Transforming Network Simulation Data to Semantic Data for Network Attack Planning

Ka Fai Peter Chan and Pedro de Souza
Council for Scientific and Industrial Research, South Africa
kchan@csir.co.za
pdsouza@csir.co.za

Abstract: This research paper investigates a technique to transform network simulation data into linked data through the use of ontology models. By transforming the data, it allows one to use semantic reasoners to infer and reason additional insight. A case study was performed, using the Common Open Research Emulator (CORE), to generate the necessary network simulation data. The simulation data was analysed, and then transformed into linked data. The result of the transformation is a data file that adheres to the Web Ontology Language (OWL) 2.0 eXtensible Markup Language (XML) format, which can be read, merged, and reasoned by ontology tools such as Protégé. Using the Web Ontology Language Application Program Interface (OWL API), it was possible to merge the transformed data with other ontology models to form a knowledge base for a specific field — particularly network warfare ontologies in this instance. The knowledge base can then be queried dynamically, similar to semantic based intrusion detection systems (IDS). For example using associated network data during a warfare operation in order to infer doctrines, operating procedures or related information. Overall, this research provides a step towards automating the transformation of network data to semantic data to aid network attack and defence strategies.

Keywords: semantic data, OWL, ontology, network simulation

1. Introduction

Network simulation is a technique used to model the behaviour of a network and the associated network nodes, in a testbed environment. Various attributes of such an environment can be modified, in a controlled manner, in order to observe the changes in network performance. These simulations are useful when planning a network attack, as it allows various setups to be tested prior to attack deployment. In addition these steps provide vital insight when defending a network.

Network simulators play a vital role in terms of supporting defensive and offensive capabilities, including:

- Network design
- Impact assessment on existing network
- Network research and development (R&D)
- Testing defence applications such as Mobile ad hoc Network (MANET)/Vehicular ad hoc Network (VANET)

Network simulation can provide a testbed similar to a sandbox model of a cyber battlefield. This allows users to create or replay scenarios, as well as receive detailed training. In traditional warfare, years of knowledge and research is passed down, enabling soldiers to be better equipped regarding how to react and handle certain situations, while optimising the completion of objectives. Unfortunately this is not as apparent in cyber warfare and as a result the use of semantic models, such as ontologies, has been introduced to aid decision making and align doctrines.

Ontology is a technology that provides a way to exchange semantic information between people and machines (Noy et al., 2001). It allows a formal, explicit specification of a shared conceptualisation that can be shared in a domain (Grüber, 1993), providing a common vocabulary.

The languages that represent ontology models include Resource Description Language (RDF) (W3C 2000), Web Ontology Language (OWL) (W3C, 2012), JavaScript Object Notation-Linked Data (JSON-LD) (W3C, 2014), and their sub-variations. They allow one to provide a semantic representation of entities through formal constructs and relationships. The use of these constructs and relationships allows the mapping of classes and instances to existing knowledge bases, exposing a larger pool of knowledge — enriching existing information and enabling better situation awareness.

This paper aims to take advantage of the information enrichment ability of semantic data, and will investigate the transformation of network data into semantic data, as well as the associated benefits thereof. The research provides a step towards automating the transformation of network data into semantic data, in order to aid network attack and defence strategies. The approach that will be taken for this paper is as follows:

- Transform network simulation data into semantic data.
- Merge semantic data with existing knowledge bases.
- Reason using the merged data to create more enriched information.

In the next section, an overview of network simulations and network simulators that are currently available is provided. Section 3 then provides an overview of how ontologies are used in network operations. The case study that was performed is then discussed in Section 4. The paper is then concluded in Section 5, summarising the results of the investigation and discussing the way forward.

2. Network simulation

Network simulations have been around since the early 1990s, beginning with Realistic and Large (REAL) (Keshav, 1988), which was derived from Network Simulation Testbed (NEST) (Dupuy *et al*, 1990). While the number of network simulation tools have expanded since then, many of the features have been carried over.

The main focus of a network simulation is to allow users to evaluate network applications, topologies and protocols under varying conditions. Allowing them to be studied, both statically and interactively, to gain a greater understanding of their behaviour and how they operate. Additional features of network simulations include abstraction, emulation, scenario generation, and trace data (Breslau *et al*., 2000).

- Abstraction refers to the granularity of the simulator to accommodate for the level of detail that entities involved in the simulation can have. By supporting detailed protocols, users can define specific protocols, and even go as far as specifying the associated state machines. On the other hand, users should not need to configure each network node, if they wish to deploy hundreds of nodes at once. A good example showing both of these levels of granularity is Riverbed's proprietary application, called OPNET (Riverbed, 2016). OPNET allows users to deploy generic nodes, even going as far as to allow configuration of the state machines of specific routers.
- Emulation refers to the inclusion of real world network nodes. This allows the simulator to interact with operational nodes and lines. This is useful in the case of testing proprietary hardware, such as firewalls or end-to-end encryption devices.
- Scenario generation refers to the automatic creation of traffic patterns, topologies, and dynamic events — allowing for more accurate real world testing, outside of a vacuum environment. Examples include line jitter and interference.
- Lastly, trace data refers to the capturing of network traffic generated by the simulation. Trace data serves a vital role in testing and assessing different configurations — to ensure optimal improvements are done to the network setup. Trace data can also be used for repeatability to confirm the validity of test results.

Trace data is the network data that this research aims to transform into semantic data. The format of the data will be explored in more detail in Section 4, once a network simulator has been selected for the case study. The next section provides a short list of some of the open source network simulators that offer the above-mentioned features.

2.1 Network simulators

This section provides a short list of open source network simulators.

- Cloonix
- CORE Emu
- GNS3
- IMMUNES
- LINE
- Marionnet

- Mininet
- Netkit
- NS-3
- Psimulator 2
- Shadow

At the time of writing, Cloonix, was not available for download and test. GNS3 is primarily focused on supporting Cisco and Juniper Software, which may be too limited in test that must be conducted as part of this research. LINE network simulator requires the user to have intermediate network knowledge to construct a simulation. Although Marionnet was designed to be an educational tool for teaching networks, the lack of documentation deterred the use of the application. Mininet focuses mainly on Software Defined Network (SDN) technology, which was considered too specific for the purpose of this paper. Netkit is primarily a command-line based simulation platform, with a minimal graphical user interface (GUI).

Although the simulators all share similar traits, each have their own limitations. The aim of this research is to consider the most accessible simulator for the case study. Accessibility refers to the ease of use and the availability of the software to reproduce the test results. Factors such as user interface and setup time is considered, as well as the authors experience using the particular tool. As such, CORE Emu was chosen for the purpose of this research. Case studies regarding the other simulators are considered for future work.

3. Background

An ontology is a formal and explicit description of concepts in a domain, consisting of concepts or classes, properties of each concept describing its features and attributes of the concept, represented as properties, and restrictions on the concept. Classes, properties and restrictions, together with individuals — form a knowledge base within a defined domain. Classes are the main entities of an ontology model, and they represent and describe a concept. These classes can be further refined into subclasses. For example, a Pet class can be refined into Cats and Dogs, which can then be further refined into different breeds of Cats and Dogs respectively.

Properties, namely data and objects, describe the classes with relations to other classes. An example of a data property would be the age of a Pet. Data properties give a class or concept value. In this example, hasAge would be the data property of a Pet, representing the age of the Pet. Object properties capture relationships of the class, for example, hasGender which would describe the gender of the Pet.

Individuals are specific instances of a class, such as Fluffy which would be a specific individual under the class Dog. With the age and gender properties, and possibly other properties, Fluffy can be described uniquely, in a similar manner to a unique entry within a database. Figure 1 represents an overview of the concepts in an ontology. In the figure, the example of the Pet class, with the Fluffy individual can be seen. The lines between the circles, such as hasPet and hasSibling represent the object properties, also known as relationships. Relationships link multiple classes together to form a network of information.

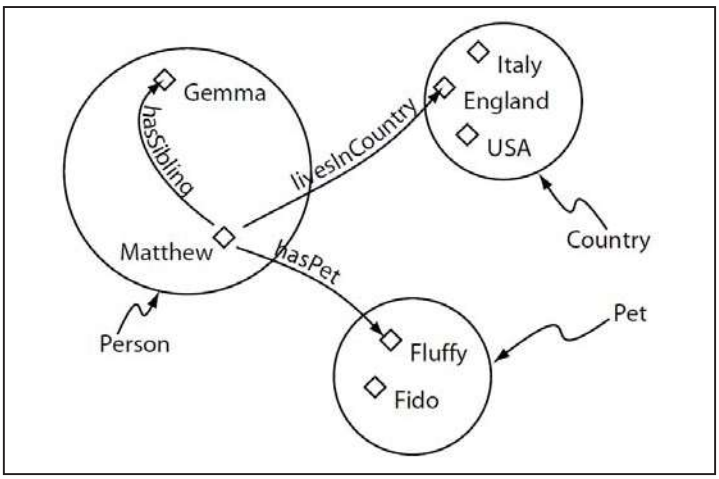

Figure 1: Ontology concepts

3.1 Ontology in network operations

The involvement of ontologies in cyber operations, as well as the evolution thereof, can be seen by the advances of Command and Control (C2) ontologies (Curts, and Campbell, 2005), (Stoutenberg et al, 2007) (Tolk, and Smith,2011), and other common data models, such as the Unified Cybersecurity Ontology (UCO) (Syed et al., 2016). Although these upper ontologies are represented at a high level, their importance is highlighted by how they provide a common understanding of the field that one is working within. Ontologies have also been used at an operational level. This has been demonstrated by the advances in ontology-based intrusion detection systems through the years (Abdoli, and Kahani, 2009), (Li, and Tian,2010), (Retnaswamy, and Ponniah, 2016), and detection of network scenarios (van Heerden, Burke, and Irwin,2012a). These ontologies make use of automated reasoners to deduce or infer the type of attack based on network traffic or events.

More specific to the domain of this research, is the use of ontology in network planning. Chan (Chan, Theron, van Heerden and Leenen, 2015), and van Heerden (Van Heerden, Chan and Leenen, 2016) have shown the ability to aid network planning by using ontologies. They proposed a top-down approach to provide detailed information on tasks. An example would be to show the chain of command when querying which military personnel has access to a network tool. To better explain this, one can use the analogy of a cook-book where users can query how to make a given meal and the results would highlight the required ingredients – capturing both the explicit and implicit items. The research in this paper aims to address the reverse, where one would start with a bottom-up approach where items in a network simulation should be mapped to operational level entities. For example, a perimeter firewall should be tied to a specific military base's policy or doctrine that is not necessarily modelled in a network simulation. This doctrine can be also linked to specific operating procedures and chain of command for work authorisation. This information may not be modelled in a network simulation, but can be vital on the operational level. By enriching network simulations, it can provide operational awareness to both the technical team and the commander. This extends the simulation by going beyond just the network and incorporating the people, and the processes involved during an actual operation.

4. Test case

The test case investigates the viability of transforming network simulation data into semantic data, and how this enriched data can provide more valuable information, facilitating better decision making in network operations. In order to create the test case, the preliminaries are identified. This includes the investigation of the data structure of CORE Emu and how to integrate the transformation capability into the application.

4.1 CORE Emu

In order to implement the transformation function, one must first understand the design of the CORE Emu application. This can be seen by the CORE Emu architecture, which is illustrated in Figure 2.

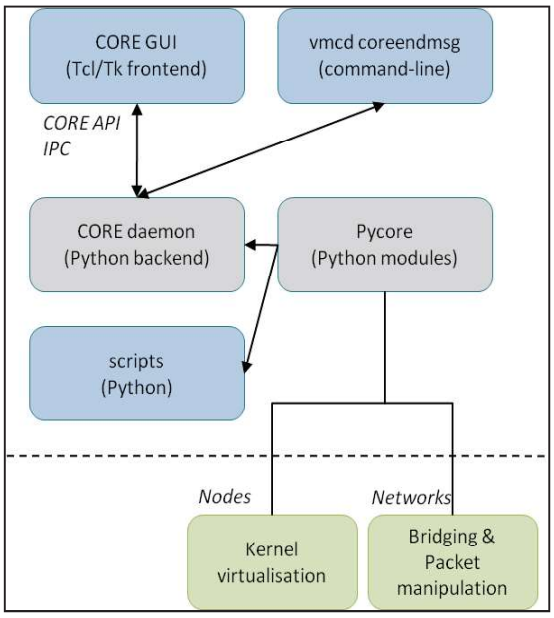

Figure 2: CORE Emu architecture

With reference to Figure 2, the Tcl/Tk frontend component, as well as the python backend which is used to handle the save file message events was the focus of our research. As part of this research's contribution, the function to transform data directly into an OWL file has been implemented in CORE (version 3.7), by expanding upon these components. Figure 3 illustrates the function that was added to the GUI of the application.

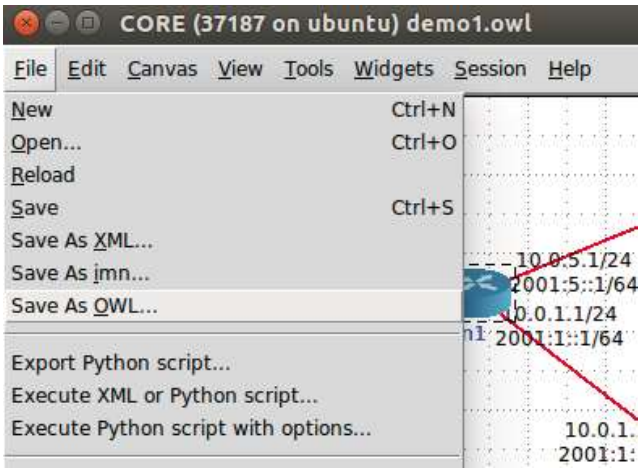

Figure 3: CORE — *Save As OWL* function

The following sections discuss how the source data structure can be transformed and the transformation process thereof.

4.2 CORE data

On the standard Graphical User Interface (GUI), the user can export the created network topology in XML or IMUNES network configuration (.imn) file format. Live data requires traffic capturing tools to be running while traffic is flowing through the network. Previous research, highlighted in the background, has shown how such live data can be used with ontologies in security fields such as the IDS environment. The focus for this paper is on the network topology data. Topology data is generated by populating the canvas with entities, such as nodes, switches and routers. Configured information on each node — such as the number of Ethernet ports, IP addresses and the associated services — can also be exported. Figure 4 shows a simple network configuration accomplished by populating the CORE canvas.

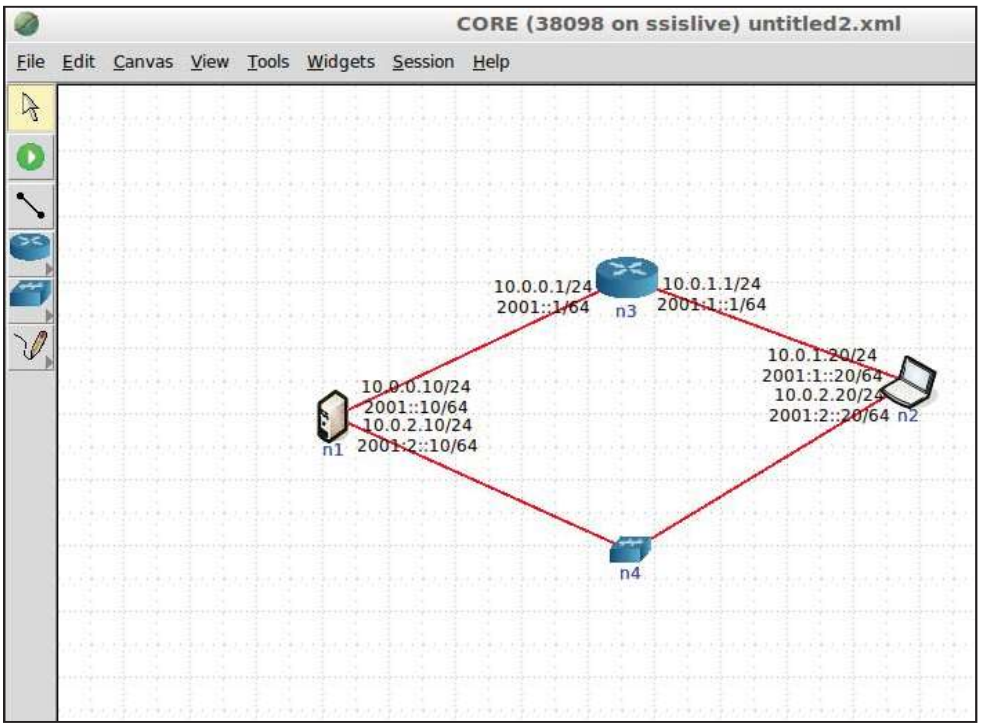

Figure 4: CORE canvas

The XML documents generated by exporting the network topology, adhere to the standard XML structure. In this manner the XML documents form a tree structure that starts with a root element that branches off to leaf elements, with parent and child relationships captured by each branch. The data is classified or named using element tags.

The IMN file format comes from IMUNES, which was an open source project from the University of Zagreb as a custom project within the Boeing Research and Technology's Network Technology research group in 2004 (IMUNES, 2015). The format of the IMN file contains a Tcl lists of nodes, links and their properties. The tabs and spacing in the topology files serve a vital role. These files begin by listing every node, followed by the links, annotations, canvasses, and then the associated options. Each entity within the file is captured in a *block* of braces.

Figure 5 illustrates the difference between the two formats. The content of the files represents the network topology as seen in Figure 4. The left panel illustrates the XML file structure, and the right panel illustrates the IMN structure.

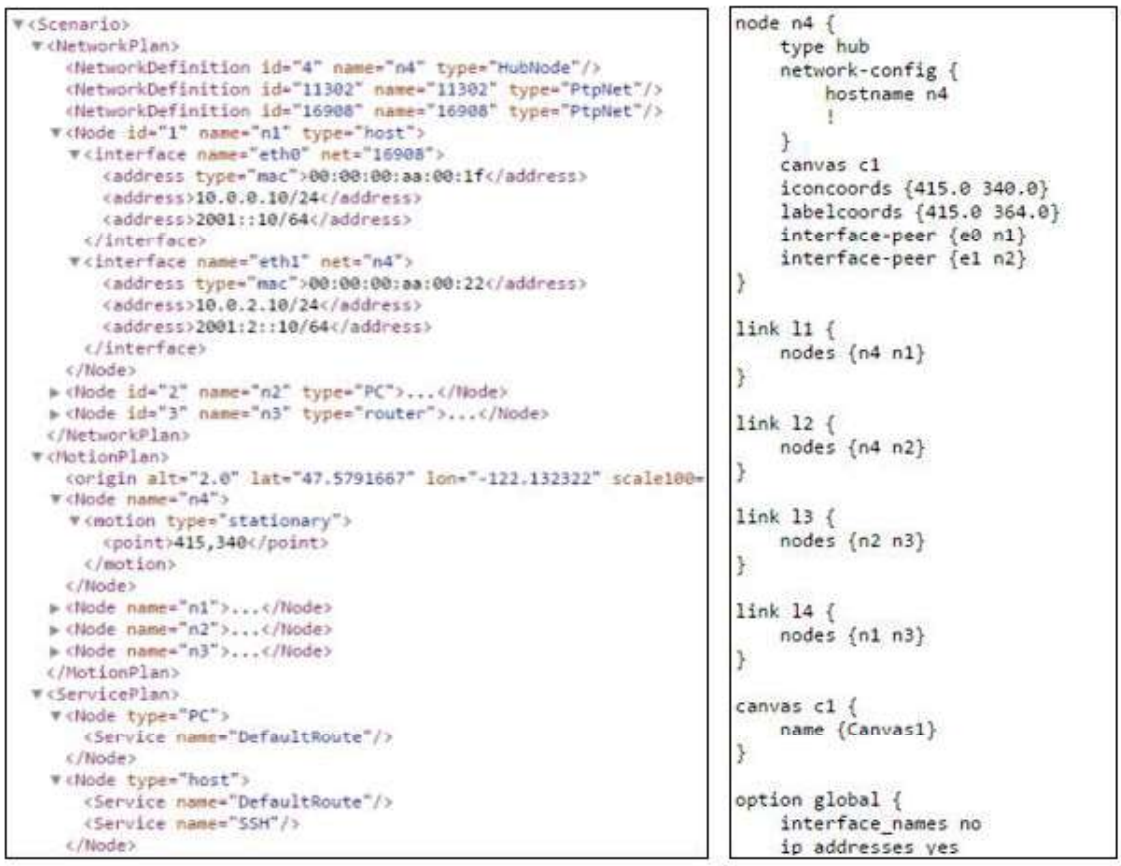

Figure 5: CORE — XML vs IMN formats

4.3 Transforming in semantic data

The lack of constructs in the IMN file, meant that it would be rather complex to generate relations between the contained network items. With regards to XML, there has been previous research proposed on the transformation of XML file structures into semantic data, using OWL (Matheus and Ulicny, 2007), (Yahia, Mokhtar, and Ahmed, 2012) (Barakat, Tan, and Tarasov, 2015). For these reasons CORE's XML was used as the source input to be transformed into semantic data.

Fortunately, XML documents were intended to easily exchange data in a structured manner, supporting interoperability and information exchange between systems and simulations. The similarities to OWL files, which are based on a structured manner to support information exchange, can be easily noticed. As a result it is

possible to use XSLT to transform data between these two structures. The figure below provides an overview of how the process works with regards to XSLT.

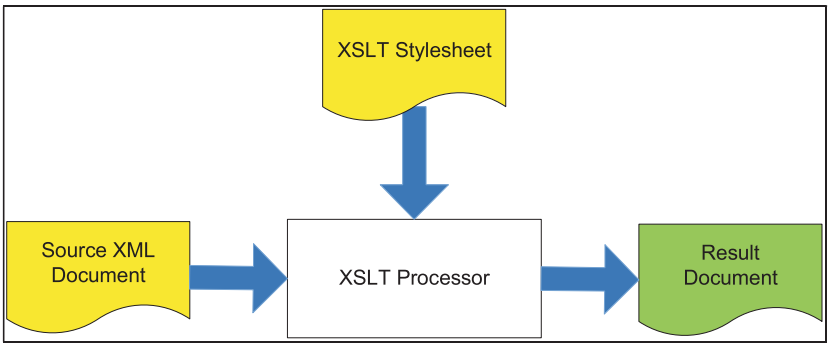

Figure 6: XLST process

4.4 Transformation structure

The following is a list and description of the data and object properties that were defined for the transformation:

Data Properties

- *hasID*: a data property that is generated by CORE and provides each node or entity with a unique ID.
- *hasIPAddress*: a data property that provides a node with a set of IP address values.
- *hasEthernet*: a data property that provides a node with Ethernet interface values.
- *hasMacAddress*: a data property that provides a node with a set of MAC addresses.

Object Properties

- *hasService*: an object property that defines that a node or entity has a relation to some sort of network service.
- *hasType*: an object property that defines the type of network entity, for example PC or a network switch.
- *onNetwork*: an object property that defines the relation of a node and the network that it resides on.

The transformed OWL file was loaded in Protégé (Figure 7), an ontology modelling tool developed by the University of Stanford, to illustrate that the file is recognisable and usable.

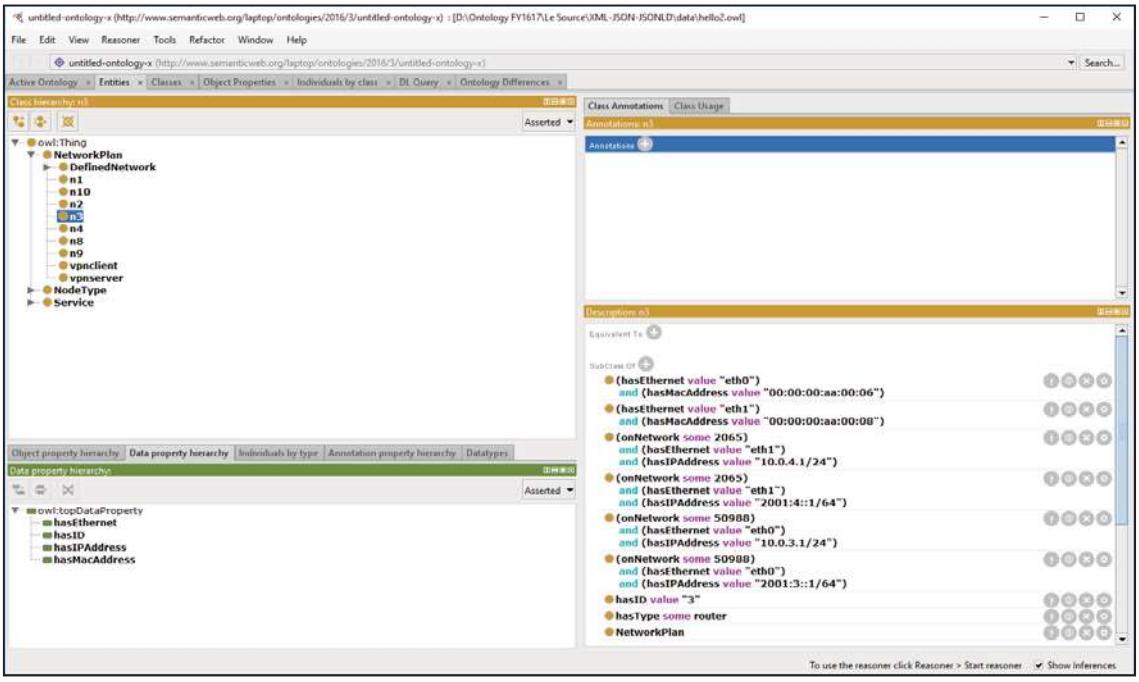

Figure 7: Transformed data in Protégé

With regards to the transformed file, additional nodes were introduced to make the data more complex. The classes are shown on the left panel, with each node's data and object relation shown in the lower right panel.

4.5 Merging ontologies

Now that the network topology is in an ontology format, it can be linked to existing ontologies. Merging ontologies or ontology models can be done with the Protégé application, through the use of the OWL API (OWL, 2016). Note that at this stage, merging of ontologies is largely a manual process. The merging process requires a user to have ontology engineering knowledge and a deep understanding of the structures of the two ontologies being merged. This process is labour-intensive and would greatly benefit from a semi or automated approach. This is a known limitation in the merging process that will be investigated in greater detail in future work.

The following figure illustrates a possible merging hierarchy of the transformed data. In this case study, the transformed ontology was merged with the Network Attack Planning (NAP) Ontology in order to infer knowledge on known or previously identified nodes to further enrich the network data. Merging this result with Network Domain Ontologies, can then provider deeper insight.

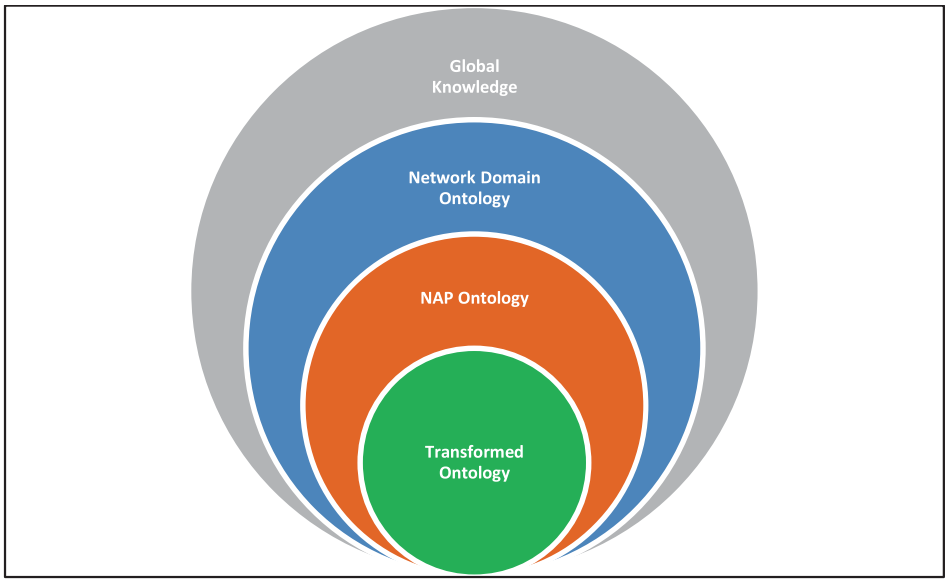

Figure 8: Merging hierarchy

Preliminary results indicate that merging or mapping show that the transformation of network simulation information into semantic-linked data yield enriched information on given subjects, namely network nodes. This can be seen by how the merged ontology model could infer useful information, as well as be queried to retrieve particular information. The merging process, however remains largely a manual intensive task and further investigation to accelerate this process is recommended before it is operationalised.

5. Conclusion and future work

Network simulations provide the ability to test network applications and protocols in a controlled environment, allowing users to observe how changes in configurations affect the network behaviour. This makes network simulations a valuable tool in military network operations.

By using ontologies to enrich network simulation data, it can provide awareness to both the technical operator and to the commanding staff. This was illustrated by transforming network simulation data into semantic data and then linking it to existing knowledge models. A test case, using network topology data from CORE, was conducted to show the viability of such technique. As part of the contribution, the function to perform the transformation directly from CORE was integrated into the application. The transformed ontology was then used in the Protégé ontology modelling tool to show that the transformation was successful. The resulting ontology was then merged with an existing ontology models to explore the inference and querying ability to enrich the information of selected network nodes. One particular challenge that was encountered was the merging process, which remains largely a manual process.

Future work will focus on streamlining this merging process and will explore the ability to allow the generic transformation of network data such that other network simulators can be automatically transformed and merged to better support network attack and defence strategies.

References

Abdoli, F., & Kahani, M. (2009). Ontology-based distributed intrusion detection system, The 14th International Computer Conference (CSICC 2009), pp. 65-70.

Barakat, G., Tan, H., & Tarasov, V.. 2015. Translating XML Models into OWL Ontologies for Interoperability of Simulation Systems. BIS.

Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidemann, J., Helmy, A., Huang, P., McCanne, S., Varadhan, K., Xu, Y. and Yu, H. (2000) 'Advances in network simulation', Computer, 33(5), pp. 59–67. doi: 10.1109/2.841785.

Chan, P.; Theron, J.; van Heerden, R.; and Leenen, L. 2015. An ontological knowledge base for cyber network attack planning. In Iccws 2015-The Proceedings of the 10th International Conference on CyberWarfare and Security, 69. Academic.

Curts, R. J., & Campbell, D. E. (2005). Building an Ontology for Command & Control. Proceedings of the 10th International Command and Control Research and Technology Symposium.

Dupuy, A., et al., "NEST: A Network Sim- ulation and Prototyping Testbed," Comm. ACM, Oct. 1990, pp. 64-74.

IMUNES, 2015. Integrated Multiprotocol Network Emulator/Simulator. [online] Available at: http://imunes.net/. [Accessed on 18 July 2016].

Keshav, S., REAL: A Network Simulator, tech. report 88/472, Univ. California, Berkeley, 1988.

Li, W., & Tian, S. (2010). An ontology-based intrusion alerts correlation system. Expert Systems with Applications, Elsevier, 37(10), 7138–7146. doi:10.1016/j.eswa.2010.03.068

Matheus, C. and Ulicny, B. (2007) 'On the automatic generation of an OWL ontology based on the Joint C3 Information Exchange Data Model', 12th International Command and Control Research and Technology Symposium, Jun, pp. 1–19. Available at: http://ww.w.dodccrp.org/events/12th_ICCRTS/CD/html/papers/149.pdf%5Cnpapers2://publication/uuid/34927E4C-AAE4-4915-83D7-E26642EE7026.

OWL. 2016. *OWL*. [online] Available at: https://www.w3.org/2007/OWL/wiki/OWL_Working_Group. [Accessed 27 October 2016].

Van Heerden, R., Chan, P. and Leenen, L. (2016) 'Using an Ontology for Network Attack Planning', International Journal of Cyber Warfare and Terrorism, 6(3). doi: 10.4018/IJCWT.2016070106.

Syed, Z., Padia, A., Finin, T., Mathews, L. and Joshi, A. (2016) 'UCO : A Unified Cybersecurity Ontology', in AAAI Workshop on Artificial Intelligence for Cyber Security. Available at: http://cps-vo.org/node/26324.

Van Heerden, R., Chan, P. and Leenen, L. (2016) 'Using an Ontology for Network Attack Planning', International Journal of Cyber Warfare and Terrorism, 6(3). doi: 10.4018/IJCWT.2016070106.

W3C, 2000. (RDF) Schema Specification 1.0. [online] Available at: https://www.w3.org/TR/2000/CR-rdf-schema-20000327. [Accessed on 18 July 2016].

W3C, 2014. JSON-LD 1.0 A JSON-based Serialization for Linked Data. [online] Available at: https://www.w3.org/TR/json-ld/. [Accessed on 18 July 2016].

W3C, 2012. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition).

Yahia, N., Mokhtar, S.A., Ahmed, A., 2012. Automatic Generation of OWL Ontology from XML Data Source. International Journal of Computer Science Issues, Volume 9, Issue 2.