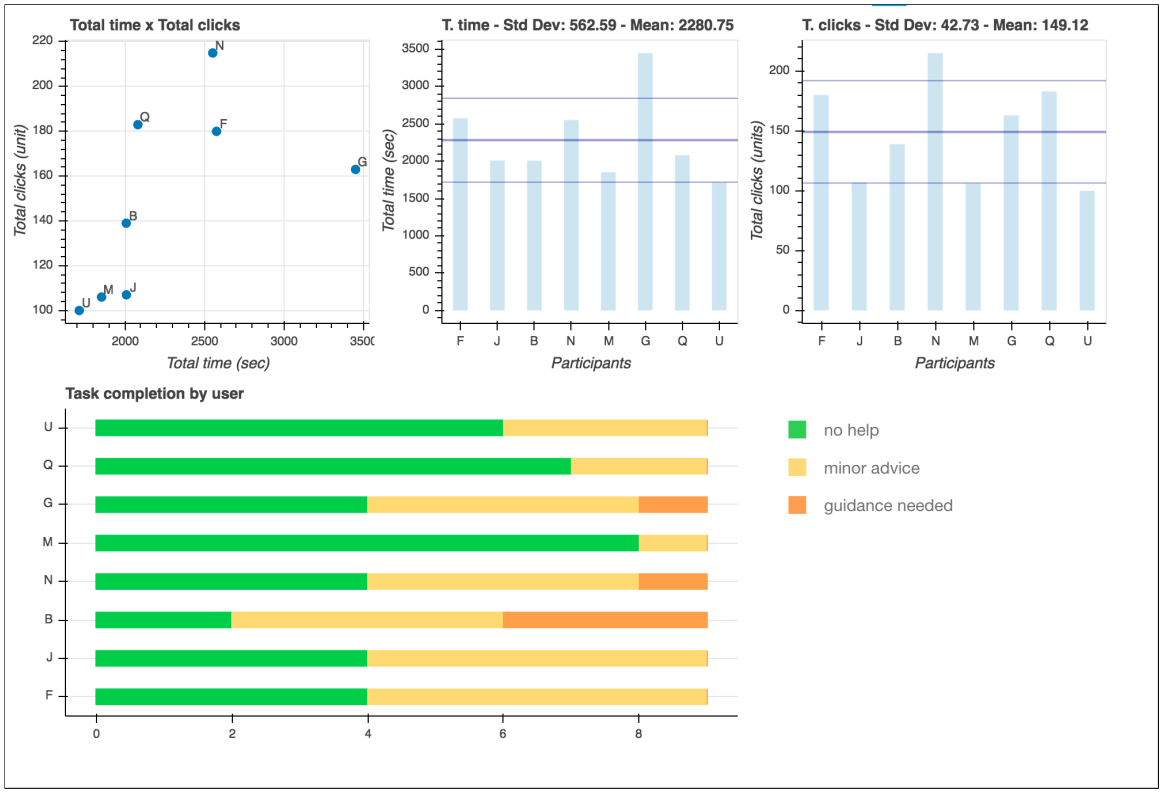

Figure 7.10: Task completion per user.

7.3.3 Interview

Similarly to Chapter 6, the interview assessment relied on a thematic analysis [73] of all the discursive questions enumerated in Section 7.2.5. Due to the small number of respondents for each question, it was not necessary to define codes, but only to identify the main themes.

Question 1

Question 1 asked the participants to tell how was their experience with the prototype. Two main themes emerged from their replies:

- **Good/powerful**: All participants gave positive feedback about the prototype. Six of them (@u, @n, @f, @j @m and @q) mentioned explicitly that the experiment was good. The two latter said: "Poderosa, muitos recursos" and "Boa. Faz o que os analisatas desejam: clusters para armazenar pesquisas, filtros sobre filtros, versões do banco" (*Powerful, lots of resources* and *Good. Does what the analysts want: clusters to store query results, filters over filters, versions of the database*, respectively). The other two participants (@g and @b) mentioned: "Faz muito sentido" and "Serve para investigação" (*makes a lot of sense* and *works for investigation tasks*, respectively).

- **Learning curve/Interface**: three participants (@n @m and @j) explicitily mentioned that more training would be necessary in order to get used to operating the prototype. More specifically, @g said:"Eu não estou acostumado com esse tipo de pensamento, de criar essas frases, mas eu entendi" (*I am not used to this kind of thinking, of creating these phrases, but i did understand*). He was referring to the process of adding triples in order to create complex queries (as asked by tasks 1 and 2, for instance). Two participants (@g, @u - the main developer and the instructor) were concerned with the GUI: the former mentioned "A interface tem muita coisa, mas não é confusa. Em alguns dias acostuma", meaning *the interface has too many things, but it is not confusing. One can adapt to it after some days.* Participant @u found the GUI complex, but also mentioned that the prototype had better features than the I2 tool, such as better querying capability (he was referring to both facet querying and the ability of keeping adding triples to the original query), possibility of the end user to add data to the investigation and the ability of creating relationships spanning the whole dataset.

  Finally, participant @n also mentioned that some difficulties he had were due to the current schema being deep-rooted in his mindset (the differences from the schema used in the experiment are listed in Section 7.1.2).

Question 2

The main theme which emerged from the replies to Question 2 was the GUI, with six participants (@u, @n, @g, @b, @m, @q) mentioning that it could be simpler and more intuitive: @b and @m agreed there are many controls in one screen, making it difficult to the final user to operate the prototype. Together with @m and @q, they suggested defining different tabs or views for the main features at least (queries and materialisation).

On the other hand, participants @f and @j liked the idea of having all controls on a single screen, having only suggested to better separate the different sections of the prototype and to expand the screen to use two monitors, respectively.

Furthermore, there were two mentions of making the *Counter* section more visible and three mentions on changing the colour of the background and making the fonts bigger (probably resembling he Analyst Notebook screen). Three participants also complained about the syntax of the variables used in connecting triples: they found in confusing having to choose, for instance, `?pessoa1` in the dropdown list. Moreover, one of them gave a good suggestion for this issue, which might not be difficult to implement: instead of adding triples as text, convert them to a graph, appending nodes to it once every triple is added. That way, it gets much easier for the user to find out if there are any disconnected triples, as most of the times that is not what the user wants.

Those were all spontaneous replies from the participants and, considered together with the replies to Question 1, it is clear that enhancing the usability of the prototype is an urgent issue. However, when asked specifically if any of the semantic features they have tried could be improved, none of them agreed with that. There are two possibilities then: either they were comfortable with the features, or did not know what to suggest, since applying such features to their routine tasks is probably novel to them.

Question 3

Having considered the replies from Question 2 (in which the participants were asked generically if any of the tested semantic features could be improved), the current question enunciated the functionalities listed in Table 1.1. All participants agreed that they should all keep the same. When prompted to talk more about them, the most reminded feature was the combination of integration and merge, with six participants explicitily mentioning it. @b said "Adicionar dados. Não tem como fazer hoje" (*Add data. Currently that is not possible to do*). That might be because the different databases normally searched by the investigators are not integrated: it is often necessary to search one of them, take notes about a specific person or something else (see the replies to Question 9 from the initial questionnaire) and search for related information in the investigation dataset, in a manual, granular way. @q stated: " No I2 dá para alimentar a base, mas aqui é mais fácil", meaning *it is possible to feed the i2 database, but it is easier to do it here.* Participant @u added that the end users cannot add data by themselves to, for instance, compare current investigation data with a potentially useful yet not completely reliable dataset.

Finally, participants @b and @j mentioned "criação" and "cluster" (*creation* and *cluster*) respectively, @f mentioned the facet query, and participants @n and @g liked the "click-and-query" feature.

Question 4

Regarding the potential difficulty in creating queries, three participants reported they had medium difficulty: @q would prefer having less text and more icons; @m had problems in the modelling of relationships and properties (as he did not know which was which), and @j stated: "(É necessário) habituar com a lógica, com a tecnologia", meaning *(It is necessary) to get used to the logic, to the technology.*

Participants @u, @f, @g and @b, said they had not problems in building queries, with the latter two explicitly saying: "Não. Dificuldade própria. Falta de vivência" and " Não. Mas tem que melhorar a estética", meaning (*No. Own difficulty. Lack of experience* and *No. But the aesthetics need to improve*). Despite their replies, both of them had poor performance in the tasks, which might indicate that they actually had some trouble in building queries.

Finally, participant @n reported he initially had problems in building queries, but after some time it got easier. Nonetheless, he suggested turning queries more graphical. Being one of the creators of the Project Tentacles and sharing the same opinion of the main developer confirms that turning the query builder even friendlier (perhaps the query builder is the feature which had more improvements since the first version of the prototype) is a necessary next step, which will nevertheless require further interactions with the end user.

Question 5

No participant reported that anything problematic had happened during the experiment.

Question 6

All participants agreed that the materialisation features demonstrated could indeed help to organize the leads of an investigation: there were five replies "Sim", and three "Com certeza", meaning *yes* and *absolutely*, respectively. When asked to further detail their answers, they replied:

- Participant @f: "Autonomia dada ao investigador é interessante. Monta o que quer" (*the autonomy given to the investigator is interesting. Create whatever she wants to*);
- Participant @g: "Não tem que ficar voltando para montar informação" (*One does not need to keep going back to assemble information*);
- Participant @j: "Dados anotados convertidos em clusters" (*Notes converted in clusters*);
- Participant @q: "Cluster permite continuar a investigação a partir de um ponto específico" (*Cluster enables continuing the investigation from a specific point*);
- Participant @n: " Com certeza. I2 não cria relacionamento" (*Absolutely. It is not possible to create relationships in I2*).

Question 7

Likewise Question 6, all participants considered that the prototype could be useful for investigating other crimes than online banking fraud, with four participants (@u, @f, @g, @q) explicitly adding "any kind of crime". When asked about examples, participants @u, @b and @g suggested social security fraud investigation, in which they could create queries searching for benefits to non-eligible people (e.g. people whose wage is higher than the threshold), defining clusters for that, and later searching for relationships with both public servants (who may have authorized fraudulent payments) and doctors (which may have forged medical certificates). Participant @n gave many examples such as money laundering, analysis of call records, patterns of money transfers, criminal hierarchy and relationship between stooges.

Question 8

According to Figure 7.11, the most relevant feature of the prototype is "merging items from distinct datasets", with a total of seven participants rating it "totally relevant", which is compatible with the replies to Question 3 of this questionnaire. Overall, all features were considered relevant: there were 43 "totally relevant" replies and 21 "very relevant" replies. Such figures suggest that, more than understanding how these features could be used in a real investigation setting, the participants agree that they could be very useful for their own tasks.

Question 9

Similarly to the final questionnaire described in Chapter 6, the results regarding the easiness of applying such features in an investigation supported by the prototype were not as good as their recognized relevance. Figure 7.11 shows that the features requiring the participants to choose variables (materialisation and query) were the most difficult ones. There were 35 negative replies ("a little" and "not at all") to this question, and 29 positive ones ("totally" or "very"), reinforcing the findings from Question 2 from this questionnaire: that the GUI needs more improvements.

Question 10

Finally, Question 10 asked the participants to rate how important they think some potential future features of the prototype are, from most important (1) to fifth most important(5). Table 7.8 shows that "Improving graph exploration" was the most wanted request, followed by "Turn the GUI simpler" and "Decreasing the learning curve". Essentially, even though not all participants found the GUI complex, all of them expect usability improvements in the prototype.

Moreover, there are two important considerations about the replies to this question: that the refactoring of the prototype was successful, as most participants did not find "making the prototype faster" an important enhancement; and that the low priority given to "integrate data with other

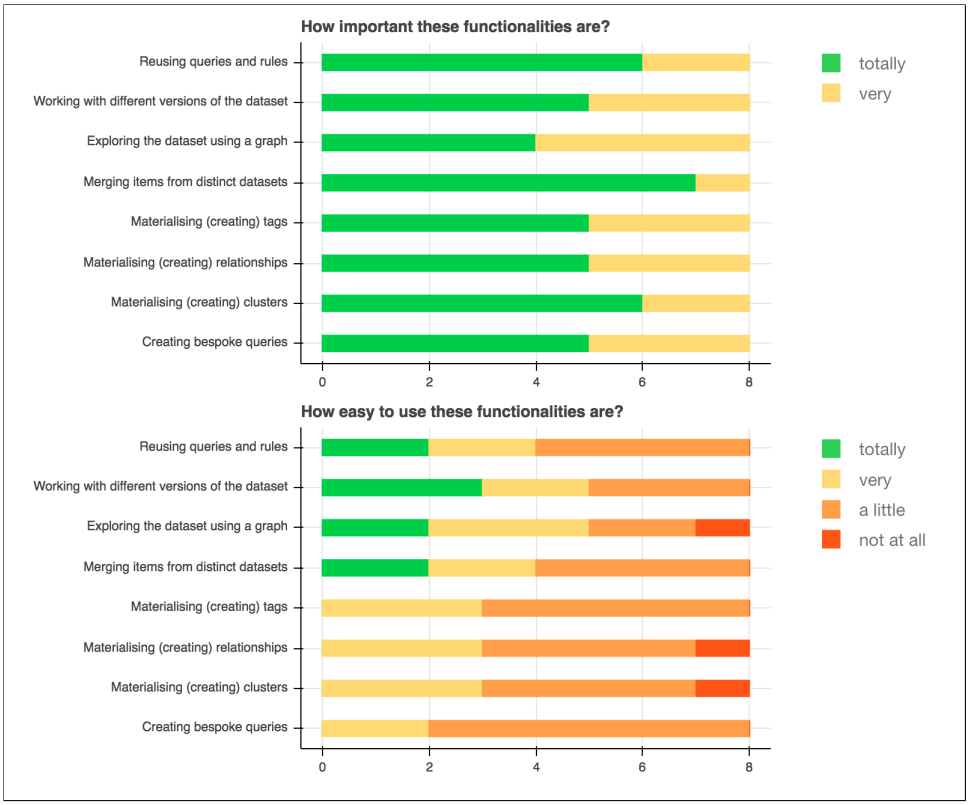

Figure 7.11: Features by relevance and easiness.

	n	f	g	b	j	m	q	u	Avg	Std Dev
Improving graph exploration;	3	2	2	2	1	1	2	2	1.88	0.64
Turn the GUI simpler;	2	1	1	1	6	4	4	1	2.50	1.93
Decreasing the learning curve;	1	3	4	3	4	5	3	3	3.25	1.16
Exchanging rules and queries with other users;	6	5	5	6	2	3	1	5	4.13	1.89
Make the prototype faster	4	6	3	4	5	2	5	6	4.38	1.41
Integrate data with other datasets;	5	4	6	5	3	6	6	4	4.88	1.13

Table 7.8: Ratings for potential future features.

datasets" is not incompatible with the relevance given to "Merging items from distinct datasets" from Figure 7.11. After all, it was made clear to the participants that the former referred to integrating official databases, and the latter referred to assessing small datasets during a specific investigation.

7.4 Conclusion

This experiment followed the same methodology from the one described in Chapter 5, for reasons outlined in Section 7.2. Thus, similarly to that chapter, it was considered complex to the participants. In addition to demanding them to learn and use multiples features in a short period of time, the

large number of classes and relationships might have confused the participants when searching the drop-down lists for specific values. Event though they are used to most of the schema used in the experiment, having that information displayed in a completely new, not-optimized interface might have made the tasks more difficult to accomplish.

Differently from the feedback given by the participants of the experiment described in Chapter 6, "querying" was not considered an issue in the current experiment. It was possible to notice that participants got used to the query builder towards the end of the experiment: although few participants still had issues with selecting the right variable, the process of selecting three values from the drop-down boxes and clicking "Append to query" became more natural. This is an important finding, as this simple yet powerful method of creating queries by appending triples will unlikely be changed in future versions of the prototype. Moreover, the participants seemed to appreciate creating queries on top of the results of previous queries.

One of the creators of Project Tentacles and the main developer were particularly interested in how the sharing of rules with investigators in different states would be implemented. Their main concern was regarding multiple investigators applying their rules to a centralized knowledge base, making it useless. Although this has not been tested yet, one solution would be keeping the knowledge base in a central repository updated, which would be replicated to every state on a daily basis. There would be "official" rules (curated by *GPA* and already loaded in the prototype), and "individual" rules, which would be created by the investigators based on the knowledge acquired in different cases, and could applied on-demand by other investigators.

Finally, the apparent bad result in Question 9 from the final questionnaire is more a confirmation that future work must be done than a negative issue for this Thesis. After all, improving the GUI was never one among its objectives, as opposed to assessing whether the semantic features listed in Table 1.1 could be useful for malware-related crime investigation, goal which was validated by this chapter.

Chapter 8

Discussion

Malware cybercrime continues to be a growing problem for society. The more people transfer part of their life to the virtual environment, the more the attack surface for criminals to act enlarges. However, such crimes are not being efficiently tackled by LEAs, which lack specialized human resources and which tools focus more in automating the processing of an ever-larger amount of data instead of assisting the human investigator in analysing it.

This thesis has explored particularly the challenge of improving malware-related cybercrime investigation conducted by humans and assisted by computers. For that, it presented the prototype *Seminv* which is based in semantic technologies, as they provide the means for the computer to "understand" real-world knowledge defined by humans. The overall goal of *Seminv* is to provide more flexibility to the end users in manipulating and analysing data. More specifically:

- Facilitate the assessment of investigative hypotheses in cases comprising highly-connected data (by leveraging the three first features listed in Table 1.1: *clustering entities*, *establishing links* and *inserting tags or comments*);
- Simplify the comparison of information stored in different datasets (by leveraging features 4, 9 and 10 of the same table - *enriching data about an entity*, *dataset integration* and *establishing provenance*, respectively;
- Improve the sharing of investigation knowledge and the automating of manual tasks (by leveraging features 5, 6 and 7 of the same table: *restoring different versions of the knowledge base* and *exchanging and reapplying queries and rules*);

The prototype *Seminv* was essential to answer the three research questions posed in Section 1.4:

1. By leveraging the semantic features listed in Table 1.1, the prototype could successfully reproduce the data exploration tasks from a real-world malware-investigation report in a potentially easier, more flexible manner;

2. Nine participants from an experiment were able to operate the prototype and understand how the technology leveraged by it could improve data exploration tasks in different domains of knowledge;

3. Eight cybercrime investigators who participated in another experiment agreed that the flexibility in exploring and manipulating data could indeed improve their current procedures of investigating online banking fraud. For this experiment, a dataset comprising over 3 million triples or real-world data was loaded into the prototype, proving that the approach is suitable for other real-world scenarios as well.

In common, the participants from both experiments relished the possibility of easily modelling domains which they are used to conduct analysis to, and to materialising facts according to their data exploration needs. In other words, the author of this thesis would describe that the participants, assuming an end-user role, enjoyed "having control over their data". For instance, the cybercrime investigators mentioned that there is relevant information which do not exist in any database (e.g. learning that two criminals know each other) and specially appreciated the possibility of materialising that information in the knowledge base and then running queries which would consider this new knowledge added to the investigation. Moreover, "merging items from distinct datasets" was the top-voted feature in relevance, as their routine tasks involves assessing datasets from multiple sources.

It is worth mentioning that the scalability issue of the prototype, as mentioned in Section 4.3.4, was also questioned by one of the reviewers of the article due to be published in January 2019 (item 5 from Section 1.5). The smooth performance of the third version prototype after loading over three million triples to it proved that that issue was resolved.

At last, the major issue identified by both groups was the non-optimized GUI, what is reflected in Table 7.8 and also by their replies to the question regarding how easy it was to operate the features of the prototype. Despite not being an objective of this thesis, improving the GUI must be tackled as future work due to its importance to any software used by humans.

Having achieved the objective of demonstrating that the innovation proposed in this thesis (as detailed in Section 1.4) and implemented into the prototype can be used for real-world cases and is considered helpful by real-world analysts, the next Section will list potential limitations of the presented approach and suggestions for future work.

8.1 Limitations and future work

During the development of the prototype, improving the scalability was thought to be left as future work. After all, a specific malware investigation normally involves less than one thousand entities, as related by one of the experts interviewed for the literature review. However, the preparation for

the final validation with the expert users forced this limitation to be tackled during the thesis (as described in Section 7.1.1).

Being a software-oriented approach which completely relies on the input of end-users, other potential limitations not captured during the experiments could emerge during real-world usage of the prototype. For instance, could the fact of having multiple investigators exchanging and applying rules lead to inconsistencies in their knowledge base? Researching about that, in addiction to improving the GUI, are suggested as future work:

- Improving the GUI: Some usability issues which emerged from Chapter 6 were fixed before running the main validation with the expert users. Some examples are editing queries and the click-and-query functionality. However, the findings from Chapter 7 indicate that the GUI still need improvements which could only be addressed by a comprehensive design project applying best practices from *visual analytics* to the prototype.

- Implementing and assessing rule sharing among multiple investigators: even though "Exchanging rules and queries with other users" was not the top-ranked feature in Table 7.8, post-experiment discussions with the head of the *GPA* and its main developer suggested that this feature is worth further research. After all, avoiding manual and duplicate work regarding data exploration tasks involving multiple analysts would certainly produce better investigation results.

Bibliography

[1] S. L. Garfinkel, "Digital forensics research: The next 10 years," *Digital Investigation*, vol. 7, pp. S64–S73, Aug. 2010.

[2] K. Ruan, J. Carthy, T. Kechadi, and M. Crosbie, "Cloud forensics," in *IFIP International Conference on Digital Forensics*, pp. 35–46, Springer, 2011.

[3] S. Malby, R. Mace, A. Holterhof, C. Brown, S. Kascherus, and E. Ignatuschtschenko, "Comprehensive study on cybercrime," *United Nations Office on Drugs and Crime, Tech. Rep*, 2013.

[4] M. McGuire and S. Dowling, "Cyber crime: A review of the evidence," *Summary of key findings and implications. Home Office Research report*, vol. 75, 2013.

[5] S. Schjølberg and S. Ghernaouti-Hélie, *A Global treaty on cybersecurity and cybercrime: a contribution for for peace, justice and security in cyberspace*. Cybercrimedata, 2011.

[6] F. Mercês, "The brazilian underground market." `https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-the-brazilian-underground-market.pdf`, 2015. Accessed: 2016-08-07.

[7] RSA, "Current state of cybercrime." `https://www.rsa.com/content/dam/rsa/PDF/2016/05/2016-current-state-of-cybercrime.pdf`, 2016. Accessed: 2017-04-05.

[8] R. Manning and G. Aaron, "Phishing activity trends report." `https://docs.apwg.org/reports/apwg_trends_report_q1-q3_2015.pdf`, 2015. Accessed: 2016-11-05.

[9] C. Sauerwein, C. Sillaber, A. Mussmann, and R. Breu, "Threat intelligence sharing platforms: An exploratory study of software vendors and research perspectives," in *Towards Thought Leadership in Digital Transformation: 13. Internationale Tagung Wirtschaftsinformatik, WI 2017, St.Gallen, Switzerland, February 12-15, 2017*, 2017.

[10] Europol, "Avalanche network dismantled in international cyber operation." `https://www.justice.gov/opa/pr/avalanche-network-dismantled-international-cyber-operation`, 2016. Accessed: 2017-02-22.

[11] R. van Baar, H. van Beek, and E. van Eijk, "Digital forensics as a service: A game changer," *Digital Investigation*, vol. 11, pp. S54–S62, 2014.

[12] R. J. Heuer Jr, R. J. Heuer, and R. H. Pherson, *Structured analytic techniques for intelligence analysis*. Cq Press, 2010.

[13] P. Szekely, C. A. Knoblock, J. Slepicka, A. Philpot, A. Singh, C. Yin, D. Kapoor, P. Natarajan, D. Marcu, K. Knight, and others, "Building and using a knowledge graph to combat human trafficking," in *International Semantic Web Conference*, pp. 205–221, Springer, 2015.

[14] N. Marie and F. Gandon, "Survey of linked data based exploration systems," in *Proceedings of the 3rd International Conference on Intelligent Exploration of Semantic Data-Volume 1279*, pp. 66–77, CEUR-WS. org, 2014.

[15] A. Bundy, "The need for hypotheses in informatics." `http://www.inf.ed.ac.uk/teaching/courses/irm/notes/hypotheses.html`. Accessed: 2017-03-23.

[16] M. Epifani and F. Turchi, "Standard for the electronic evidence exchange," in *Handling and Exchanging Electronic Evidence Across Europe*, pp. 311–335, Springer, 2018.

[17] W. Alink, R. Bhoedjang, P. Boncz, and A. de Vries, "XIRAF – XML-based indexing and querying for digital forensics," *Digital Investigation*, vol. 3, pp. 50–58, 2006.

[18] S. Raghavan, "Digital forensic research: current state of the art," *CSI Transactions on ICT*, vol. 1, no. 1, pp. 91–114, 2013.

[19] S. Garfinkel, P. Farrell, V. Roussev, and G. Dinolt, "Bringing science to digital forensics with standardized forensic corpora," *Digital Investigation*, vol. 6, pp. S2–S11, 2009.

[20] S. L. Garfinkel, "Forensic feature extraction and cross-drive analysis," *Digital Investigation*, vol. 3, pp. 71–81, 2006.

[21] Fabio Marturana and S. Tacconi, "A machine learning-based triage methodology for automated categorization of digital media," *Digital Investigation*, vol. 10, no. 2, pp. 193–204, 2013.

[22] A. Case, A. Cristina, L. Marziale, G. G. Richard, and V. Roussev, "FACE: Automated digital evidence discovery and correlation," *Digital Investigation*, vol. 5, pp. S65–S75, 2008.

[23] Access Data, "The forensic toolkit." `https://accessdata.com/products-services/forensic-toolkit-ftk`. Accessed: 2015-12-05.

[24] S. Bunting and W. Wei, *EnCase Computer Forensics: The Official EnCE: EnCase Certified Examiner Study Guide*. John Wiley & Sons, 2006.

[25] S. L. Garfinkel, "Automating disk forensic processing with sleuthkit, xml and python," in *Systematic Approaches to Digital Forensic Engineering, 2009. SADFE'09. Fourth International IEEE Workshop on*, pp. 73–84, IEEE, 2009.

[26] S. Garfinkel, "Digital forensics XML and the DFXML toolset," *Digital Investigation*, vol. 8, no. 3, pp. 161–174, 2012.

[27] M. Cohen, S. Garfinkel, and B. Schatz, "Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow," *Digital Investigation*, vol. 6, pp. S57–S68, 2009.

[28] S. Garfinkel, D. Malan, K.-A. Dubec, C. Stevens, and C. Pham, "Advanced forensic format: an open extensible format for disk imaging," in *Advances in Digital Forensics II*, pp. 13–27, Springer, 2006.

[29] Damir Kahvedžić and Tahar Kechadi, "DIALOG: A framework for modeling, analysis and reuse of digital forensic knowledge," *Digital Investigation*, vol. 6, pp. S23–S33, 2009.

[30] C. Hargreaves and J. Patterson, "An automated timeline reconstruction approach for digital forensic investigations," *Digital Investigation*, vol. 9, pp. S69–S79, 2012.

[31] B. W. P. Hoelz and C. G. Ralha, "A framework for semantic annotation of digital evidence," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, (New York, NY, USA), pp. 1966–1971, ACM, 2013.

[32] S. Dossis, "Semantically-enabled digital investigations," Master's thesis, Department of Computer and Systems Sciences, Stockholm University, Sweden, 2012.

[33] V. R. Basili, G. Caldiera, and H. D. Rombach, "The goal question metric approach," in *Encyclopedia of Software Engineering*, Wiley, 1994.

[34] E. Casey, G. Back, and S. Barnum, "Leveraging cybox to standardize representation and exchange of digital forensic information," *Digital Investigation*, vol. 12, pp. S102–S110, 2015.

[35] S. Barnum, R. Martin, B. Worrell, and I. Kirillov, "The cybox language specification," *The MITRE Corporation*, 2012.

[36] D. A. Mundie and D. M. Mcintire, "An ontology for malware analysis," in *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*, pp. 556–558, IEEE, 2013.

[37] D. McIntire and D. Mundie, *The mal: A malware analysis lexicon*, 2013.

[38] C.-B. Jiang and J.-S. Li, "Ontology-based botnet topology discovery approach with ip flow data," *International Journal of Innovative Computing, Information and Control*, vol. 11, no. 1, pp. 308–325, 2015.

[39] Hsien-De Huang, G. Acampora, V. Loia, C.-S. Lee, and H.-Y. Kao, "Applying FML and fuzzy ontologies to malware behavioural analysis," in *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pp. 2018–2025, IEEE, 2011.

[40] J. Kornblum, "Identifying almost identical files using context triggered piecewise hashing," *Digital Investigation*, vol. 3, pp. 91–97, 2006.

[41] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, *et al.*, "Swrl: A semantic web rule language combining owl and ruleml," *W3C Member submission*, vol. 21, no. 79, pp. 1–31, 2004.

[42] IEEE ICSG Working Group, "Malware metadata exchange format version 1.2." `http://grouper.ieee.org/groups/malware/malwg/Schema1.2/`. Accessed: 2016-04-27.

[43] I. Kirillov, D. Beck, P. Chase, and R. Martin, "Malware attribute enumeration and characterization. the mitre corporation," tech. rep., Tech. Rep, 2010.

[44] S. Barnum, "Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX)," tech. rep., MITRE, Feb. 2014.

[45] W. Gibb and D. Kerr, "Openioc: back to the basics." `https://www.fireeye.com/blog/threat-research/2013/10/openioc-basics.html`, 2013. Accessed: 2017-03-23.

[46] S. Caltagirone, A. Pendergast, and C. Betz, "The diamond model of intrusion analysis," tech. rep., Center for Cyber Intelligence Analysis And Threat Research, 2013.

[47] C. Wagner, A. Dulaunoy, G. Wagener, and A. Iklody, "Misp: The design and implementation of a collaborative threat intelligence sharing platform," in *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*, pp. 49–56, ACM, 2016.

[48] Europol, "Eu forensic experts call for action on new cyber investigation standard." `https://www.europol.europa.eu/newsroom/news/eu-forensic-experts-call-for-action-new-cyber-investigation-standard`, 2017. Accessed: 2018-09-06.

[49] M. Sporny, D. Longley, G. Kellogg, M. Lanthaler, and N. Lindström, "Json-ld 1.0," *W3C Recommendation*, vol. 16, 2014.

[50] H. Abelson, R. Anderson, S. M. Bellovin, J. Benaloh, M. Blaze, W. Diffie, J. Gilmore, M. Green, S. Landau, P. G. Neumann, R. L. Rivest, J. I. Schiller, B. Schneier, M. A. Specter, and D. J. Weitzner, "Keys under doormats: mandating insecurity by requiring government access to all data and communications," *Journal of Cybersecurity*, vol. 1, no. 1, pp. 69–79, 2015.

[51] S. Creese, T. Gibson-Robinson, M. Goldsmith, D. Hodges, D. Kim, O. Love, J. R. Nurse, B. Pike, J. Scholtz, and others, "Tools for understanding identity," in *Technologies for Homeland Security (HST), 2013 IEEE International Conference on*, pp. 558–563, IEEE, 2013.

[52] A. Bernardi, G. A. Grimnes, T. Groza, and S. Scerri, "The nepomuk semantic desktop," in *Context and Semantics for Knowledge Management*, pp. 255–273, Springer, 2011.

[53] M. Welsh, M. K. Dalheimer, and L. Kaufman, *Running Linux*. O'Reilly & Associates, Inc., 1999.

[54] T. Berners-Lee, J. Hendler, O. Lassila, and others, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.

[55] N. Shadbolt, T. Berners-Lee, and W. Hall, "The semantic web revisited," *IEEE intelligent systems*, vol. 21, no. 3, pp. 96–101, 2006.

[56] D. Beckett, T. Berners-Lee, E. Prud'hommeaux, and G. Carothers, "Rdf 1.1 turtle," *World Wide Web Consortium*, 2014.

[57] A. Iosup, T. Hegeman, W. L. Ngai, S. Heldens, A. Prat-Pérez, T. Manhardto, H. Chafio, M. Capotă, N. Sundaram, M. Anderson, *et al.*, "Ldbc graphalytics: A benchmark for large-scale graph analysis on parallel and distributed platforms," *Proceedings of the VLDB Endowment*, vol. 9, no. 13, pp. 1317–1328, 2016.

[58] Y. Nenov, R. Piro, B. Motik, I. Horrocks, Z. Wu, and J. Banerjee, "Rdfox: A highly-scalable rdf store," in *The Semantic Web - ISWC 2015* (M. Arenas, O. Corcho, E. Simperl, M. Strohmaier, M. d'Aquin, K. Srinivas, P. Groth, M. Dumontier, J. Heflin, K. Thirunarayan, and S. Staab, eds.), (Cham), pp. 3–20, Springer International Publishing, 2015.

[59] Bokeh Development Team, "Bokeh: Python library for interactive visualization." `https://bokeh.pydata.org/en/latest/`. Accessed: 2016-06-23.

[60] P. V. Biron, A. Malhotra, *et al.*, "Xml schema part 2: Datatypes," 2004.

[61] D. Calvanese, M. Giese, D. Hovland, and M. Rezk, "Ontology-Based Integration of Cross-Linked Datasets," in *The Semantic Web - ISWC 2015*, vol. 9366, pp. 199–216, Springer International Publishing, 2015.

[62] The Apache Software Foundation, "A free and open source java framework for building semantic web and linked data applications." https://jena.apache.org/. Accessed: 2015-06-16.

[63] S. Gupta, P. Szekely, C. A. Knoblock, A. Goel, M. Taheriyan, and M. Muslea, "Karma: A system for mapping structured sources into the semantic web," in *The Semantic Web: ESWC 2012 Satellite Events*, (Berlin, Heidelberg), pp. 430–434, Springer Berlin Heidelberg, 2015.

[64] IBM, "Ibm acquiring i2 for criminal mastermind software." https://www.pcworld.com/article/239228/ibm_acquiring_i2_for_criminal_mastermind_software.html. Accessed: 2017-12-08.

[65] IBM, "i2 analyst's notebook." https://www.ibm.com/us-en/marketplace/analysts-notebook. Accessed: 2017-12-05.

[66] IBM, "Creating links in the analysis repository." https://www.ibm.com/support/knowledgecenter/SSXVXZ_2.1.6/com.ibm.i2.portal.doc/adding_links_anbp.html. Accessed: 2017-12-08.

[67] N. Moran and B. Koehl, "The italian connection: An analysis of exploit supply chains and digital quartermasters." https://drive.google.com/file/d/0Bw35r_AUUldgMEZUdXUyWUR0T3M/view. Accessed: 2016-03-02.

[68] "What analysts can learn from shadowserver's "italian connection" report – CYINT analysis." http://www.cyintanalysis.com/what-analysts-can-learn-from-shadowservers-italian-connection-report/. Accessed: 2016-03-02.

[69] R. Carvalho, M. Goldsmith, and S. Creese, "Applying Semantic Technologies to Fight Online Banking Fraud," in *2015 European Intelligence and Security Informatics Conference*, pp. 61–68, IEEE, Sept. 2015.

[70] S. Lohmann, S. Negru, F. Haag, and T. Ertl, "VowlÂ 2: User-oriented visualization of ontologies," in *Knowledge Engineering and Knowledge Management*, pp. 266–281, Springer International Publishing, 2014.

[71] Google, "Introduction to structured data." https://developers.google.com/search/docs/guides/intro-structured-data. Accessed: 2017-02-21.

[72] C. Gena, "Methods and techniques for the evaluation of user-adaptive systems," *Knowledge Eng. Review*, vol. 20, pp. 1–37, 03 2005.

[73] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, pp. 77–101, Jan. 2006.

[74] E. P. de Siqueira, "O projeto Tentáculos da Polícia Federal," Master's thesis, University of Brasilia, Brasilia, 2014.

[75] A. Bryman, *Social research methods*. Oxford university press, 2016.

[76] C. Guarnieri, M. Schloesser, J. Bremer, and A. Tanasi, "Cuckoo sandbox-open source automated malware analysis," *Black Hat USA*, 2013.