DEVELOPMENT OF A CYBER EFFECTS ONTOLOGY FOR USE IN MILITARY SIMULATION

Kent O'Sullivan

Bachelor of Information Technology

A thesis in partial fulfilment of the requirements of the degree of Bachelor of Information Technology (Honours)

UNSW
AUSTRALIA

Australian Centre for Cyber Security
School of Engineering and Information Technology
University of New South Wales Canberra
Australian Defence Force Academy

04 November 2015

PLEASE TYPE

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: O'Sullivan

First name: Kent

Other name/s: Daniel

Abbreviation for degree as given in the University calendar:
BIT (Hons) (Cyber Security)

School: School of Engineering and Information Technology

Faculty: Australian Centre for Cyber Security

Title: Development of a Cyber Effects Ontology for use in Military Simulation

Abstract 350 words maximum: (PLEASE TYPE)

The rise of Information Technology enabled military forces and the subsequent development of cyber warfare capabilities means that our military forces are facing the ever increasing threat of being required to fight in an information-degraded environment in the aftermath of a cyber attack. This ability to *fight through* a cyber attack is directly dependent on the resilience of the digital networks of the attacked force. To build resilience, militaries need to increase their cyber situational awareness and begin preparing for the probable and plausible futures facing their networked systems. Preparedness will grant resilience when the time comes that a deployed force comes under sustained and effective cyber attack. To the end of predicting these futures, this thesis identifies how these future states can be predicted and identified the underlying requirement for a transparent and comprehensive knowledge representation to support this. This thesis will identify a novel method for ontology development towards building a solution to this problem. The proposed Agilitology is the first ontology development approach to apply an agile, usecase-centric development methodology. The Agilitology approach is then used to construct the Cyber Effects Simulation Ontology. The Cyber Effects Simulation Ontology and its two component sub-ontologies: the Cyber Simulation Terrain and the Threat Simulation Ontology are able to effectively, transparently and comprehensively represent cyber effects on military systems in a simulation context. These novel ontologies are applied to a test usecase drawn from a hypothetical military combat scenario to demonstrate the functionality and utility of the Cyber Effects Simulation Ontology before defining future research directions and applications of this work.

FOR OFFICE USE ONLY

Date of completion of requirements for Award:

Originality Statement

‘I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.’

Signed .....

Date .....

Abstract

The rise of Information Technology enabled military forces and the subsequent development of cyber warfare capabilities means that our military forces are facing the ever increasing threat of being required to fight in an information-degraded environment in the aftermath of a cyber attack. This ability to *fight through* a cyber attack is directly dependent on the resilience of the digital networks of the attacked force. To build resilience, militaries need to increase their cyber situational awareness and begin preparing for the probable and plausible futures facing their networked systems. Preparedness will grant resilience when the time comes that a deployed force comes under sustained and effective cyber attack. To the end of predicting these futures, this thesis identifies how these future states can be predicted and identified the underlying requirement for a transparent and comprehensive knowledge representation to support this. This thesis will identify a novel method for ontology development towards building a solution to this problem. The proposed Agilitology is the first ontology development approach to apply an agile, usecase-centric development methodology. The Agilitology approach is then used to construct the Cyber Effects Simulation Ontology. The Cyber Effects Simulation Ontology and its two component sub-ontologies: the Cyber Simulation Terrain and the Threat Simulation Ontology are able to effectively, transparently and comprehensively represent cyber effects on military systems in a simulation context. These novel ontologies are applied to a test usecase drawn from a hypothetical military combat scenario to demonstrate the functionality and utility of the Cyber Effects Simulation Ontology before defining future research directions and applications of this work.

Acknowledgements

Significant gratitude is owed to my thesis supervisor Dr Ben Turnbull for the immense help and support that he has provided throughout the year. His contagious passion for research and excellent use of humour has been inspiring and made the year fly past. May your stockpile of GIFs never empty.

Special thanks also go to Major David Ormrod, who has provided an immense amount of assistance in the development and refinement of my research throughout the year, particularly in the final moments of frantic editing prior to submission.

I would also like to thank Professor Jill Slay and the Australian Centre for Cyber Security for facilitating the study of this degree. I have learned more in a year than I thought humanly possible and it was significantly due to the support provided by the Centre.

Thanks also go to the Directorate of Future Land Warfare and the Australian Army for supporting this study under the Chief of Army's Honours Program.

Thanks to Dr Tim Turner, Dr Gideon Creech and everyone else who encouraged me to undertake further study and helped to shape the conditions to make it possible.

Thanks to the faculty and research students of the Australian Centre for Cyber Security for all of your help and support throughout the year, each of you has helped to influence this work in some way.

Thanks to the other members of the Chief of Army's Honours Program for helping to maintain my sanity throughout the year and not letting me take myself too seriously.

Finally is a significant vote of thanks to my fellow honours students Ben, Jay, Kyle, Matt and Rob for the countless coffee runs and terrible banter that has characterised the last eleven months of our lives. The morale in that office will never again reach the same heights that it achieved this year.

Publications

The following papers were produced in the course of this honours research:

Accepted for Publication

Ormrod, D, Turnbull, B and **O'Sullivan, K** (2015)**.** *System of Systems Cyber Effects Simulation Ontology*. Paper in Press, Accepted for the 2015 Winter Simulation Conference in Huntington Beach, California, United States of America.

Elements of this paper are used through Chapter 2 to contextualise the problems that the Cyber Effects Simulation Ontology is seeking to solve and in section 5.2 of this thesis to characterise the functional requirements of the Cyber Effects Simulation Ontology.

Under Review

O'Sullivan, K and Turnbull, B (2015). *The Cyber Simulation Terrain: Towards an Open Source Cyber Effects Simulation Ontology*. Submitted to the 16th Australian Information Warfare Conference in Perth, Western Australia, Australia.

Elements of this paper are used in section 5.3 to describe the design and development of the Cyber Simulation Terrain as a component of the Cyber Effects Simulation Ontology.

Novel Contributions

This thesis has produced a number of novel contributions to the fields of Cyber Security and Ontological Engineering.

Agilitology

An agile, usecase-centric development methodology for Ontologies.

The Agilitology approach proposed in Chapter Four is the first ontology development methodology to adopt an agile approach. The Agilitology approach significantly reduces the complexity of ontology design, improves the quality of the ontology by focusing work on achieving key usecase requirements and makes the practice of ontology development accessible to novice ontology developers.

The Cyber Simulation Terrain

An open source cyber-terrain ontology towards the modelling of cyber effects at high granularity.

The Cyber Simulation Terrain is an ontological terrain model that effectively represents the underlying network and computing infrastructure of an organisation or system. The Cyber Simulation Terrain improves existing models by facilitating the representation of wireless connections, virtualisation, disks and data as it interacts with a larger networked system. The Cyber Simulation Terrain achieves a more granular approach than has been used in the field before, towards generating more accurate results from analysis of the ontology. The terrain is being released open source to encourage critical review, improvement and continued development.

The Threat Simulation Ontology

An open source Threat Ontology inspired by the Structured Threat Information eXchange (STIX).

The Threat Simulation Ontology is designed to maximise interoperability with existing efforts and standards, adopting elements of a threat-intelligence approach to form a threat ontology capable of effectively representing attacks in the simulation contest and receiving information from a threat intelligence service to enhance the representation of contemporary threats.

Cyber Effects Simulation Ontology

A binding ontology that combines the Cyber Simulation terrain and Threat Simulation Ontology together to represent the effects of Cyber Attacks in a military simulation context.

The Cyber Effects Simulation Ontology is the first ontological framework to employ ontology bridging principles to determine inferentially the effects of a cyber attack on a network while maintaining the representational integrity of its component models.

Open Source Availability of Cyber Effects Simulation Ontology

The source code for the Cyber Effects Simulation Ontology, The Cyber Simulation Terrain and the Threat Simulation Ontology schematic structures is available for public review on the Australian Centre for Cyber Security's GitHub Repository:

---

https://github.com/AustralianCentreforCyberSecurity/Cyber-Effects-Simulation-Ontology

---

Also available are the usecases created for the development of this thesis, the queries and the expected results from those queries to facilitate validation of testing and enable the reproduction of the results of this thesis.

Contents

Table of Figures

Glossary

APT	Advanced Persistent Threat
Bty	Battery. Grouping of three artillery troops
BC	Battery Commander
C2	Command and Control
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance
CAMUS	Mapping Cyber Attacks to Missions and Users
CASCADES	Cyber Attack SCenario And network DEfence Simulator
CMIA	Cyber Mission Impact Assessment
CO	Commanding Officer
CPE	Common Platform Enumeration
CQ	Competency Question
CTI	Cyber Threat Intelligence
CVE	Common Vulnerability or Exposure
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
CybOX	Cyber Observable eXpression
DB	Database
DC	Domain Controller
DIMFUI	Effects Framework: Degrade, Interrupt, Modify, Fabricate, Unauthorised Use, Intercept
DLP	Data Loss Prevention
DS	Design Science
DSRM	Design Science Research Methodology
DVT	Dynamic Virtual Terrain
ECCARS	Event Correlation for Cyber Attack Recognition System
FO	Forward Observer
FuSIA	Future Situation and Impact Awareness
IDS	Intrusion Detection System
INFERD	INformation Fusion Engine for Real-time Decision-making
IODEF	Incident – Object Description Exchange Format
IP	Internet Protocol
ISTAR	Intelligence, Surveillance, Target Acquisition and Recconissance
JFCC	Joint Fires Communication Centre
JFT	Joint Fires Team
JSON	JavaScript Object Notation
LAN	Local Area Network
MAC	Media Access Control

MTNDM	Moving Target Network Defence Measure
NIC	Network Interface Controller (Alt. Network Interface Card)
OWL	Web Ontology Language
OS	Offensive Support (e.g. Artillery Fires)
Portkey	Concept for Bridging between two ontologies through a common object
RDF	Resource Description Framework
S6	Communications Officer
SPARQL	Sparql Protocol And Rdf Query Language
SSID	Service Set IDentification
STIX	Structured Threat Information eXpression
TANDI	Threat Assessment of Network Data and Information
TAXII	Trusted Automated eXchange of Indicator Information
TDD	Test Driven Development
TP	Troop (Abbrev.) – A Collection of Soldiers
TTL	RDF Turtle Syntax
TTP	Tactic, Technique or Procedure
VM	Virtual Machine
VPN	Virtual Private Network
VT	Virtual Terrain
VT.2	Virtual Terrain version 2
VTAC	Virtual Terrain Assisted Impact Assessment
WAN	Wide Area Network
WAP	Wireless Access Point
WEP	Wired Equivalent Privacy
WPA	Wi-Fi Protected Access 2
XML	eXtensible Markup Language

Chapter 1 – Introduction

1.1 – Introduction

On the modern battlefield, the integrity of the networked systems that support operations is paramount. A commander who requires precision fires to support mission objectives relies heavily on their networked Command, Control, Communications, Computing, Intelligence, Surveillance and Reconnaissance (C4ISR) capabilities to deliver these effects. Compromise of C4ISR systems by an opposing force will deny a commander free use of their capabilities and lead to possible mission failure. Understanding the interrelation of these systems with the physical and human systems of the battlespace is essential to developing an understanding of their potential vulnerabilities. Understanding vulnerabilities is the first step towards developing resilience strategies. The effect of a cyber attack is nuanced, and its impact on assets and capabilities is often not immediately apparent against mission objectives.

This work seeks to investigate a means of enhancing the resilience of military forces in the future as they increasingly rely on their networked C4ISR assets. Enhancing resilience means planning for how to fight through a severe and sustained cyber attack that may lead to a state of total information degradation [5].

The contribution of this thesis towards developing the cyber resilience of military forces is work that moves towards increasing the reliability of predictive the modelling that is used to enumerate plausible and probable future states of military C4ISR networks under cyber attack. This thesis will examine methods to increase the transparency, reliability, comprehensiveness and communicability of the knowledge structures that underlie these predictive simulations, identifying and rectifying gaps as they emerge in the course of the research. Conduct of a literature review towards identifying suitable current approaches to knowledge representation in a military simulation context identified a clear gap in research that defined the work of this thesis. The underlying approach of this thesis is a design-science problem-centred approach that seeks to identify a technical solution to the identified gap towards addressing the problem of resilience crossing the domains of land combat, cyber security and technology.

This thesis identifies some deficiencies in the existing work and will examine opportunities to enhance existing research and development methodologies. The aims of this approach are to make knowledge representation more accessible, identify a suitable knowledge structure to support the representation of cyber effects in a military simulation context and work towards developing an implementation of this knowledge structure. By the conclusion of this thesis, a new development methodology for the development of ontologies based on the principles of agile software development will be defined. An ontology to represent the effects of cyber attacks on military networks is proposed and the designed solution will be implemented and evaluated in a realistic scenario to determine if it is fit-for-purpose. Finally, future research directions will be defined.

1.2 – Motivation

Two key motives drive this thesis. The first is the significant threat facing military forces in the highly connected, cyber-enabled future operating environment and the desire to do something to assist in the preparation for this future. The second is as an opportunity to integrate with existing work in a related area, primarily by leveraging some of the shared concepts between this work and the Ph.D. thesis of David Ormrod. The direction of which helped to define and shape the problem statement and the need for this work.

1.3 - Research Questions

Research Question:

How can we effectively develop a method of representing cyber effects on military systems in a simulated environment that promotes transparency, comprehensiveness and understanding?

Subquestion 1:

What is the current state of knowledge representation for the field of cyber security? Is it comprehensive and does it promote transparency of representation? Does it allow the effective modelling of cyber effects in a military simulation context?

Subquestion 2:

What are the areas, fields and disciplines that a comprehensive, effects-focused knowledge structure should encompass or represent?

Subquestion 3:

What is a suitable agile, usecase-centric development methodology that can be used to develop an appropriate knowledge structure?

Subquestion 4:

What is an appropriate knowledge structure to support the representation of cyber effects in a simulation context?

Subquestion 5:

How are the results of the knowledge representation validated; what is a suitable collection of relevant usecases to facilitate this evaluation?

1.4 - Thesis Structure

This thesis is organised in the following manner: Chapter two is a review of the relevant literature considered during this thesis. Chapter three will discuss research methodology for addressing the research questions and survey relevant development approaches feasible for use in implementing the solution. Chapters four and five will detail the novel work of this thesis. Chapter four will describe a new, agile ontology development methodology. Chapter five will detail the creation of a new knowledge representation for the purpose of

representing the effects of cyber attacks in a military simulation context. Chapter 6 will define future research directions and conclude.

Chapter 2 – Literature Review

At the edge of chaos, unexpected outcomes occur. The risk to survival is severe."

- Michael Crichton, 1995

2.1 – Background

2.1.1 – The Future Operating Environment

Determining the most effective method to represent cyber effects in a military simulation context requires a thorough understanding of what that context is. The first section of this literature review will address the major trends and influences that are driving the imperative to answer this research question. In their 2014 futures document, 'Forward: 2035' [6] Australia's Defence Science and Technology Organisation (DSTO) assert that:

"Society is increasingly operating in cyberspace to connect with, deliver and access services, to obtain information and to perform transactions. This is enabled by traditional computer networks, with a growing reliance on wireless transmissions. Similarly, digital media is allowing people to form new connections and selectively access information through multiple channels with subsequent erosion of trust in traditional sources."

This assertion raises some compelling questions about the emerging vulnerabilities and threats into the future. In the civilian context, risks to critical infrastructure, communications channels, financial institutions, privacy and personal security are just a selection of these challenges. The rapid monetization of computers and the internet over the last decade have fuelled cyber and electronic crime rates, with the estimated cost of cyber crime in 2014 approaching 575 billion dollars [7]. The rise of the 'Advanced Persistent Threat' (APT) has seen cybercrime become increasingly organised, sophisticated and dangerous. Kaspersky Labs maintains an online logbook of APT campaigns and actors [8] that clearly illustrates the scale of the problem. High Profile attacks like APT1 [9], the theft of the plans for the Ben Chifley Building [10], Stuxnet [11], Crouching Yeti [12], Desert Falcons [13], the OPM Data Breach [14], the Sony Hack [15] and attacks targeting the Free Syrian Army [16] have between them compromised significant amounts of sensitive commercial, industrial and national security information [17]. These attacks have highlighted the extensive vulnerabilities present in systems that are encountered every day, and the quantity of criminal attackers willing to exploit them.

The Australian Army's Future Land Warfare Report describes the future operating environment as an increasingly lethal, crowded, collective, connected and constrained environments [18] created by the identified technological [19] social, scientific and cultural trends [20] shaping the world. Environments of the future will require that conflict is "*waged by information technology enabled forces in land, sea, air, space and cyberspace*" [21] and will *"use modern information technology to link sensors, weapons systems, commanders and their personnel in a networked environment*" [22]. The rise of cyber-enabled forces exploiting their C4ISR

capabilities provide an additional, non-kinetic target for hostile forces. It is likely future wars will involve the conduct of cyber attacks independently and as part of a broader spectrum of conflict. The increasing prevalence of cyber attacks and effects into the future dictates the need for commanders to become familiar with attacks, the conduct of attacks and the effects that are generated by an attack.

2.1.2 – Cyber Attacks and Effects.

The research question posed by this thesis requires the effective representation of cyber effects in the context of a military simulation. Creating a high-fidelity representation of cyber effects requires a detailed examination of what constitutes a cyber effect, the delivery vectors, and effect capabilities. The United States Cyber Warfare Lexicon [23] cautions that "*Describing the effect(s) of using a cyber weapon is not nearly so straightforward as determining its functions, since its effects depend not only on whether or not the developer's environmental assumptions and expectations have been properly characterized and satisfied by the operational environment, but also how well the target itself has been characterized"* Two of the core characteristics of a *cyber effect* are their *collateral* and *cascading* nature [24]. That is, cyber attacks are not easily controlled or bounded, making targeting a difficult task.

Cyber effects will typically result from a cyber attack. The character of a cyber attack varies immensely dependent on context and objective. Most sophisticated, multistage attacks follow a similar structure. Analyses of these structures are available in the APT 1 report by Mandiant [9], the InfoSec Institute [25], Deloitte [26], SANS [27] and Mitre [28]. The core idea of an Access-Assure-Leverage-Pillage cycle is consistent across most of the described attack processes. The Mandiant attack process in Figure 1 was defined in their APT1 report [9] and will be the attack methodology referenced throughout this thesis.

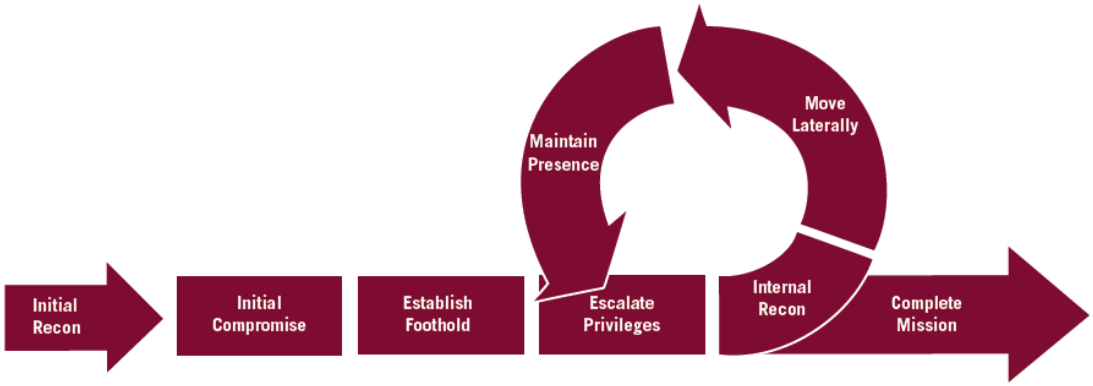

Figure 1 - The Mandiant Attack Cycle

Most attacks will follow an attack cycle analogous to this one to infiltrate targeted systems and deliver effects (whether intentionally or unintentionally). Understanding the possible effects that a cyber attack can have on a system is crucial to effectively representing them. The United States Strategic Command's Cyber Warfare Lexicon Document [23] lists six effects available to military cyber operation planners:

- **Denial** - *Degrade, disrupt or destroy access to, operation, quality of service, or availability of target resources, processes and/or data.*

- **Manipulation -** *Manipulate, falsify or distort trusted information on a target.*
- **Command and Control** - *Provide operator control of deployed capabilities.*
- **Information / Data Collection** - *Obtain targeting information about targets or target environments.*
- **Access -** *establish unauthorised access to a target.*
- **Enabling** - *Provide resources or create conditions that support the use of other capabilities.*

The list is brief and imprecise. Though considerable attention goes toward explaining the component elements of denial, there is limited information available to explain the remaining effects [23]. Traditionally, the impact of effects in the physical world is easily observed. For example, an observer can witness the physical destruction of something. Ease of assessment is not present in the cyber domain. The Cyber Lexicon provides an interesting discussion of context and need for an effective battle-damage-assessment capability. This battle-damage-assessment capability would be used to gauge the success of a given cyber attack in generating the intended effect. The conduct of effective Cyber Battle Damage Assessment is outside the scope of this thesis though it is linked and is an area that will be able to build on the work of this thesis.

Attack Category	Effect on Process	Effect on Information
Degradation	Speed of process is slowed by some multiple	Rate of information delivery is decreased; quality or precision of information produced by an activity is decreased
Interruption	Process is unavailable for some time period and will not commence until the incident is recovered	Information is unavailable for some time period
Modification	Process characteristics have been altered in a way that can affect the output/result of the process	Information has been altered, meaning that the processes that use it may fail, or produce incorrect results
Fabrication	A false mission instance has been inserted into the system, which may interfere with real mission instances	False information has been entered into the system
Interception	The process (perhaps software, perhaps embodied in hardware) has been captured by the attacker	Information has been captured by the attacker
Unauthorized use	Raises the potential for future effects, or unexpected outcomes on processes	Raises the potential for future effects on information

Figure 2 - MITRE Classification of Cyber Effects

Related work into the characterisation of Cyber Effects by the MITRE [29-31] proposes a different framework of effects. These effects link to the traditional principles of information security of *Confidentiality Integrity* and

Availability [32]. Mitre's proposed framework of cyber effects is summarised in Figure 2 *[29]*

The authors of the paper use the acronym *'DIMFUI'* to group the effects together. *Degradation* and *interruption* link to the *Availability* principle. *Interception* links to *confidentiality*, *modification* and *fabrication* are related to the *integrity* of system data. *Unauthorised use* refers to the ability to conduct unauthorised operations on a system or access data without adequate permissions. Additional information security principles like *non-repudiation, authenticity* and *identification* [32] are either encompassed in these six elements of the effects framework or not something that can be statefully modelled, for example, *non-repudiation*.

These effects represent the capabilities available to commanders. Planners and decision makers on both sides of a conflict must have a deep understanding of the effects available to them, how they can assess their vulnerabilities and planning their defences.

2.2 – Cyber Dependence and Risk

Dependence is defined as the state of relying on or being controlled by someone or something else. Cyber-dependence is the reliance of a military force on its networked C4ISR networks to operate effectively. Communications capabilities are critical enablers for all facets of the military operation. The risk posed by rising dependence on networked communications to operate in the context of the global development of cyber warfare requires close monitoring.

The rise of networked military forces and cyber-enabled conflict has bought a collection of benefits and challenges. The networking is beneficial as it offers unparalleled command and control over forces. Cyber attacks give the attacker to the capability to remotely engage targets for effect. The dependence paradox balances between embracing networked technology and the improvements and benefits it brings and retaining the ability to operate without external support. It is vital to utilise these networks to facilitate improved command and control on the battlefields of the future. To fall behind in the development of C4ISR capabilities is to be at a major disadvantage in the future operating environment. If hostile forces can cripple the C4ISR networks that support the operational capacity of their forces they will severely disrupt the command and control capacity and feasibly significantly weaken the force cohesion of the targets of the cyber attack [33].

The United States Army assert that their network infrastructure must be *"survivable, resilient, redundant, and reliable to enable continuity of operations and disaster recovery in the presence of attack, failure, accident, and natural or man-made disaster"* [34]. Resilience is not achievable solely by attempts to prevent intrusions at the edges of the network. While there is a substantial benefit to security from current work on antivirus, advanced Intrusion Detection Systems [35-39], commercial endpoint protection [40-44], and protective policy frameworks [45].

Goldman et al argue that *"The current philosophy of trying to keep the adversaries out, or the assumption that they will be detected if they get through the first line of defense, is no longer valid. Given the sophistication, adaptiveness, and persistence of cyber threats, we can no longer assume that we can completely defend against*

intruders and must change our mindset to assume some degree of adversary success and be prepared to "fight through" cyber attacks to ensure mission success even in a degraded or contested environment" [46].

The United States Air Force extend this approach and actively advocate for the need to contextualise the impact of cyber attack from a mission perspective, stating: "*The time has come to think of cyberspace in a new light; not only must we defend against any attack, we must be able to "fight through" any attack, accomplish our missions and retain the ability to respond–thus giving us mission assurance in the face of future attacks or other disruptions* [47]. '*Fighting through'* is organisational resilience in a military context. Resilient combat forces must be able to continue to fight and win in a 'degraded information environment' where trust in cyber systems is partially or wholly compromised. In addition to training soldiers how to operate with the total loss of communications as suggested by Scott [5] resilience plans should cover a spectrum of scenarios including operating in scenarios of reduced system trust, suspected compromise of systems or partial degradation of capabilities.

2.3 – Cyber Situational Awareness and Resilience

Endsley defines Situational Awareness as *"the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future"* [48]. Research into cyber situational awareness expands this definition, asserting that it is a three stage process encompassing the *recognition, comprehension* and *projection* of a given situation [49]. To achieve situational awareness, the decision maker needs to address the seven elements of situational awareness. Barford et.al define these seven elements as the awareness of the *current state*, the *impact* that an attack will have, the *evolutions* of a situation, the *behaviour of adversaries*, *why and how* the current situation was achieved, the *quality of information* available to the decision-maker and the *development of plausible futures* for the current situation.

Goldman asserts that the characteristics of a resilient system are: *Protection* and *Deterrence*, *Detection* and *Monitoring*, *Constraining* and *Isolating*, *Maintenance* and *Recovery*, and *Adaption* [50]. *Human resilience* is equally important. An approach to achieving resilience is predicting potential threats and planning for how to avoid, counter or mitigate them. A prepared decision maker is a resilient one, able to draw on their experiences, prediction and planned reactions to response efficiently and effectively to a wider set of situations than an unprepared person in the same context.

2.3.1 – Commercial Approaches to Achieving Resilience

Internationally, commercial entities like Symantec [51] and RSA [52] are working towards making systems more resilient. These two companies are working on developing technological solutions to ensure Data Loss Prevention (DLP). DLP aims to make systems more resilient through *constraint, recoverability* and *isolation* of critical data. Extensive development of *Cyber Threat Intelligence* (CTI) services [53] is also underway. CTI aims to promote the *protection, deterrence, detection, monitoring* and *adaption* of linked systems by fusing large amounts of disparate knowledge to enhance Cyber-Situational-Awareness and 'move down the kill chain'. The threat intelligence products are primarily commercial solutions like the one provided by FireEye [54], or the research being conducted by organisations like Lockheed Martin [55], Deloitte [26] and Mitre [43, 56]. The

majority of these products focus on the concept of the Cyber-Kill-Chain proposed by Lockheed-Martin in 2011 [57] This cyber-kill-chain is shown in Figure 3.

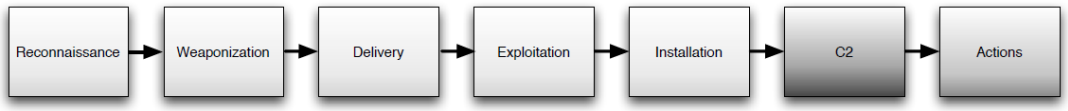

Figure 3 - Cyber Intrusion Kill-Chain

The paper by Lockheed Martin theorised that there is a chain of linked events that lead up to a cyber attack at the point of compromise and continue through the initial period of exploitation. The premise of threat intelligence services is that by identifying these patterns and actively detecting them an organisation employing CTI can move 'down the kill chain' and detect imminent attacks based on earlier 'links' in the chain of events. These recurring patterns and data points on the kill chain are called *Cyber-Observables*. Open-source implementations of this approach include the Cyber Observable eXpressions (CybOX) [58], the Structured Threat Information eXpression (STIX) [59], openIOC [60] and the Incident-Object Description Exchange Format (IODEF) [61].

In addition to DLP and CTI approaches, most major organisations will have Incident Response (IR) strategies. By identifying possible weaknesses, predicting problems arising from these weaknesses and identifying appropriate responses, the organisation becomes increasingly prepared to respond to a crisis and, therefore, will be more resilient if a real crisis occurs. The NIST [62] Incident Response Model is broadly accepted as an industry standard framework for incident response. Recent research has begun to explore evolutionary incident response techniques that aim to develop robust and integrated incident response methodologies [63] with the intent of significantly improve the practices of *recovery* and *adaption.*

The technological and methodological solutions to increasing cyber resilience through DLP, CTI and IR all make valuable contributions towards developing resilience. Furthering the development of resilience requires an additional step - future planning. Methodologies for manual war gaming and red-teaming exist and are used extensively to help decision makers prepare for the future. Both wargaming and red-teaming processes follow an *Action, Reaction, Counter-Action* process to attempt to predict the behaviours of hostile forces and answer *What-If* questions about the future based on these behaviours or actions. To effectively achieve the aims of wargaming or red-teaming, the outputs must be accurate and detailed enough to enhance the decision-making processes that they are informing. A prerequisite of the accurate output of information is the accurate input of data. Structuring, verifying and validating this information have been a topic of research for some years.

2.3.2 – Simulation

2.3.2.1 – Simulation Background

To automate war gaming, red teaming and other predictive activities analysts employ simulations to perform the calculative aspects of the modelling and reduce the inaccuracy of human calculation. Simulation is the process

of specifying, implementing and executing a model over time. A simulation is a model observed over time. A model is a representation of some or all of a device, system or object. [64]

2.3.2.2 – Related work Utilising Simulation to Enhance Cyber Situational Awareness and Resilience

Current systems in the public domain do not provide a robust, transparent and comprehensive cyber simulation model that can effectively map cyber effects to military networks. Research is continuing in this area. The work of Musman et al. on computing the effects of cyber attacks onto missions [31] and complex missions [29] towards developing a Cyber Mission Impact Assessment (CMIA) framework [30] has a meritorious aim but focuses too heavily on the mission impact at the cost of a robust cyber infrastructure model – representing effects as binary attributes. Similarly the work of Machado, Barreto and designing a cyber defence simulator in 2013 [65] extended by Machado and Yano in 2014 [66] towards attaining cyber situational awareness focuses heavily on consequences to the *availability* of nodes. This narrow perspective misses the significance of confidentiality and integrity and hence is unsuitable.

Barros-Barreto, Costa and Yano's work in 2012 towards mapping cyber-physical effects [67] focuses heavily on the impact of mission business process rather on the fidelity represented effects themselves when mapped to a network. The Cyber-ARGUS framework forwarded by Barros-Barreto, Costa and Hieb in 2014 [68] builds on the research into cyber-physical interaction. It utilises impact dependency graphs taken from the work of Jakobson [69-71] with the intent of developing a framework to simulate the effects of cyber attacks on Command and Control and battlefield situational awareness [68]. The ARGUS framework holds merit as a potential methodology to follow for developing a cyber effects simulation though the detail it represents regarding the actual attacks conducted needs to be more granular to be useful.

Projects such as The Advanced Research Defence Agency's (ARDA) *INformation Fusion Engine for Real-time Decision-making* (INFERD) [72] and *Event Correlation for Cyber Attack Recognition System* (ECCARS) used an information fusion approach to address the problem of real-time situational awareness. The ECCARS project aimed to solve the problem of sustainably performing real-time detection of complex, multistage, coordinated cyber with minimal a priori settings and is sustainable across a monitored environment [73].

These approaches were evolved by the Rochester Institute of Technology's (RIT) *Threat Assessment of Network Data and Information* (TANDI) Holsopple, et al. [74]. The TANDI project focused on correlating Intrusion Detection System (IDS) alerts to determine the *How, When, Where* and *What* of a cyber attack and use this information to project '*what-next*' hypotheses for network stats to predict likely attack paths. TANDI made the design choice to split the *How, Where* and *What* into three separate models. These were the attack sequence model, the logical topology and the information graph

The design TANDI of inadvertently made a significant contribution to the field of cyber security knowledge representation. TANDI was the first project to separate the generation of a logical network topology from the other elements of the model. Separation of the component models allows the logical topology to be generated independently – reducing the interference from external influences and creating a realistic depiction of a network. This significance of this is addressed in detail in Section 2.6 of this thesis.

The *Virtual Terrain Assisted Impact Assessment* (VTAC) approach uses data points about network missions and nodal criticality to simulate the impact of a cyber attack on a given network node. The cyber situational awareness project *Future Situation and Impact Awareness* (FuSIA) [75], an evolution of the TANDI project, added nodal 'states' that aimed to simplify the generation of plausible futures by explicitly designating the objects of interest with states such as 'Normal' or 'Compromised'. *The Automatic Mapping of Cyber Assets to Missions and Users* (CAMUS) [76, 77] project is an evolution of FuSIA with additional attention given to the semantic links between users and cyber assets towards supporting mission success. It attempts to use information fusion techniques to automate the mapping – addressing one of FuSIAs noted shortcomings. CAMUS explicitly states the value of an established ontology and introduces the idea of using an external RDF store to address issues of scalability that the developers identify in the FuSIA and VTAC approaches. In 2009, the work into the *High-Level Information Fusion for Tracking and Projection of Multistage Cyber Attacks* [78] saw a revival of TANDI and INFERD. This application attempted to fuse information produced by INFERD from correlated IDS alerts and use TANDI to project likely futures based on that information. Though it made notable improvements in the efficiency of information fusion over its component models it still faced many of the same limitations and additionally its heavy focus on the correlation of IDS alerts to possible intrusions, makes its scope too narrow to be appropriate to address the problem of representing cyber effects in a military simulation context.

The design of the *CyberSim Modular Cyber Attack Simulator* [79] and by extension the *Multistage Attack Scenario Simulator* (MASS) [80] were further evolutions of these projects. The MASS' intended use is the creation of datasets for training and evaluating cyber situational awareness tools. The next iteration of the MASS is the *Cyber Attack SCenario And network DEfense Simulator* (CASCADES) [81, 82]. CASCADES is designed to be a controlling core for multiple dispersed context models, a federating framework that provides flexibility and extensibility to facilitate the effective simulation of dynamic networks.

A core theme underlying all of these existing efforts to simulate or generate cyber situational awareness is the use of a simplified world model to underlie the simulation or tracking efforts. The process of abstracting a problem to make a simplified representation of the real world is a common activity in modelling and simulation. The problem with the abstraction processes in this context it that the reviewed efforts towards modelling cyber attacks and generating situational awareness lack transparency. The opacity of their design obfuscates their underlying world models and knowledge representations. Private schemas prevent a detailed analysis of the simulation tool from occurring and make the results of these simulations difficult to verify. Automated methods have many of the same problems as manual tools of inaccuracy, opaque input as a manual war game. The next section of this thesis will address the importance of this transparency for improving the accuracy and results of these simulation efforts.

2.4 – Knowledge Representation

Ultimately a simulation is a reasoning and prediction process produced by running predictive algorithms over an established knowledge structure. The fundamental truth of knowledge representation and modelling is that the

represented knowledge is a surrogate for the real world. The two key considerations of designing this surrogate are 'what is the surrogate for?' and 'how close does the surrogate have to be to the real thing?' [83] When representing cyber attacks and effects, simulations need to be as close as possible to real life. Any process of abstraction, simplification and representation fundamentally changes the nature of the system under observation and alters its behaviour. Perfect representation to achieve complete fidelity is impossible. Designers of knowledge representations must understand that representations are imperfect and despite how meticulous the design and implementation process is, it will, by its nature, contain imperfections. These imperfections mean that eventually all simulations based on this representation will eventually reach incorrect conclusions. The smallest imperfection in one aspect of the knowledge representation or the smallest change to facilitate modelling can result in a significant variance to the expected outputs, particularly when representing large, complex systems [83].

Understanding the complexities of these systems and minimising the number of imperfections requires clear identification of the entities, properties and relationships of the represented systems. Application of a network representation approach to representing the systems can achieve the necessary clarity. Philosophically, networks are simply a collection of concepts with defined interrelations [84]. Complex systems are collections of entities that interact. A network approach to understanding complexity is a natural fit. The benefit of applying a defined network structure to deconstruct the complexity is that it formalises the representation and enables the analyst to validate the Emergent Phenomena observed on the network. Emergent Phenomena are collective behaviours observed in networks akin to second and third order effects. These phenomena offer the best window of insight into possible future states [84].

The emergent phenomena of a network are highly dependent on the underlying network structure [84]. Simulation is essentially the automated process of generating and observing these phenomena. Therefore, the fidelity of the underlying representation is crucial. The paradox of this situation is that changes to the network structure will alter the phenomena produced by it. Simulations must be capable of reliably generating emergent phenomena to enumerate the plausible and probable futures available to the system. If even the smallest imperfection misrepresents the system it could significantly change the results. This problem was first identified by the Theory of Deterministic Nonperiodic Flow. This Theory asserts that *'...a system whose future state we desire to predict* [initialized from] *two states differing by imperceptible amounts may eventually evolve into two considerably differing states"*[85]. The implication of this discovery is that *"If, then, there is any error whatsoever in observing the present state-and in any real system such errors seem inevitable-an acceptable prediction of an instantaneous state in the distant future may well be impossible"[85]*. Increasing complexity of systems and attempts to predict further into the future amplify the scale of the variance in results. The fundamental implication of this theory (known colloquially as *the butterfly effect)* is that tiny imperfections from the abstraction process used to generate a knowledge representation can significantly impact on the output of the simulation potentially rendering any results completely invalid.

There are some negative implications for the reliable use of simulation as a tool for enumerating potential future towards developing the knowledge necessary to inform decision makers for resilience planning. For a simulation

to be reliably employed by the analysts developing it, there is a requirement that to be able to observe every element of the initial conditions to validate comparatively the results of the simulation against the initial system conditions [64, 86]. Managing uncertainty in these systems through this validation approach is the reason for the compelling need for *transparency* in the effective representation of cyber effects in a military simulation context. Transparency of the underlying structure of the represented system enables analysts to validate the output of the simulation against the known initial conditions. Being able to control and validate these conditions is essential to achieving accuracy and managing misrepresentation and inaccuracy in representation.

Computer networks and cyber attacks interact with the battlefield and decision makers as a collection of inextricably linked, complex systems of systems. The complexity of the systems, issues of system to decision maker trust, the inherent uncertainty of complex interactions and the resulting emergent phenomena intersect to form a wicked problem for decision makers [87] and makes the enumeration of futures exceedingly difficuly. Accounting for the dependence between the underlying network structure and the emergent phenomena [84] requires a formal definition of the modelled network (In this context - the complex systems of cyber-physical-cognitive interaction and their emergent phenomena on the battlefield). A popular method for the formalisation of networks of knowledge and facts is through the use of an ontology. Ontologies have emerged as a major force in the realm of knowledge representation and intelligent reasoning. They have some advantages over more traditional means of knowledge representation and analytics like *Expert Systems*, *Object Oriented* and *Taxonomic* approaches.

In comparison to *Expert Systems,* Ontological structures are far more likely to produce reliable and verifiable inferences about the represented knowledge. Ontologies are a knowledge-based approach that focuses on the feasibility of paths and consequences [88]. Expert Systems, rather than being grounded in semantic links between concepts tend to be constructed to make guesses that a human expert would make. These guesses are typically based on a *modus ponens / modus tollens* structure and often have no foundation in the underlying model or knowledge store, relying heavily on the opinion and experience of the designers [83]. This subjectivity in development and disconnection from representation and reasoning mean that Expert Systems are substantively less reliable for achieving a true representation of the world than an equivalent Ontological structure with defined semantics that can make inferences and deductions based on the interrelationship of concepts.

Object-oriented approaches to the problem of knowledge representation are judged unsuitable for two primary reasons. First, implicit in adopting an object-oriented approach is the difficulty in separating the form of the ontology from its content [89]. The form of an ontology refers to the schematic structure of the knowledge and the relations between concepts prior to instantiation. The Content of an ontology is what inferential reasoning and analysis is conducted on. It is the part of the ontology that is used to generate the phenomena required by the designers. The requirement to separate these two ontological concepts stems from the principle that a knowledge representation is not a data structure [88]. This requirement for the distinction between the form and function of an ontology is the second reason that the object oriented approach is unsuitable.

Taxonomies are precursors to Ontologies. They are a kind of knowledge representation semantically stronger than a *Glossary* but weaker than an *Ontology* [90]. Taxonomic links are of an 'IS-A' format to denote subtypes and define a hierarchical structure of relationships. Though taxonomies are extremely useful in the context of classification and categorization of concepts, they *"Lack the necessary and essential constructs to enable IDS to reason over an instance that is representative of a domain of a computer attack…"* [88]. They are fundamentally a classification system used to arrange concepts into related groups, not define interrelationships of any semantic complexity or meaning. *Taxonomies*, for these reasons, are not suitable for the detailed representation of concepts and relations required by the network structure underlying this simulation.

Each of these three methods of knowledge representation is useful in their context. However, their limitations make them unsuitable for use in the representation of cyber effects in a military simulation context. Therefore, the focus of further examination for this thesis' knowledge structure is that of an Ontology. Onologies offer a number of benefits including extensibility, modularity, semantic strength and machine readability [89].

2.5 – Ontologies

Ontologies are powerful constructs that enable machines to interpret the concepts within a domain of knowledge and the relations between them [89]. The previous discussion on knowledge representation theory extracted the need for a transparent and robust underpinning network model. Undercoffer et al. [89] describe ontologies as highly structured knowledge representation formats that enable explicit definitions of the semantic links between the entities and properties of a given domain of knowledge. Any form of structured knowledge is useful for communicating with humans; this is partly why we find tables, graphs and reports useful. Structure makes ingestion of knowledge easier. A major strength of ontologies is the ubiquity of understanding they enable. Ontologies are both human and machine readable when correctly implemented. The machine readability permits smooth interaction with any system or application that may be utilising an underlying ontological knowledge structure to format their underlying knowledge representation. Machine readability is essential for facilitating the effective conduct of simulation, machine learning algorithms and artificial intelligence systems.

2.5.1– Ontology Definition

An ontology is an explicit specification of a shared conceptualization [91]. It is a unifying framework that unites multiple viewpoints to facilitate problem solving [92]. Their use was popularised in the early 1990's by the artificial intelligence community who wanted to provide a standard reusable framework that would support the development of increasingly large knowledge bases [93].

A 'strong' ontology is judged by its real-world semantics, use of logical axioms and machine readability [90, 94]. Ontologies are built by collecting and integrating numerous subject-predicate-object triples. Each triple should define a single fact within the ontology. The effectiveness of this triple to represent only a single fact reflects the triple's semantic strength. In the context of the big data problems associated with modelling complex systems of systems, a semantically strong ontology will optimise searching; promote automation and efficiency of scale.

There are several key elements and related definitions within the field of ontology design. Nay and McGuinness

provide the most detailed overview [95] and combined with several other key publications of the field [1, 91] encapsulates the key design principles applicable to Ontology development. The common elements, first specified by Gruber, include a number of principles of ontology design that appear in numerous other works and form the basis of ontological design [91]. They are *Clarity, Coherence, Extensibility, Minimal Encoding Bias* and *Minimal Ontological Commitment*. Most of these are self-explanatory and unchallenged in literature. Minimal ontological commitment, however, appears to be fundamentally misunderstood. A comparison of Gruber's assertion to the work by Davis, Shrobe and Szolovits [83] seems to show a contradiction in perspective. The latter work celebrates ontological commitment, seeing it as the most effective manner to establish a knowledge representation that has a controlled set of abstractions (and hence, inherent imperfections) and a clear, coherent worldview. Both papers advocate the need for extensibility in the design of their knowledge representation, Davis' extends this by foregrounding the benefits of a modular approach to the representation design. Given the major similarities of each approach, when Gruber refers to minimal ontological commitment the intent is to select only a single worldview to represent in the ontology. Adding multiple perspectives to the same ontology confuses and dilutes the worldview, forcing the representation to compromise the fidelity of its representation to include fundamentally incompatible elements or attempt to represent simultaneously multiple levels of granularity. Using the modular philosophy outlined by Davis, Shrobe and Szolovits to represent a complex domain with multiple non-orthogonal domains of knowledge would require that the domain is split and represented in discrete ontological structures.

An ontology bridging approach should be used to map related objects across disparate ontologies. Ontology Bridging is not the same as mapping or mapping or translation [96, 97]. Rather, it is an effort to use common objects shared between ontologies to form links between them. To conceptualise this, consider the structure of a Venn diagram. Though the individual elements of a Venn diagram are all linked, they have their individual perspective and distinct meaning. Where the components of a Venn diagram interact there is some common aspect of the representation that is shared between the two differing perspectives of the same object. For example where a threat-focused ontology may see a computer as just an IP address with a list of vulnerabilities, an organisational asset ontology would see the same computer as a collection of software and business functions. Rather than attempting to represent both of the differing perspectives and levels of granularity in the same ontological structure and compromising the integrity of their worldviews, an ontology bridge can be used to link the two and access the elements of interest of each whenever they become relevant. What this means is that ontological commitment is a good thing, provided that it is only to a single worldview per ontology to preserve the consistency of worldviews that compromise the integrity of other ontologies.

2.5.2 – Application of Ontologies

Ontologies are used widely to provide structure and meaning to large, complex collections of data and domains of knowledge where semantic linking between objects is essential to developing a coherent understanding. The most widely used ontology is the ontological model underlying the *Google Knowledge Graph* [98]. This underlying Ontological model has permitted Google to develop highly efficient semantic-link searches that use the mantra of *'things, not strings'*[99] to enable Google search results to suggest semantically similar results to the search in addition to the keyword search. Google uses the metadata that it finds embedded in 20% of the web's content to facilitate this semantic matching [100]. Searching semantic relations is wildly more efficient

than keyword matching searches and provides a richer result set. The discussion by Raskin et. al. highlights, a data structure is not a representation of knowledge, it is the instantiation of that structure [88]. The *Google Knowledge Graph* is the instantiation of their knowledge ontology.

The most significant project in ontological development is the *Cyc* project. *Cyc* aims to create an ontology that can represent the 'common sense' reasoning capabilities of human existence. The *Cyc* project started in the mid-1980's - inspired by the artificial intelligence depicted in science fiction films of the time such as '*2001: A Space Odyssey'*. The reasoning of the creator of Cyc, Douglas Lenat was that to facilitate the interactions seen between the fictional Artificial Intelligence HAL and the human crew it would need to have a deep understanding of all knowledge, conclusions and information that humans take for granted. [101] Cyc was the first step towards quantifying this knowledge. The purpose of Cyc as a high-level ontology is to support advanced artificial intelligence tasks. According to their design documents, these tasks include the understanding of important truths such as "*If you cut a lump of peanut butter in half, each half is also a lump of peanut butter; but if you cut a table in half, neither half is a table."* [102]. Cyc has been continually developed since 1984 amassing ever-increasing amounts of information about the world to facilitate making common sense judgements. It has been used to create several applications including a terrorism knowledge base [103], a clinical query knowledge base [104] openCyc [105], researchCyc [105] and cycSecure – the cyber security implementation of Cyc [106] that is discussed in detail in Section 2.6.1.3.

Freebase, a community driven ontological knowledge structure *for all of the world's information* was conceptualised in 2004 by Daniel Hills [107] and is a significant contributor to the field of ontologies. Freebase's community-driven approach has helped significantly to build a large community of ontology-aware and ontology-advocating designers, developers and users. Freebase continued to grow and develop for some years after its acquisition by Google and integration into the Google knowledge graph in 2010 until 2015 when it announced the discontinuation of the project. There are also numerous examples of ontologies being used to enhance knowledge representation in other fields including, but not limited to law [108], chemistry [109], information science [110], security [111], robotics [112], Genetics and Bioinformatics [113]. The same underlying concept that enables accurate representation of the complex interconnection of legislature, chemical compounds and genetic properties together lends itself to application in the cyber security and cyber warfare domains.

2.5.3 – Applicability of ontologies to the realm of cyber security

Ontologies can be used to map the complex interrelationships of nodes, networks, software, vulnerabilities, exploits, and effects that make up the cyber domain. Because of the strength of knowledge representation offered by the ontological structure, these interrelationships can generate accurate system models to enhance the situational awareness of decision makers while retaining functional transparency of the instantiated representation. The transparency of the representation means that any modelling, querying or analysis of the results it produces is verifiable. Verifiability increases the efficacy of any simulation run on the instantiated knowledge structure. Generating these futures using a base ontological representation allows the decision makers of an organisation to use the situational awareness that the initial representation provides for applying

predictive futures algorithms to develop resilience strategies and move towards a desirable future state of preparedness.

2.6 – Related Work in Cyber Ontologies

This section of the thesis will examine existing ontological approaches to cyber security, in an attempt to identify existing transparent, accurate, comprehensive knowledge structures that are suitable for the representation of cyber effects in a military simulation context.

In 2001, Raskin, Triezenberg and Nirenburg first pushed the move towards the use of Ontological knowledge representations in the fields of information and cyber security [88]. These authors recognised the benefits of extensibility, machine-readability, modularity and systematisation of knowledge offered by ontologies. The authors also advocated the predictive capabilities of ontologies [88]. Their aim in proposing an information security ontology was to facilitate the processing of natural language reports and entries to gradually build up the collective 'know-how' of the information security community. The limitation of this approach is the implicit requirement to represent simultaneously multiple levels of representational granularity in the same knowledge structure. Broad approaches such as this risk compromising the ontological commitment of the representation and weakening the validity of the representation.

The 2006 work of Tsoumas and Gritzalis [114] aimed to develop a security ontology that could elaborate on the security aspects of an information system. Essentially, it was envisioned as an auditing tool to ensure compliance with standards. The focus on policy and process reflects the chosen worldview of the ontology and would make the modelling of attacks infeasible to represent and is hence unsuitable. Work into using a cyber security ontology for security requirements elicitation [111] demonstrates the capacity to determine current states of a system and then project potential improvements. However, the ontology is built on an asset-risk-mitigation framework that is steeped heavily in standards and policy. It is also fundamentally an auditing and planning tool and unsuitable for modelling the cyber effects on a military network.

The CRATELO ontology conceived by Oltramari et. al. [115] works towards describing secure operations in the cyber domain and improving the situational awareness of defenders. The paper also examines the disparate work done into the depth versus the breadth of existing cyber-ontologies. The authors assess that though there has been a significant enhancement to the field resulting from the categorization of concepts under programs like *the Common Platform Enumeration* (CPE) [116], *Common Weakness Enumeration* (CWE) [117], and *Common Vulnerability and Exposures* (CVE) [118, 119] there is a lack of a cohesive framework to tie them all together. The *CRATELO* ontology, though noble in its goal to collect and collate the disparate analyses of cyber-information to combine them into a broader ontological framework, is not suitable for this problem as it lacks granularity in its representation of assets and networks.

The development of cyber security ontologies within the context of the cloud by Takahashi, Kadobayashi and Fujiwara [120, 121], is focused heavily on audibility and implementation of controls and hence unsuitable for the modelling of effects moving through a network. Though focused on the development methodology over the substance of the ontology, work by Obrst, Chase and Markeloff toward developing an ontology of the cyber

security domain [122] leveraging the existing *Malware Attribute Enumeration and Characterisation* (MAEC) [123] is a newer development in the field. By using a conceptual diamond framework that links *Actors* to *Victims* to *Users* to *Infrastructure* and *Capability*, this ontology integrates into a multi-level, modular structure. The focus on modularity is in line with the ontological commitment guidance from Davis [83] and Gruber [91] and follows (unsurprisingly) the best practices laid out by Obrst in his earlier work on ontological architectures [90]. Currently, their ontological structure, in addition to lacking a clear, public schema is too malware-centric for adoption at this time.

The modular approach championed by Obrst enables the designer to select carefully the ontological commitments they wish to make and focus on them, maximising the chance of accurate representation of knowledge from a predetermined worldview. This philosophy of modularity and bridging between linked but disparate ontologies is useful in the context of understanding the effects of a cyber attack on a military network.

There are two prerequisites for effectively representing cyber effects. The infrastructure that is the target (or facilitator) of the attacks is the first component and the attacks themselves are the second. Preserving the differing requirements for ontological commitment between these perspectives necessitates that their representation should occur in separate ontologies. This process has been commonplace in the development of cyber effects simulation tools (see Section 2.3.4) since the separation of topology from attacks by the *TANDI* [74] information fusion approach in 2006. These disparate ontological structures should be bridged appropriately to maximise their cooperative utility. The following sections of discussion will highlight existing work in the domain of cyber ontologies focusing first on the representation of systems and infrastructure and second on those intended for modelling attacks.

2.6.1 – Ontological Network Models

The United States Air Force holds that one of the first prerequisites of a "*fight through* [approach to cyber resilience is] *...to map USAF network to USAF missions with end-to-end forensics approach.* [47]. The assertion here is that development of a detailed network model is the first step towards achieving resilience. Following the discussion of the work on terrain models, the current popular threat models will be examined.

The majority of the work into ontological representations of computer networks has focused on the representation of networks as an independent logical topology or terrain model. This section surveys the influential work in terrain model development and extracts the elements of each ontology useful to the development of work in this thesis. As part of this thesis a visualisation of this evolution of the dominant terrain model, Virtual Terrain was produced and is in Figure 4. The model depicts the evolving terrain elements (red boxes) the applications that used them (blue) and some older contextualising work (green).

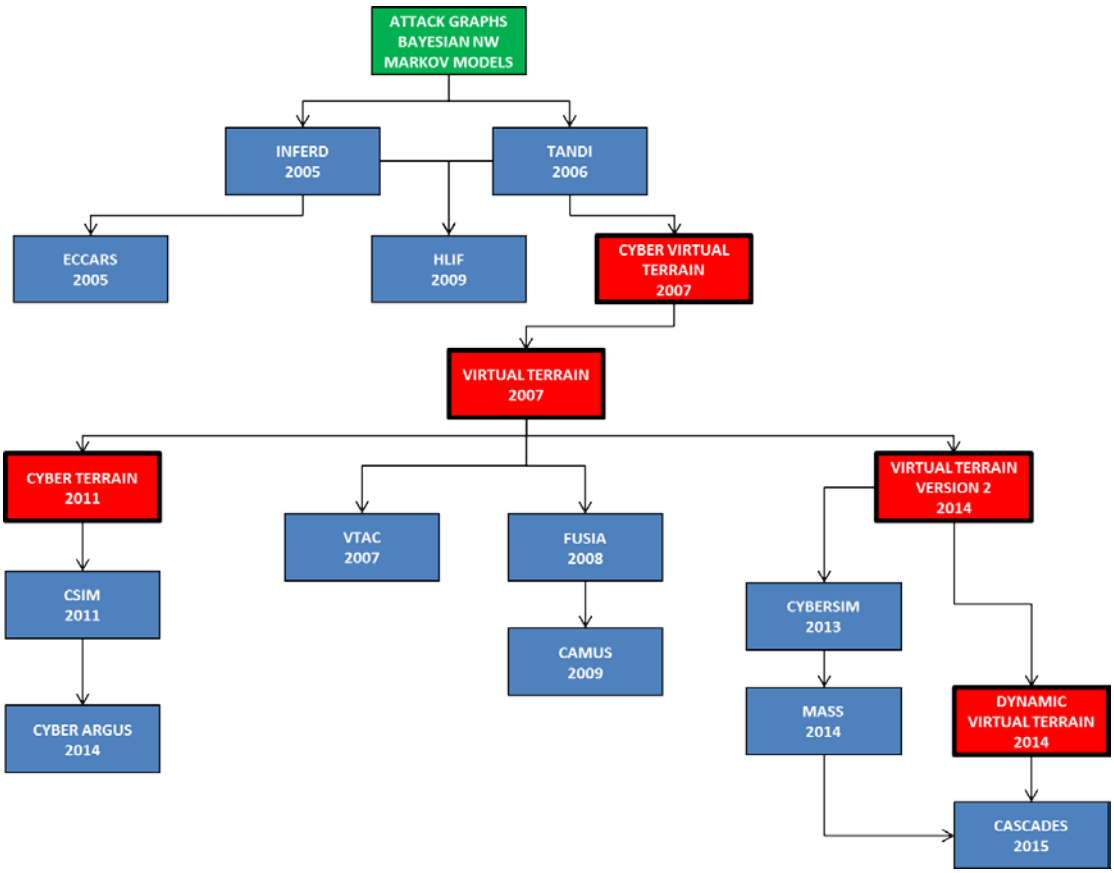

Figure 4 - The Evolution of Independent Terrain Models

2.6.1.1 Origins of Terrain Models

The representation of a network in an abstracted way to study its emergent phenomena is not a new concept. In the cyber security realm, there has been a significant amount of work conducted towards this since the mid-1990's. At that time, the focus was on utilising attack graphs, vulnerability trees Bayesian networks and Markova models [124-127] to model networks, threats against them and begin projecting likely future states. There are some limitations to this approach, salient among them being the computational complexity of enormous graphing problems.

The cumbersome complexity of graphing problems was further compounded by a paradigm shift toward real-time analysis of Intrusion Detection System (IDS) alerts in the early 2000s. The intent of this approach was to identify signature attack patterns and then use that information to predict future attacks. Attempts to make this process more efficient led to a focus on information fusion techniques and technologies. It was efforts to

improve the efficiency of these information fusion approaches that eventually resulted in the development of the *Cyber Virtual Terrain.*

2.6.1.2 Cyber Virtual Terrain

The *Cyber Virtual Terrain* (CVT) is an independent logical topology model developed as part of the TANDI project [74]. It rapidly became apparent that the generation of an independent terrain made the overarching modelling process much more efficient and had applications beyond the scope of the TANDI project. The idea of an independent representation of a computer network or 'terrain' began to gain attention. Work by Fava et .al. on terrain and behaviour modelling [128] applied the idea of an independent terrain to complex multistage cyber attacks, expanding the work in TANDI to model the logical accessibility between domains and introduce the concepts of access control and service trees.

2.6.1.3 Virtual Terrain

RITs Virtual Terrain (VT) project [129] has been the core base that has been used to develop most of the work in this area, as depicted in Figure 4. VT was the next evolution of CVT and aimed to be a "*common representation of crucial information about network vulnerabilities… creating a standard representation of a computer network so that research can focus on how to elicit information directly from the model rather than also needing to determine who the information is extracted from the network itself"* [129]. The early work on VT identifies that manually creating terrain at scale is infeasible and should seek to achieve maximum automation. It uses tools like NMAP[130] and Nessus [131] to feed information into a Java 2.0 application that generates the VT (including a visual rendering) based on a defined XML schema. This schema is the ontology that underlies VT. While it is briefly explained in papers on VT [129] and Impact Assessment for Cyber Attacks [132] the only meaningful public presentation of the schema is in Brian Argauer's 2007 Masters' thesis from RIT [133]. The details contained within the thesis are only a sampling of the full schema and not enough to make an accurate assessment of it. The VT has been implemented in a number of applications including VTAC [132] Cyber Situational Awareness [75-77], (FuSIA) [75] and (CAMUS) [76] CAMUS explicitly states the value of an established ontology and also introduces the idea of using an external RDF store to address issues of scalability suffered by XML. The development of VT then took two paths. One of the paths focuses on mission impact analysis and understanding system interdependencies. The other is used to support simulations of cyber attacks and network responses.

2.6.1.4 - Cyber Terrain

Jakobson's *Cyber Terrain* (CT) [69] is an evolution of VT. It differs from VT in two key ways. It split into three sub-terrains – The *Hardware, Software* and *Service* sub-terrains and it maps specific dependencies between and within each of them. This detailed mapping of dependencies greatly increases the granularity of the interconnections of represented hardware, software and services represented in the domain. CT supports Jakobson's Cyber Security Incident Model (CSIM)[70] and is extended by incorporating it into a broader context of mission planning and resilience building [71]. The core problem with the CT is, like the VT, no details of the underlying ontological schematic structure could be located to assess for suitability. The influence

of CT is also seen in the Cyber-ARGUS framework developed by Barros-Baretto [68] and the modelling of Situational awareness[65, 66].

2.6.1.5 - Virtual Terrain version 2

The other branch of the VT family led to the development of Virtual Terrain version 2 (VT.2). VT.2 is introduced as an extension to the VT as a part of the design of the 'CyberSim' Modular Cyber Attack Simulator [79] and by extension MASS [80]. The changes to VT.2 from its predecessor lie in the capability to define an incomplete network, improvement of the 'internet' concept and addition of a 'router' object that is a node with no associated vulnerabilities. VT.2 is intended for use in conjunction with a Vulnerability Hierarchy to efficiently map software and network configurations to their associated vulnerabilities.

2.6.1.6 - Dynamic Virtual Terrain

The most recent evolution of VT is the Dynamic Virtual Terrain (DVT) developed by Ben Wheeler in 2014 to accommodate the modelling of networks utilising Moving Target Network Defence Measures (MTNDM) such as IP-hopping, Port-hopping and Dynamic Firewalls. The static VT, CT and VT.2 models are unable to accommodate this. The four significant changes to DVT from VT.2 are the centralisation of the access control permissions list into a network-level matrix and the creation of terrain-manager agents to monitor and update the terrain based on the triggering of MTNDM algorithms. The third shift is to the creation of an evaluation function on the terrain to verify the ground truth of the network matches the attacker's perception of it (this accounts for the MTNDMs gaming against the attacker) DVT also includes node status and threat-level properties for the network nodes to support the MTNDM algorithms. DVT is currently being implemented in the CASCADES project under development at RIT [81, 82]. Identified future work for DVT includes implementing the modelling of network traffic, implementing connection sessions, and optimising the validity evaluation algorithm. The underlying issue with DVT as with VT, CT and VT.2 is that there is no publicly available ontological schema for analysis. DVT's enhanced capability to represent complex aspects of dynamic networking is at the cost of fidelity – lost by centralising the access control list to a network -level matrix.

2.6.1.7 – Summary of ontological network models.

Each of these terrains has been assessed to determine their suitability for addressing the problem of representing cyber effects in a military simulation context. However, lack of a publicly available schema for any of the terrain models has prevented a detailed investigation. Based on the publicly available information, a consistent issue among the models is a lack of granularity. CVT, VT, VT.2 and DVT all implement node clusters to represent groups of computers. Node clustering simplifies the modelling at the cost of fidelity. The abstraction also limits usecases such as multiple network connections per computer. Lateral movement through a network uses pivoting, a fundamental element of the cyber attack process [9] (described in Figure 1) and needs to be part of any meaningful representation of cyber attacks. The mapping of dependencies by the VT family of models is not sufficient to realistically represent the network. CT maps the dependencies in a very granular manner but then abstracts much of the associated detail of these relationships. DVT has stepped further away from reality, centralising many functions of the network and modifying the terrain structure to represent complex MTNDMs in the terrain as an isolated tool. Achieving this has required a broader spectrum of ontological commitments to

be made and has begun to impact on the accuracy of representation. If the accuracy of what is represented is reduced, it poses questions about the credibility of the representation. Existing terrain models also do not address the actual flow of information across the network, sessional communication between nodes, wireless as its own usecase, data spill or virtualisation. To touch briefly on the choice of encoding, XML is not a robust ontology description language. XML limits the implementation to being a syntactic representation of the ontology, unable to articulate properly the semantics, relationships, characteristics or attributes that it is representing [89]. The semantic strength of ontology is one of its greatest strengths. Though Gruber's assertion that an ontology should avoid an encoding bias is important, so it the strength of implementation being able to support the strength of the design. XML does not possess the equivalent semantic capabilities to a formal knowledge representation language such as RDF or OWL. This will make a product implemented in this language inherently weaker. This is why the decision by the designers of CAMUS to transition to the use of RDF allows for a much stronger representation of the ontological knowledge structure is assessed to be a much stronger choice than XML. Ontologically strong representations allow the knowledge representation can be more precise, more accurate and is more likely to produce accurate results. RDF is also more easily human-readable than XML (Particularly when written in the Turtle Syntax [134]) and is easier for a human to read, understand and validate what they are seeing.

The work into existing terrain models has been evolutionary in nature. The confinement of the development to two institutions (The Rochester Institute of Technology and Altusys) and a distinct lack of published work for the amount of development that has occurred have impeded the spread of the independent terrain model. This survey has not covered all the simulations and frameworks that have incidentally separated out their network model but rather has focused on the underlying design of these independent models, deducing the functionality of each. Ideally, one of these terrain models would have been used to support the representation of knowledge required to support the representation of cyber attacks and effects in the military simulation context. Lack of public schema and issues with granularity have resulted in the decision to build a new terrain model that addresses the identified gaps and improvements of these models.

2.6.2 – Threat-Focused Cyber Ontologies

Some ontological models exist to represent the domain of knowledge of the attacks carried out on a network by a threat. Though there is existing work towards other implementations of a threat-centric ontological structure such as those used by openIOC [60] and the IODEF framework [61], (and many of the other threat intelligence solutions) The three most influential of these are STIX, ThreatExhange and CycSecure. Each is examined in turn for suitability.

2.6.2.1 – Structured Threat Information eXpression

The emerging industry standard for the representation of threat intelligence is the *Structured Threat Information eXpression* (STIX) [59]. STIX focuses on the detection of cyber observables at a point along the 'kill-chain'. Once detected, it will analyse the attack associated with that observable to generate knowledge of other observables that are then collected into more holistic views of the attack and all the related components. A summary of these compiled into a report that is which it sends to other STIX users using the *Trusted Automated*

eXchange of Indicator Information (TAXII) [135]. The intent for this is to minimise the time it takes for details of an attack to be promulgated across the community, thereby reducing the window for an attack that a threat has available to engage multiple targets with the same techniques. The observed elements of the attack can be analysed further to profile the attackers and identify possible parent campaigns.

Figure 5 is a visual representation of the STIX Schema. One of the key aims of STIX is to facilitate the attribution of attacks for prosecution and threat intelligence. The STIX schema is attribution-focused, using the structure of a *Campaign* to aid this effort. A *Campaign* groups related *Threat Actors*, their *Tactics, Techniques and Procedures* (TTPs) and *Incidents* to which they have previously been attributed. The *threat actor* structure is used to characterise individual attackers – linking them to known attacks, *TTPs*, skill levels and motivations if known. *TTPs* or represent the behaviour of threat actors. These behaviours could reflect things like preferred targets, commonly used exploits and attack patterns. An *Incident* construct contains details of a security incident. Typically this is Time, Location, an outline of the incident and details of any response. The *Incident* links to the *Threat Actor,* who perpetrated it, the *Observables* related to the incident and the *Target* of the incident. *Indicators* are chains of *observables* map to a *TTP*. The *indicators* are used to determine when attacks may be occurring and what the possible *TTP* is that is causing the incidents that are generating observables and indicators. The *Exploit Target* is the link to the vulnerabilities that are exploited by an attacker. They link to TTPs as the particular vulnerability that they represent. They also contain information linking to some the MITRE standards like *Common Vulnerabilities and Exposures* (CVE)*, Common Weakness Enumeration* (CWE)*, Common Platform Enumeration* (CPE) *and the Common Vulnerability Scoring System* (CVSS). Finally, *Courses of Action* are the responses, mitigations or preventative measures associated with *Exploit Targets*, *Incidents* and *TTPs.*

The main criticism of STIX is that it is "too comprehensive"; the detail that it contains is a collection of *"sprawling definitions that are hard to maintain and port into machine-readable formats"* [115].

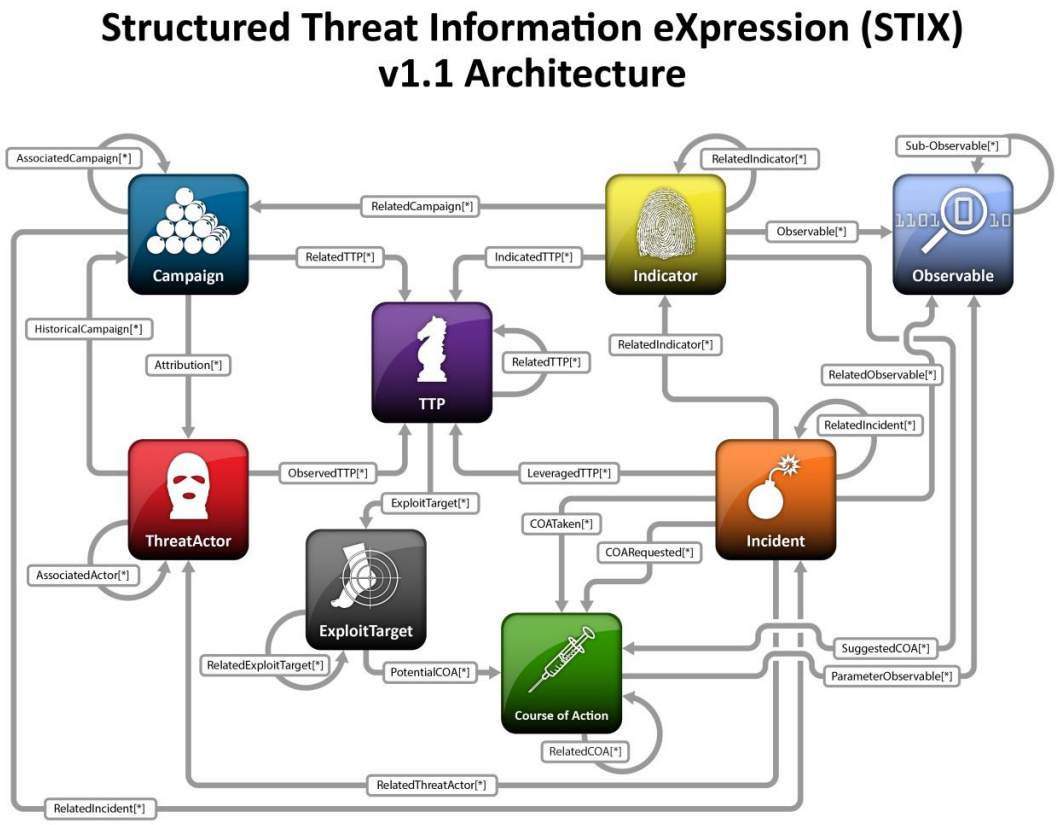

Figure 5 - STIX High Level Architecture

Similar to the problem of VT, the underlying XML that STIX is built on is an inelegant implementation. Currently, the community is working on a *JavaScript Object Notation* (JSON) implementation of STIX and has already produced a python API called python-STIX to improve usability. The functionality of JSON for a similar usecase has been demonstrated by DSTOs Parallax: Battlemind 1.5 [136] project. The project serves as a proof-of-concept that demonstrates the capability of JSON based systems to operate at a similar scale to the XML in STIX.

Though it is intended as a threat intelligence platform, STIX is already being experimented with for other, related but distinct functions. In addition to those organisations that are formally recognised users of STIX [137] the 'STIX 2 IDS' project [138] is attempting to implement STIX as an IDS. This project is still in its infancy but it does indicate that others also see the potential for using STIX in roles beyond pure threat intelligence. STIX is an open-source, collaborative framework with large community and significant commercial uptake. It is a valuable tool that should be considered for future use, despite the contentions about its comprehensibility and XML encoding.

2.6.2.2 – Facebook ThreatExchange

The conceptualisation of the *Facebook ThreatExchange* [139] originated from the response by Facebook and several of its industry partners to counter the botnet associated with the *Regger malware* in late 2013. In a talk delivered to a *Spam Fighting @ Scale* conference in May of 2015, Mark Hammell, the *ThreatExchange* project lead, explains how their ad-hoc response to the *Regger malware it* made it apparent that to continue intelligence

sharing into the future they needed to adopt a threat intelligence solution. After assessing and determining that none of the commercial solutions were fit-for-purpose, Facebook elected to create an open-source and free-access application that integrates with their platform.

Facebook envisions *ThreatExhange* as the first step towards automating the sharing of threat intelligence, "*an API-based platform that allows users to share what they want with who they want when they want" [139]*. The Facebook *ThreatExchange* is a graph database built off an underlying ontological structure. The ontological structure defines what the shareable elements are (Indicators of Compromise, Malware Signatures, etc.) and uses the interconnection of these elements to create a contextual model that allows them to model attacks as interlinked chains of observables. These associate with metrics of confidence, maliciousness and severity that can be used in the analysis of threats to business function. The ThreatExchange system leverages the Facebook backend (retaining compliance with Facebook standards and conventions) and primarily uses RESTful JSON. It is linked to Facebook's searching power able to leverage this in its querying [139].

Facebook has claimed that they are attempting to build tailored usecases for specific, niche user groups including research editions, social media and corporate network variants. Facebook claims that the uptake of the Beta has gone beyond the 'silicon valley' users to include clients working in critical infrastructure, corporate clients and wider web-service clients [139]. Elements of their API are available on GitHub [140] and Facebook is actively encouraging participation by developers and partner clients to continue to develop the ThreatExchange effort. Their API Documents [141] define Objects, Types, Search endpoints and Miscellaneous Endpoints that are all interconnected to form a cohesive threat intelligence picture.

Though it is publicly available, their API does not include an explicit schema document. The definitions that are available show a threat intelligence system that forms around two cores. These are the 'Threat Descriptor" object that is the contextual conglomeration of the indicators and their metadata and associates with the 'owner' who entered it into the system and the malware object that accrues metadata about specific malware elements. Currently no academic literature or independent review has been conducted into the *ThreatExchange* regarding its use or structure. Lack of detailed, impartial analysis makes it harder to assess its credibility. Lack of formal evaluation combined with *ThreatExchange's* dependence on integration with Facebook application software makes it unsuitable for the representation of cyber effects in the context of military simulation.

2.6.2.3 – CycSecure

CycSecure was the first of the cyber security ontologies, introduced in 2001 by Cycorp as the first commercialisation of the Cyc project. CycSecure was designed as a tool to be a *"Network risk assessment and network monitoring application that relies on knowledge-base artificial intelligence assessment"* [106]. It comprises of four main components: *The Knowledgebase, The Inference Engine, The Planner* and the *Natural Language Parsing and generation* components. The asserted capabilities of CycSecure include *compound vulnerability analysis*, *criticality analysis, chaining vulnerabilities into sequences, what-if analyses* and *continuous monitoring*. The process that CycSecure would follow to achieve this is to scan the network, identify vulnerabilities, plan multistep attacks, conduct what-if analyses and iterate as many times as required.

The Paper introducing CycSecure [142] claims that the knowledge base was rapidly assembled, taking four 'knowledge enterers' only 18 months to enter the relevant knowledge to support the database. The assessment by Cycorp that almost 7000 labour hours as being 'rapid' indicates problems in construction and maintenance of the knowledge structure. CycSecure's concepts are entered almost exclusively manually indicating that these maintenance activities will also be labour intensive. It also is noted by the paper that to function effectively on bigger networks it "needed to be provided extra computing power", perhaps lacking due thought to the potential cost associated with this. CycSecure is fundamentally dependant on human intervention, though the paper alludes to 'daemons' that will automate the process but does not supply sufficient detail about the implementation. The approach lacks transparency and modularity. The commercial nature of CycSecure means that the underlying ontological structure is not accessible and therefore it will not be possible to validate any emergent phenomena generated from the model. In addition to these limitations, the fundamental purpose of CycSecure is to be a threat assessment tool, not a simulation model. This limits its applicability to the problem of representing cyber effects in a military simulation context.

Overall, CycSecure is designed to try and be a vulnerability assessment tool that also does threat modelling and attempts to fulfil numerous other roles that seem inherently incompatible from a knowledge representation perspective. The schema is proprietary and therefore not available for review and is an old structure, with at least fifteen years of history behind it and a distinct lack of new literature regarding it in recent years.

2.6.2.4 – Summary of Threat-Centric Ontologies.

CycSecure was a leading effort when it was first developed though it has not aged well and continuing research and development into it has slowed. The Facebook ThreatExchange brings a modern encoding standard (JSON) to the development of threat intelligence and can leverage off the existing Facebook infrastructure. STIX brings a very comprehensive, open source knowledge structure with large community and commercial buy-in.

Issues underlying the three models include the lack of public schema and reliance on Cyc for CycSecure, a reliance on the Facebook backbone and a lack of breadth in implementation and contained knowledge for Facebook and the 'sprawling definitions' of STIX that make it inaccessible to some. STIX also is built in XML, a language not designed for ontology description.

2.7 – Summary of Literature Review

This literature review has examined the changes that the proliferation of technology is having on the world. The requirement for networked command and control to operate effectively in the future operating environment potentially places militaries are at risk of becoming dependent on their networked C4ISR capabilities. These capabilities will become targets for enemies during the conduct of hostilities and could lead to compromise of these systems. Total compromise could force militaries to continue the fight in a 'degraded-information' environment. To "*fight through"* these scenarios, military forces need to be resilient. Resilience is gained by achieving situational awareness and then using that to begin enumerating the possible, plausible and probable futures. Once these futures are enumerated, decision makers take action to prepare for them, attempting to shape

the actual future towards a preferable future state that benefits them or conduct contingency planning to reduce the response time and improve resilience.

The problem that underlies this goal is that knowledge representation is inherently imperfect both manually and through automated tools like simulation and may not always produce accurate results. Variance in results will become more pronounced in proportion to the complexity of the represented system and the distance into the future that is being predicted. To maximise the accuracy of simulations using knowledge representations, a networking approach to deconstructing system complexity is used. This approach permits the representation of concepts and interconnections of the system in a knowledge representation. Prediction of futures is achieved by observing the emergent phenomena of this knowledge structure. The Emergent phenomena, and hence futures, are highly dependent on the underlying network strcture. To maximise the trust between decision makers and the results of their simulations, a robust, transparent knowledge structure that will enable the validation of the results of the simulation in the face of doubt is required. The selection of an appropriate knowledge structure is dependent on the formality and semantic strength it offers, for this reason, an ontological structure has been selected. The existing work on cyber ontologies has either tried to achieve too much or was not suitable for the representation of cyber effects on a military network. The bridging of two related but independent ontologies is the most effective method to balance granularity and disparate world views. To understand how to represent cyber effects and the interaction of interdependencies that generate them, there is a need to represent the infrastructure and the attacks separately. An analysis of the existing simulations and SA tools reinforces the need for the conscious separation between the logical terrain of the underlying network and system infrastructure and the red-team elements that represent the attackers. This separation produces individually and then collectively stronger ontologies.

The discussion of this chapter clearly identifies a gap in existing research. There is no existing (publicly available) knowledge structure that is complete, transparent, accurate and permissive of the modelling of cyber effects in the context of a military simulation. There is no existing combination of existing ontologies designed to represent network terrain models or cyber attacks that are suitable for this context. There is also no existing ontological framework to combine these two component models into a broader effects knowledge representation, utilising ontology bridging techniques to maximise the controlled interaction of the ontologies to this end. This gap in the context of the problem facing military forces in the future creates the imperative to answer the research question of determining how to represent effectively the cyber effects on military systems in a simulated environment that promotes transparency, comprehensiveness and understanding.

Chapter 3 – Research Methodology

3.0 – Preliminaries

Explain how, and on what basis you conducted your study…for a minor thesis, understand how your research design affects your overall argument"

- Gruba and Zobel 2014 [143]

3.1 – Introduction

The aim of this thesis is to produce research results that will make a positive contribution to the fields of knowledge representation and cyber security. The robustness of these contributions will is assured through the use of a formal methodology. The use of a formal methodology will structure the research, inform inquiry decisions and assist in the scoping of the thesis and specific research questions. Creswell asserts that the considerations of this methodology must address the research approach, the researcher's philosophical underpinnings, the research design and the research methods all in the context of the research problem [144].

Section 3.2 will recap each of the research questions and define the success criteria, contextualising the problem. Section 3.3 will briefly explain the underlying research theory and the epistemological standpoint of this thesis' research design and research methods. Finally, section 3.4 will apply the methods in the context of this thesis, developing a detailed and robust research methodology.

3.2 – Review of research questions

Research Question:

How can we effectively develop a method of representing cyber effects on military systems in a simulated environment that promotes transparency, comprehensiveness and understanding?

Subquestion 1:

What is the current state of knowledge representation for the field of cyber security? Is it comprehensive and does it promote transparency of representation? Does it allow the effective modelling of cyber effects in a military simulation context?

To determine the need for, and context of, a knowledge structure that effectively represents cyber effects in a military simulation context a broad survey of the current work in the area is required.

To answer this question, the existing work into cyber-centric knowledge representations must be analysed and assessed based on their ability to comprehensively and transparently represent cyber effects in a military simulation context. The conduct of this review will also conclude the most effective existing approaches and seek to identify reusable or extensible elements of existing work to maximise interoperability with the existing body of knowledge.

Subquestion 2:

What are the areas, fields and disciplines that a comprehensive, effects-focused knowledge structure should encompass or represent?

The area of cyber security is exceptionally large and diverse. The representation of every single element of cyber security listed by Authors like Stallings and Brown [32] and organisations like the International Information Systems Security Certification Consortium ($(ISC)^2$) [145] is both impractical and unhelpful. To produce a knowledge structure that is succinct as well as comprehensive, an analysis of the domain of knowledge is required. This analysis will determine the elements that are relevant to the representation of cyber effects in a simulation context. This will be conducted through a review of current cyber effects knowledge, a survey of select anthologies of domain knowledge [32, 145-147] and through an analysis of contemporary issues and reporting in the field of cyber security.

The answer to this question will clearly articulate the content requirements for a knowledge structure appropriate for representing cyber effects on networks in a simulation context. The answer to this question will include the import or generation of suitable usecases to represent the specified requirements and focus the development methodology.

Subquestion 3:

What is a suitable agile, usecase-centric development methodology that can be used to develop a suitable knowledge structure?

Development of knowledge representations is typically a time and resource intensive task that requires the colocation of experts from multiple domains of knowledge. The more formal and machine-readable a knowledge representation is the less focus there is on an agile approach to development. The shifting of software development paradigms towards agile approaches such as Test Driven Development [148] has the potential to influence similarly induce a shift in the paradigm of knowledge representation. In the context of this research, an agile development approach will facilitate an evolutionary development based on incremental usecases. An agile approach will also compensate for the inexperience of developers and limited domain knowledge of the designers.

The answer to this question will define an agile, usecase driven knowledge structure development approach appropriate for use by designers and developers who are not domain experts but would benefit from the use of a knowledge representation.

Subquestion 4:

What is a suitable knowledge structure to support the representation of cyber effects in a simulation context?

To develop a method of effectively representing knowledge to support the simulation of cyber effects in a military context a suitable knowledge structure must be designed, implemented and evaluated.

To answer this question, a knowledge structure will be constructed using the methodology resulting from Subquestion 3. To evaluate to a satisfactory level, the knowledge representation will be required to effectively represent a series of usecases and elicit relevant information from them towards enhancing understanding of the domain.

Subquestion 5:

How are the results of the knowledge representation validated; what is a suitable collection of relevant usecases to facilitate this evaluation?

To evaluate the capacity of the knowledge structure to represent effectively the specified requirements of the domain of knowledge, a collection of relevant usecases for the domain of knowledge to must be identified. The intersection of domains that this knowledge structure is positioned at requires that consideration be given to cyber-warfare, cyber-crime, tactical land combat, and operational cyber networks to elicit suitable usecases.

To answer this question, a robust testing methodology must be established to check that the knowledge structure is capable of representing, and eliciting information from a selection or relevant usecases.

3.3 – Theoretical underpinnings of research methodology

Creswell asserts that the selection of a research approach is dependent on the nature of the research problem, the experiences and biases of the researcher and the target audience. The character of the research problem is the most significant and deterministic factor in determining the approach. The research approach must bridge the character of the problem to the nature of the audience that will be reviewing it to be meaningful.

The research approach is the overarching framework that is broadly defined by Creswell as:

"[the] *plans and procedures that span the steps from broad assumptions to detailed methods of data collection, analysis and interpretation… informing this decision should be the philosophical assumptions that the researcher brings to the study…"*

The three approaches advanced by Creswell are the *Quantitative approach*, the *Qualitative approach* and the *Mixed Methods approach*. The *quantitative approach* most closely aligns with the traditional scientific method of research [149]. The *qualitative research approach* is much better-suited to exploratory studies. Its primary limitation is the inability easily mathematically or computationally analyse the results as part of a mathematical or computational assessment The *mixed methods research approach* enables the researcher to take the best aspects of the *quantitative* and *qualitative* approaches, often using the results of one approach to inform the conduct of the other. Underlying these methodologies is the interconnection of philosophical worldviews, designs and research methods.

A worldview is traditionally a perceived as the highest manifestation of philosophy and is often used to emphasise a particular perspective on an issue, problem or system. It is a global image of the world constructed in a way that will enable the understanding of as many elements of the experience as possible. The worldview traditionally consists of six subcomponents. These are the *ontology*, *the explanation*, the *prediction*, the *axiology*, the *praxeology* and the *epistemology* [150]. These components represent the model of reality (what is), the model of the past (what was), the model of the future (what will be), the theory of values (what is right), the theory of actions (what should we do) and the theory of knowledge (what is true). In the context of research, this can be thought of as 'what is the domain of knowledge that we are working in?', 'how did this body of

knowledge come to be?', 'where will this work take it?', 'what should this be trying to achieve?', 'how do we reach this point?' and 'how do we know when we have answered these questions?'. Creswell defines it more succinctly as a core set of beliefs that guide action [144]. Defining the components of the worldview is critical to ensuring that the cohesiveness of the research approach and the compatibility with addressing the research problem. The following table more clearly summarises these relationships [144, 150]:

Components of a Worldview			
Element	**Nature**	**Inquiry**	**Contextual Inquiry**
Ontology	Model of Reality	What is?	What is the Domain of Knowledge?
Explanation	Model of the Past	What was?	How did the Domain of Knowledge come to be?
Prediction	Model of the future	What will be?	Where will this work take the Domain of Knowledge?
Axiology	Theory of Values	What is right?	What should this work be trying to achieve?
Praxeology	Theory of Action	What should we do?	How do we achieve this?
Epistemology	Theory of Values	What is true?	How do we know we have achieved it?

Table 1 - Components of a Worldview

The research designs are the framework to guide the specific research activities. The research design is the second element of the research approach. The third component of the research approach is the research method. The research methods are the tools and techniques that the researcher will use to collect, analyse and interpret their data.

3.4 – Research methodology of this thesis

This section will address the research methodology in some parts. First will be an overview of the applicability of Creswell's framework to this research from a broad context. Following this will be an overview of the three most influential research and development approaches on this thesis: the Design Science Research Methodology, the Methodology and Test Driven Development. This review will show that none of these are suitable to inform independently the research and development of this thesis, establishing the context for Chapter 4 to introduce a novel approach to agile ontology development.

3.4.1 – Research Methodology – Application of Creswell's Framework

The research problem that this thesis is addressing is fundamentally the exploration of a new area of research at the intersection of the knowledge domains of military operations, cyber security, simulation and knowledge representation. The research approach that is most suitable to this research problem is a mixed methods approach. This approach will allow the adoption of an exploratory perspective that qualitative approaches to inform a quantitative assessment of the knowledge structure that is developed. In this context, the substantive

detail of the research approach including the worldview, research design and research methods will be provided by the *Design Science* approach to Information Technology Research. To most effectively produce a solution to this problem agile development techniques will be applied to iteratively develop the final product.

3.4.2 – Research Approach: Design Science

The particular approach that is most suited to the exploratory research and development of this nature is *Design Science* (DS) (Depicted in Figure 6). DS is a type of mixed methodological research specific to research in Information Technology. DS focuses on the development of technological artefacts to meet organisational needs and the associated theoretical work to support these artefacts [151]. The DS approach is built heavily on the pragmatic worldview. The *ontological* element of the DS worldview assumes that the knowledge domain of interest will change over time as additional knowledge accrues, requirements are specified and the focus of activity shifts. The *explanation* element is achieved through the theoretical research to understand the context of the required artefact and the nature of the gap to be filled. The *prediction* element of this approach is achieved by deducing what the requirements of that artefact are to solve the gap identified by the explanation. *Axiologically,* the success of a DS research project is determined by the utility of the produced artefact to meet the requirements specified in the prediction. Determining this requires a quantitative evaluation of the ontology – testing to ensure it meets the functional requirements. The *praxeological* element of DS centres on the process used to develop the artefact. The chosen development methodology will be discussed in the next chapter in depth. The *epistemological* element of this worldview is driven by the intervention of the researcher to effect change. The intervention is the design and development of the artefact, and the change is the assessment dependant on the real (or potential) impact that the artefact has to influence organisational change. In this particular case, how useful will the ontology be at enabling the representation of the effects of cyber attacks on military networks?

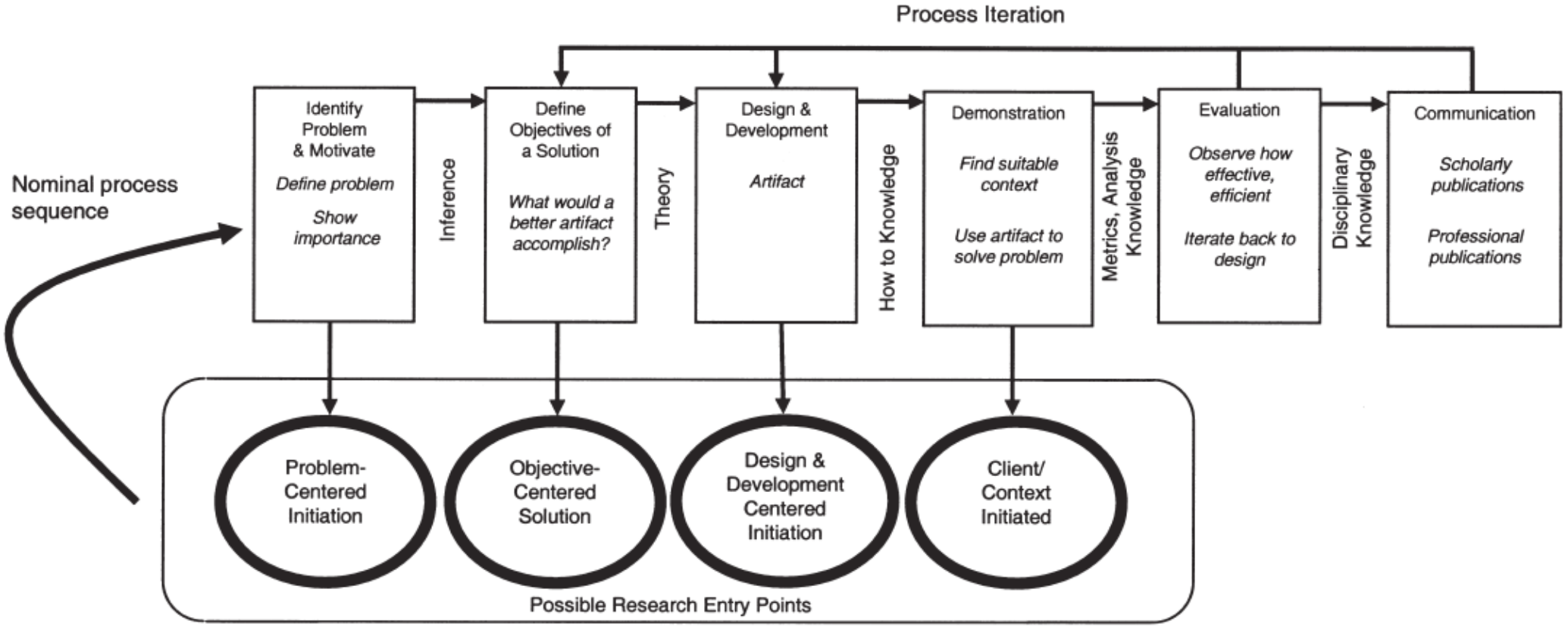

Figure 6 - Design Science Research Methodology Process Model [2]

3.4.2.1 – Applying Design Science: the Design Science Research Methodology (DSRM)

The specific process that is most useful to follow within the broader realm of DS is the Design Science Research Methodology (DSRM) [2]. As shown above in Figure 6, the DSRM approach consists of six activities. The first is the identification of the problems and the motivation for the work to be undertaken. Second is the definition of the solution objectives. The third activity is the actual design and development of the artefact. The methodology that has been employed to achieve this stage will be discussed further in Chapter 4. The fourth activity is a demonstration of the capability of the artefact. The fifth activity consists of a formal evaluation of the performance of the artefact against the defined objectives. The sixth activity is the effective communication of the results of the research and development process.

DSRM is an exploratory sequential mixed methods approach to research. The use of mixed methods enables the smooth transition from the qualitative gathering of requirements through the design and development of the artefact and finishing with a formal quantitative evaluation of the ontology. The drawback to the use of the DSRM is that it is largely a principled framework that lacks some of the specific procedural frameworks useful to novice designers and developers. Additionally, there is scarce information about the direct employment of this research approach in ontology design. For this reason, consideration must be given to specific ontology design methodologies to support the DSRM process.

3.4.3 – Development Approach: Methontology

A formal development methodology will be used to ensure the efficacy and robustness of the artefact developed as part of DSRM. This formal development methodology will be a subordinate to the principles of ontological design espoused by Gruber in 1995 [91] that form as the foundational tenets of the field. These principals have guided the development of numerous different methodologies. The leading methodology in the area of ontology development is the *Methontology* approach described by Fernández-López, Gómez-Pérez and Juristo in 1997 [152]. The Methontology approach correlates with traditional software engineering approaches (particularly the IEEE Standard for Developing a Software Project Life Cycle Process [153]) and is designed to guide the creation of an ontology 'from scratch' based on first principles. The approach targets enterprise-level ontology development scenarios. Additional advantageous comes from a demonstrated utility in the fields of law [108], chemistry [109], information science [110], security [111] and robotics [112] among many others to be highly effective. The disadvantage is that because of its targeting at the larger, enterprise-level ontology developers it can be difficult for novice developers to employ the Methontology process effectively.

The Methontology process itself is conceptually straightforward. There are three domains: *development*, *management* and *support.* The *development* domain consists of five activities: *specification, conceptualization, formalisation, implementation* and *maintenance.* The management domain has two activities: *control* and *quality assurance.* The support domain has five domains: *knowledge acquisition, integration, evaluation, documentation* and *configuration management.* (See Figure 7).

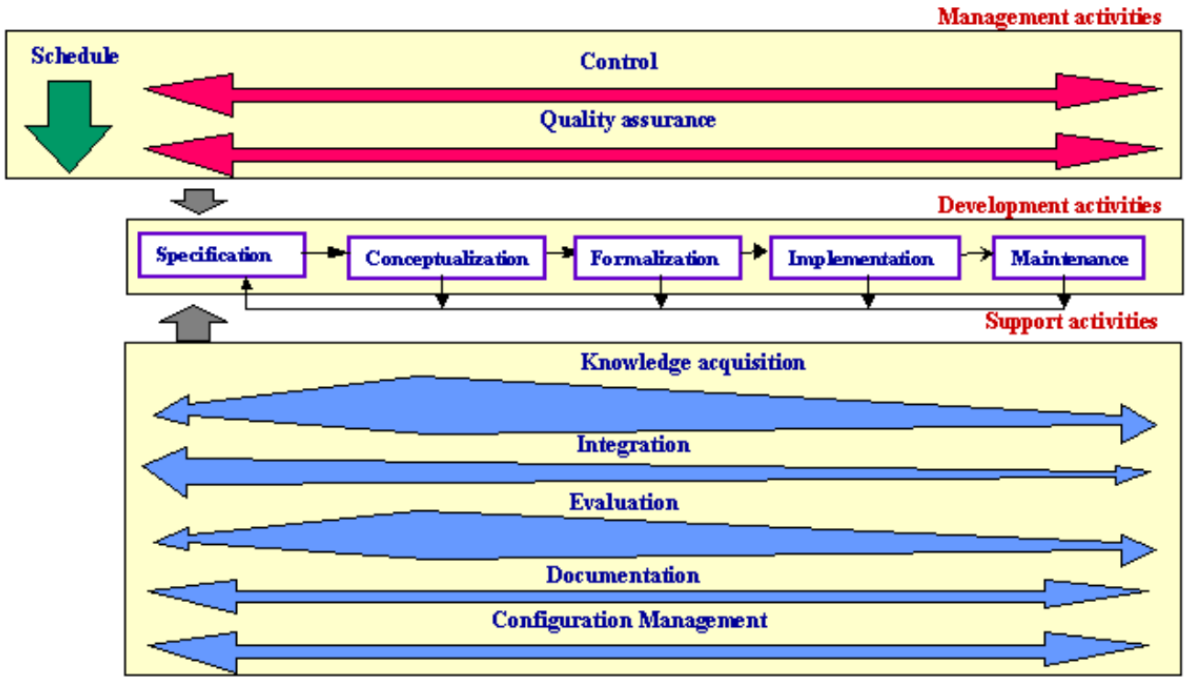

Figure 7 - The Methontology Development Activities [1]

The key elements of the Methontology development domain are:

Specification: The specification activity of the Methontology focuses on the development of an ontology specification document that will state the purpose, formality and scope of the ontology. The produced specification document is to be concise, complete and consistent.

Conceptualization: The conceptualisation activity is where the informal design takes place. The key terms are identified, structured and given relationships to each other. This activity also requires the developer to identify any existing ontologies, taxonomies, or knowledge stores that could be reused or extended to assist with the development.

Formalisation: The formalisation activity is where the designer takes the informal concept model and adds detail to the concepts and relationships. Attributes, by definition, are constraints, rules and axiomatic statements. It is good practice to integrate with the relevant elements of the previously identified ontologies where possible. This activity should also see some initial evaluation of the theoretical model – using competency questions defined in the specification phase to desk check the ontology's functionality before encoding it. The aim of the formalisation stage is to get the ontology as close as possible to being machine-readable.

Implementation: The Implementation phase is the formal encoding of the ontology in an appropriate language such as RDF [154] or OWL [155]. This phase will encompass the formal verification and validation of the encoded ontology.

Maintenance: As the evaluation process identifies shortfalls, they must be fixed. Similarly, expansion of the ontology into the future and changes to the structure may be required and are included in this activity.

The flow of these tasks is depicted in Figure 8. Figure 8 is a tailored version of the Methontology process as defined by Fernández-López, Gómez-Pérez and Juristo [152] with elements from some usecases of legal ontology design [108], Information-Science Ontology Design [110] and chemistry ontology design [109] that was produced in the course of this thesis towards enhancing the understanding of the *Methontology* process

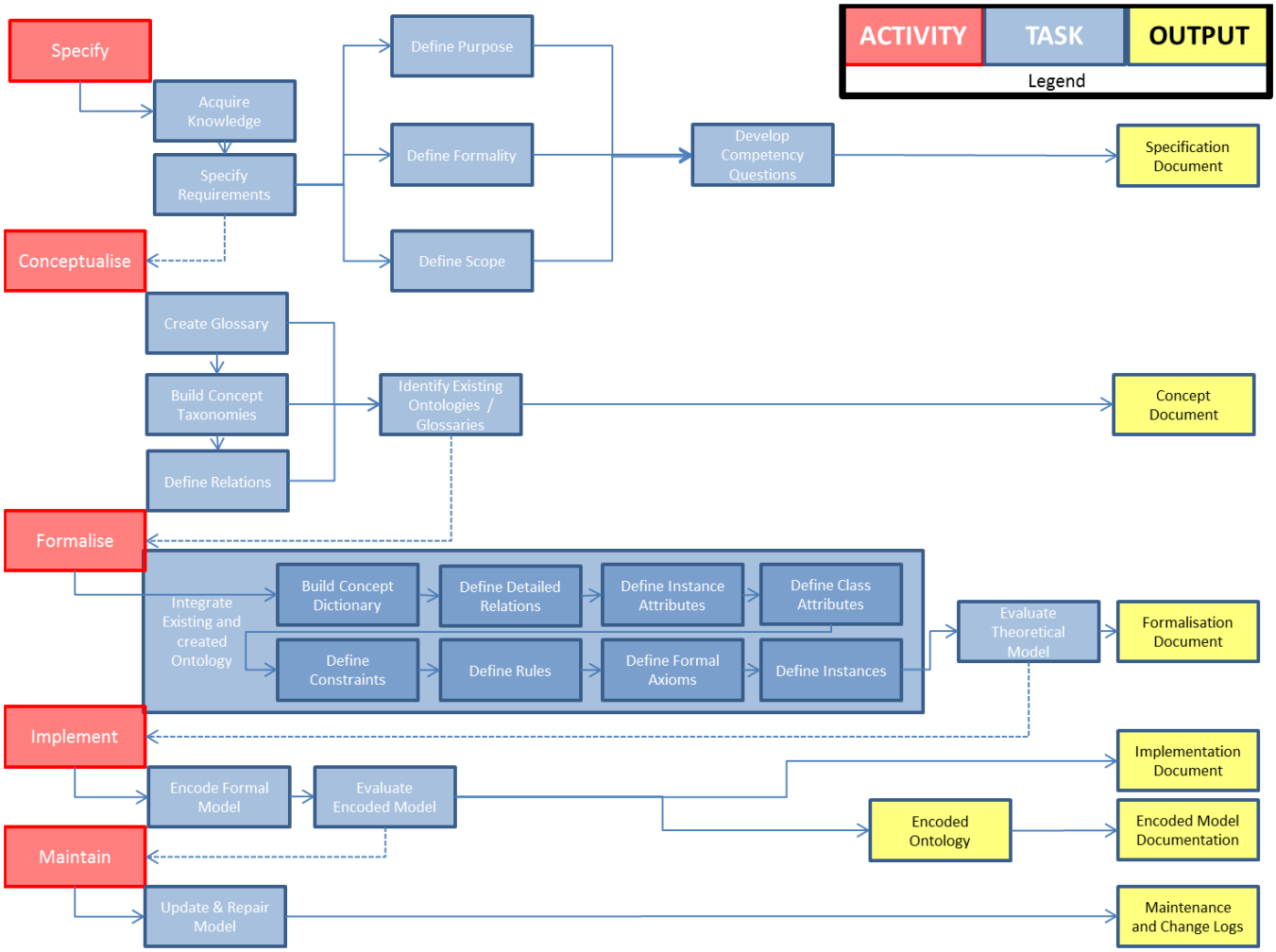

Figure 8 - Summary of Methontology Task Flow

As is apparent from the task flow and the descriptions above, the *Methontology* process is a difficult and is targeted toward experienced ontology developers. When this experience is lacking, the entire process is slowed and made more difficult. Utilising the Methontology method in the exploratory development of a cross-domain is handicapped by the activity of 'knowledge acquisition'. The intersection of the *Simulation, Cyber Security*, *Military* and *Knowledge Engineering* domains is not well established or documented at present and, as a result, many of the requirements, interdependencies, relationships, and evolutionary development issues are unknown. This quagmire of unsorted information can easily become a trap for the inexperienced developer's

underdeveloped scoping skills. An agile approach is much more suited to this scenario. Agile software engineering approaches are now customarily taught to novice developers working on small or evolutionary projects to improve teamwork, reduce the learning curve and make software engineering more accessible and enjoyable. Agile approaches allow incremental building based increasing scope and complexity gradually, allowing a more natural growth of ontological completeness and less up-front domain knowledge than the Methontology approach.

3.4.4 – Development Approach: Test Driven Development

The agile development approach that will best enable the gradual, incremental growth of an ontology under development in an exploratory manner is the Test Driven Development (TDD) approach.

Test Driven development is an agile development approach that differs from traditional development by moving to a test-first rather than test-last process (see Figure 9). The process will identify a high-level architecture and define the project objectives and then commence writing tests that must be passed to continue development. The artefact is then developed to pass the tests in the most minimalistic way possible before being refactored (genericised, tidied and optimised) to improve efficiency. This process is conducted iteratively for each test defined along the road to the project desired end state. The normal evaluation and testing period will still occur at the end of the project. Due to the consistent testing throughout the lie of the project it is far less to discover bugs or problems [3]. Figure 9 four contrasts TDD against a traditional test-last development approach.

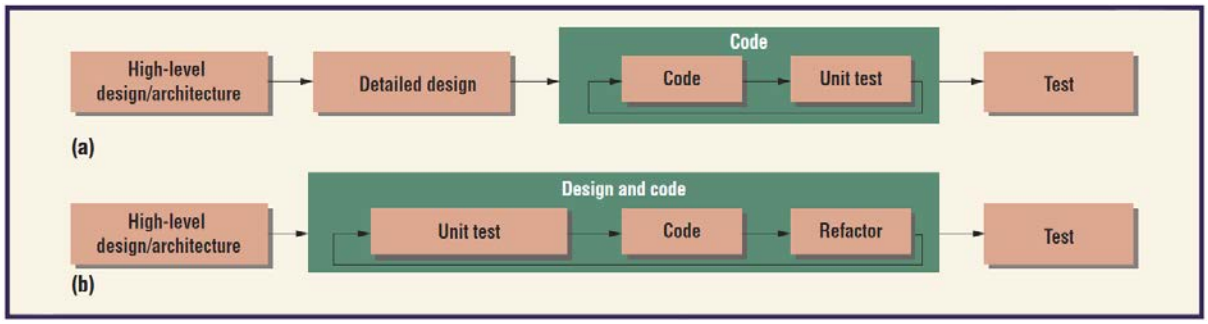

Figure 9 - Normal Development Flow versus Test Driven Development Flow [3]

The two overarching principles of TDD are [148]:

1. *Don't write a new line of code unless you first have a failing automated test; and*
2. *Eliminate Duplication.*

The order of the activities in the TDD methodology is shown in Figure 10. The order of activities can be expanded out to the 'rhythm of TDD' described by Beck [148]:

1. *Write a test*
2. *Run all tests and see the new test fail*

3. *Make a change to the code*
4. *Run the tests again and see success*
5. *Refactor to remove duplication, dependency and genericise the code.*

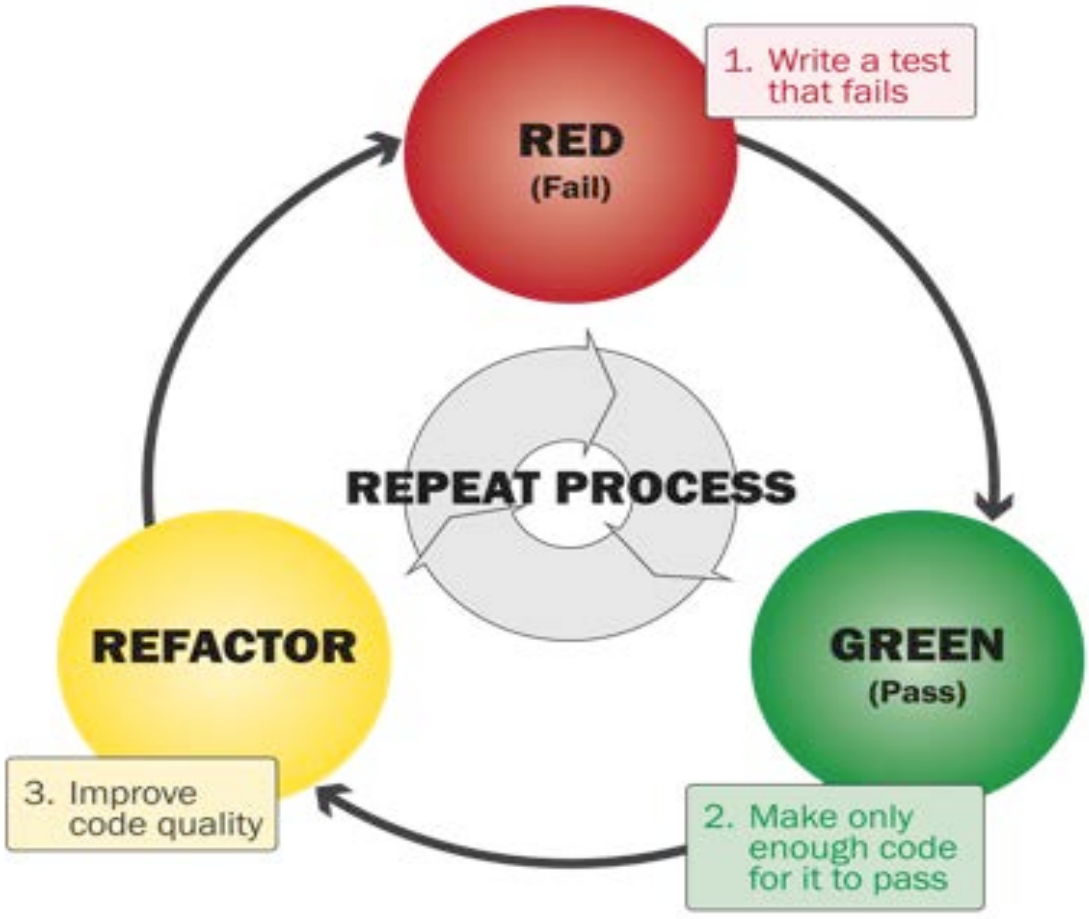

Figure 10 - Test Driven Development Cycle [4]

There are three strategies within TDD for successfully developing an artefact based on tests. The first is the 'fake it' approach, where the developer will develop a very specific solution to a test, usually primarily employing constants to give the desired answer. The refactoring process for the 'fake it' approach requires the gradual replacement of constants with variables to achieve generic functionality. The second approach is 'Triangulation'. Triangulation is a more cautious version of the 'fake it' approach in which several implementations are discretely tested and independently genericised. This process continues to a point where the redundant implementations can be deleted leaving the most appropriate generic solution to retained in the final product. The third approach is the 'obvious solution' approach in which the developer writes the code that they think is most likely the correct solution to the test.

The problem with the TDD approach in this context is that it is designed for software development and is envisioned to be applicable primarily during the encoding process. There has been no previous application of

TDD to ontology development, making it difficult to justify its use. However, the principles of TDD are sound. We can abstract them to their root and apply them to a high-level design. The application of TDD principles alone will not form a credible approach but can assist in the development of a new approach.

Chapter 4 – An Agile Approach to Ontology Development

4.1 – Introduction

There has been an increasing shift towards agile software development approaches in recent years as too many projects go over time, over budget and have failed to meet client expectations due to the rapid rate of change in the technology sector. This shift also has pertinence in the realm of ontology development. Though traditionally an ontology has been the domain of immense projects with significant resources and extensive timelines (Cyc, for example, had an initial 20 year timeline [101]) there is an emerging paradigm shift towards ontological problem solving as community ontology projects like Freebase have increased awareness and participation in the field [156]. The ontological approach to problem solving has evolved and now is not an increasingly appealing option for developers seeking a new method to fulfil client needs. The problem with the existing approaches discussed in Section 3.4.3 is that they are inaccessible to inexperienced developers and lack the flexibility and evolutionary quality asociated with agile approaches.

For this reason, the Agilitology approach has been proposed to address the identified gap of an agile ontology development methodology and enable the agile creation of a knowledge structure that is suitable to represent cyber effects in a military simulation context.

4.2 The Agilitology Approach

Independently the DSRM, *Methontology* and TDD approaches are not appropriate for use as research and development methodologies for inexperienced developers working in an exploratory manner in the field of ontology development. While thorough, the DSRM and *Methontology* approaches are targeted towards more experienced developers and larger scale development projects. They lack the flexibility to support the agile, exploratory development that better suits lower levels of experience and capabilities. The TDD approach supports this exploratory development but is not designed for use in ontologies or at an overarching architecture level, rather just being used as a part of the actual encoding process.

By extracting the principles of each of these three approaches, combining them and tailoring an agile ontology development approach termed '*Agilitology*' has been developed. *Agilitology* is grounded in DSRM's philosophical approach utilising the structure given by *Methontology* and the iterative, agile, exploratory principles offered by TDD. Figure 11 demonstrates the relationships between these approaches and how they mould together. This ontology development methodology will enable an inexperienced developer working at the intersection of multiple knowledge domains to effectively create an ontology to meet the domain requirements.

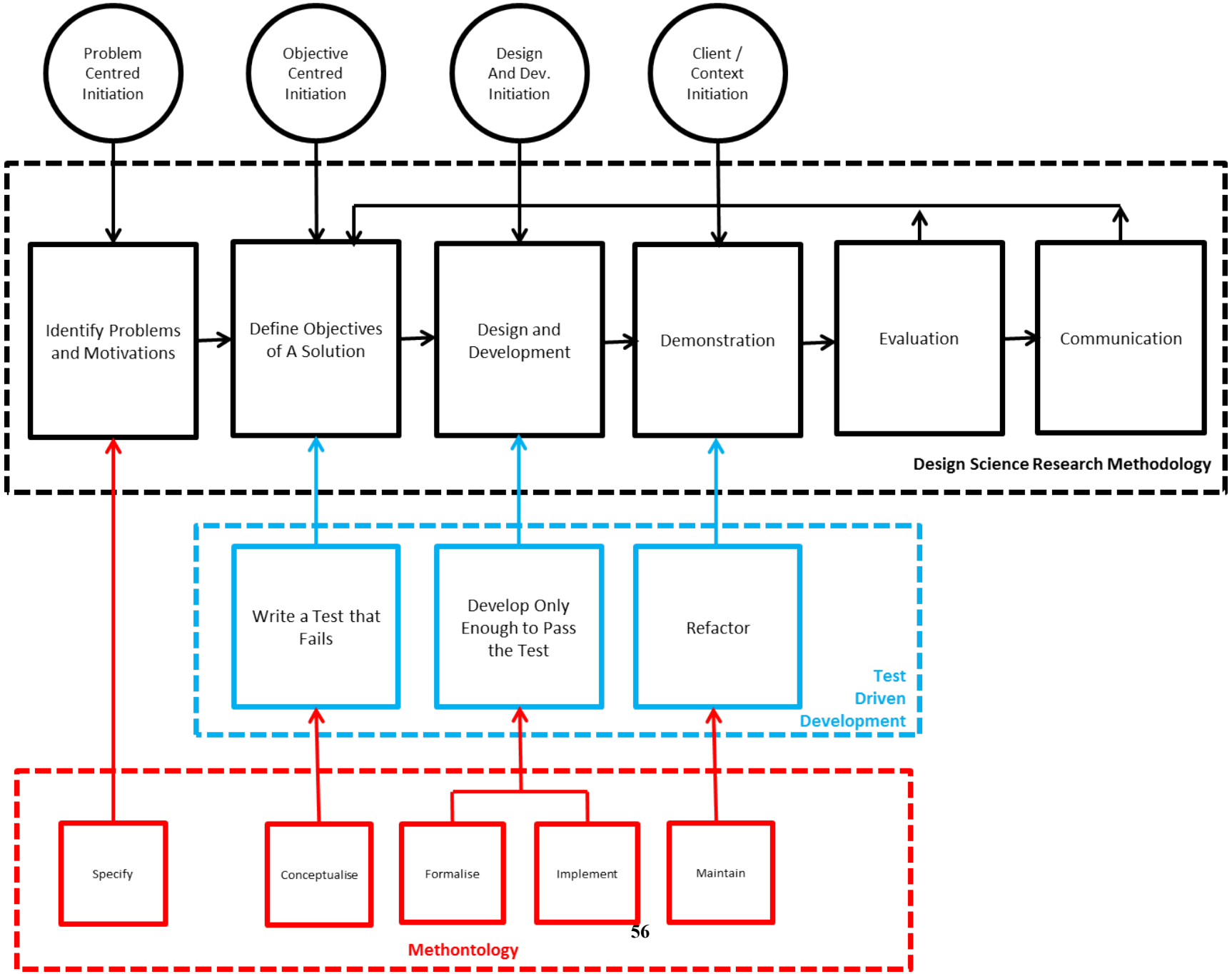

Figure 11 - DSRM, METHONTOLOGY and TDD – How the approaches map together.

The *Agilitology* approach is in Figure 12. This approach begins with a problem-centred initiation, that is, a problem in the domain of knowledge is identified and research is conducted to develop a solution. The *Agilitology* method works on two layers – the higher-level 'design' layer is based broadly on the DSRM. The 'development' layer is based more closely on the *Methontology* approach and is a subset of the design process. The TDD elements constitute 'influences' on the design and development layers as opposed to specific activities.

4.2.1 – Theoretical foundations

When considering the *Agilitology* approach in the context of the DSRM worldview the following mappings occur:

The *problem-centred initiation defines the Ontology of this approach*. The gap that is identified is problematic to a given domain (or intersection of domains) of knowledge and the development and implementation of an artefact is required to address this gap. To determine the functional requirements, we conduct an exploration of the domain of knowledge to explain the current situation and firmly identify and hence specify the expected high-level functionality of the ontology. The *Prediction* is achieved partly through the *specification* activity and partly through the *conceptualisation* of the first usecase. This usecase will define the first chunk of functionality required of the ontology. The 'test' phase of the TDD strongly influences the conceptualisation activity. The *Axiological* elements of this approach are part of this usecase – the developed solution should be aiming to achieve the given conceptualised usecase.

The *Praxeological* element of this approach is defined almost in its entirety by the development layer. The 'develop' phase of the TDD influences the *'Design and Develop'* activity. The *Design and Development* activity is the gateway between the design and the development layer. In The design and development activity, the conceptual usecase is fed from the design level into the development level. This conceptual usecase is the *specification* that the process must achieve. The specification then passes to the *development conceptualisation* activity. In this activity, the specification splits into atomic usecases that when combined will be able to satisfy the specification. The atomic usecases are then passed one at a time to the *formalisation* activity in which they are transformed into a formal representation as an ontological solution to the usecase. This formalisation activity can include searching for and implementing elements of other, existing ontologies that may solve the usecase. This formal solution is the passed to the *Implementation* activity where it is encoded in the language of choice. Finally, it is passed to the *refactoring activity*. Here the encoded solution to the atomic usecase is tested independently and as part of the wider collection of existing encoded atomic usecases to ensure functionality and compatibility. As the testing occurs any revisions, updates or improvements to the implementation occur to *maintain the integrity* of the wider ontology. If the entire specification is not satisfied then the Conceptualise, Formalise, Implement and Maintain activities will iterate. This iteration will continue until all of the atomic usecases are added to the ontology and the specified conceptual usecase is satisfied.

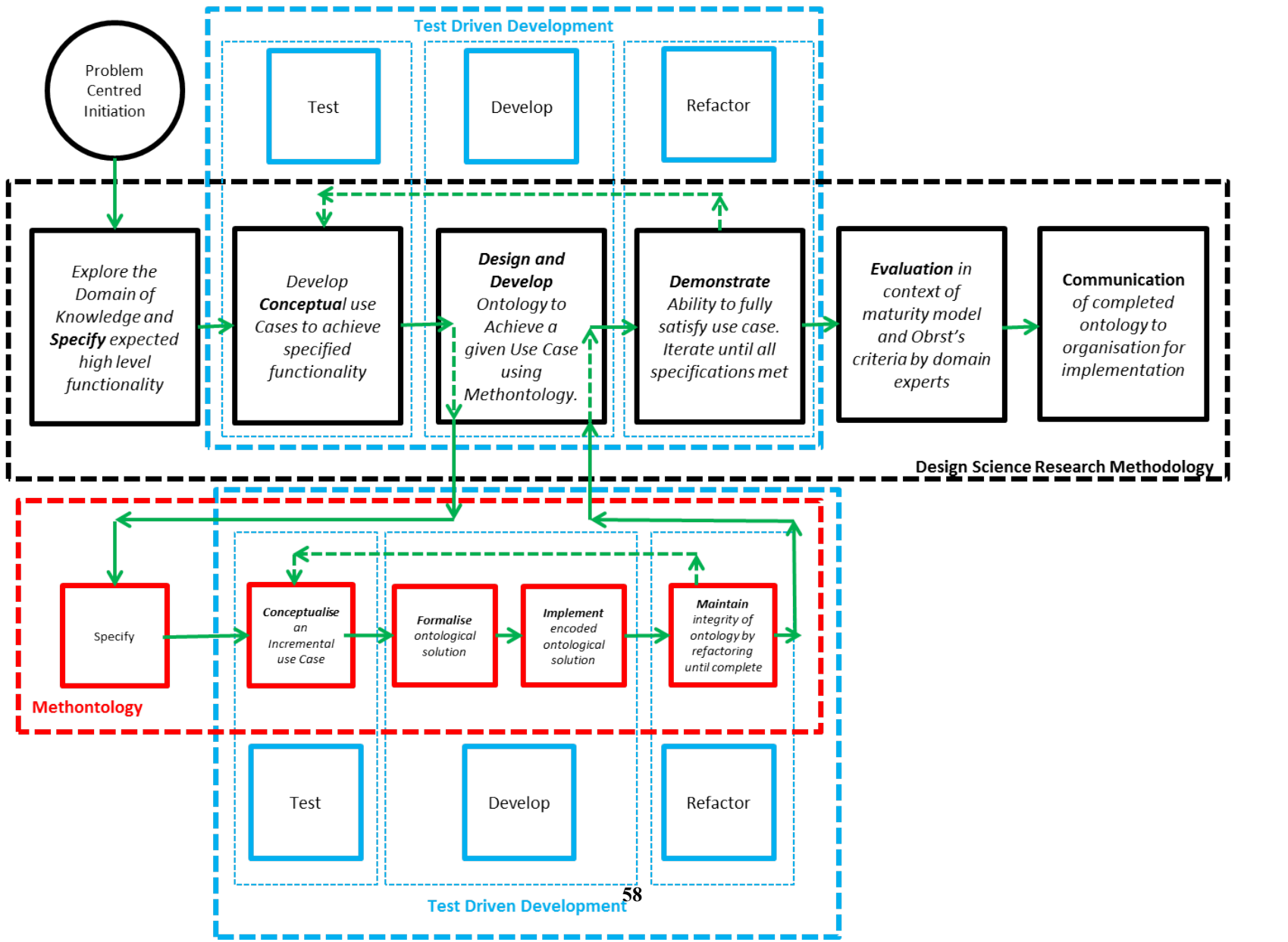

Figure 12 - AGILITOLOGY

On satisfying this specification, the development layer then passes the collection of encoded solutions back to the design layer. The encoded usecase is then passed to the *demonstration* activity where it is tested against the conceptual usecase and the ontological specifications. If the full specifications of the ontology are not satisfied, then the process will iterate back to the conceptualisation of another high-level usecase. This iteration will continue working towards achieving the specification, or aiming to add additional functionality or to solve a problem that was identified as part of the previous usecase. Once the encoded solution has demonstrated that it is capable of functionally depicting the collection of conceptualised usecases and achieves the specified functionality the ontology is ready to be evaluated.

4.2.2 – Evaluation Methodology

Traditional methods for evaluating an ontology are covered in Chapter Three. The Agilitology approach, however, is evaluated in two parts. First, is the ability for it to comprehensively represent its overarching usecase to a level that satisfies its designer. The second is the provision of this usecase to a domain expert to domain experts for approval. The conduct of the

4.2.2.1 – Evaluating ontologies based on Usecases

The evaluation of ontologies on a usecase basis is the *Agilitology* approach's informal evaluation stage. It is intended for use by the developer in two phases to determine if their developed ontology is fit-for-purpose. The first phase of this evaluation is continuous through the development and is iterative in nature. It is based on the usecase centric development approach of the ontology. As soon as the designer is satisfied that the ontology component can satisfy the incremental usecase they add it to the greater ontological structure and move on to the next component. The second phase of the usecase evaluation is the holistic usecase evaluation at the conclusion of the development process. In this evaluation, the designer will take the initial problem usecase that was deconstructed to create the component usecases and test their ontology against it. If there have been additional components that have become required during the development process as a result of changing client requirements or logical extensions of ideas then they are added to the initial usecase. There are then a number of competency questions developed about the usecase. These questions should be developed by (or approved by) the client to determine their validity. If the ontology is able to give satisfactory answers then it is considered usecase valid. The *Competency Questions (CQ)* should address each area of intended functionality. At the conclusion of the second phase of usecase evaluation the ontology is ready for release to clients or domain experts for further review, testing and evaluation.

4.2.2.2 – Evaluating ontologies based on client acceptance and feedback from domain experts.

Traditionally ontologies are built by domain experts and evaluated by domain experts for completeness, accuracy, robustness and semantic strength [94]. Though the Agilitology approach is designed to make the development of ontologies accessible to non-experts, this places a greater emphasis on the evaluation of the ontology by the clients and domain experts. Evaluation by the client as the fundamental measure of success for the design science approach is the acceptance of the product

by a client and the positive impact that this has on the conduct of business. [2, 157]. This will address whether or not the ontology solves the problem posed by the client in the first instance and how well it does this

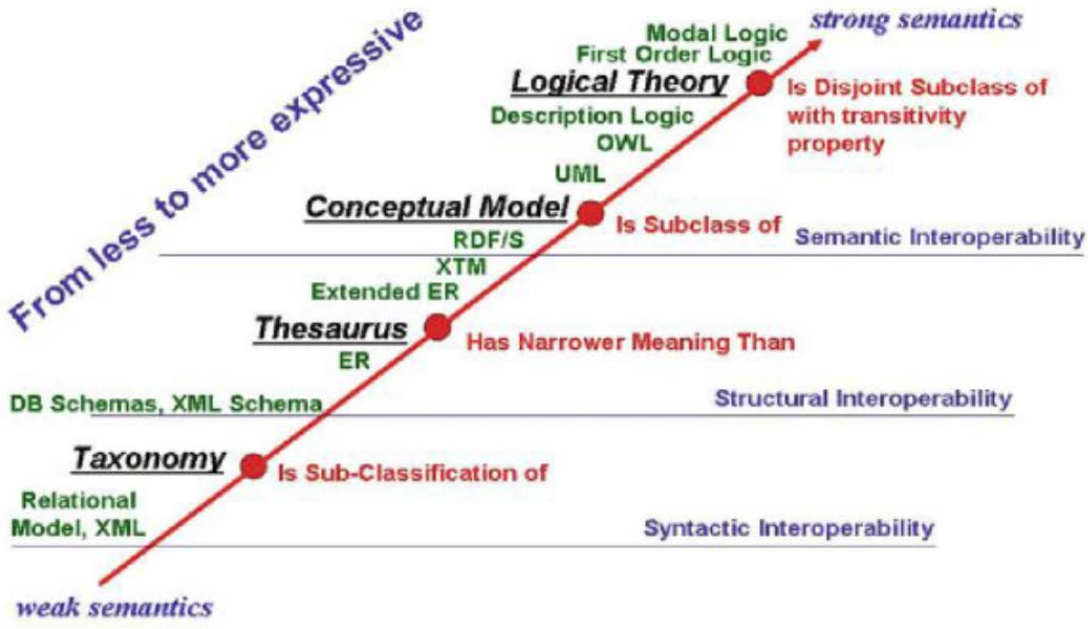

Figure 13 - Obrst's Ontology Spectrum

For expert evaluation, an ontology developed by the *Agilitology* approach will be assessed and placed on the Ontological Maturity Model (Figure 14). This maturity model was developed specifically to address the creation of community-sourced ontologies based on social media interaction of the domain users [158]. Despite its narrow scope the principles the model is built on are robust. These principles are largely drawn from Schmidt's knowledge maturity model [159] and are validated by strong correlations to Obrst's ontology spectrum (Shown in Figure 13)[90]. The experts should base their opinions on a combination of some of the possible metrics for ontology evaluation proposed by Obrst to determine where on these spectrums the ontology sits. These criteria include: *domain coverage, richness, complexity, granularity, ability to address usecases, formal properties, representation language, mappability, underlying philosophical theory, inference engine, verification* and *validation* [94]. There are, however, five recommended assessment criteria proposed by Obrst for use in the evaluation of ontologies.

1. *Expressivity of the knowledge representation language*
2. *Usecases and domain requirements of the domain ontology*
3. *Semantic agreement and consensus building*
4. *Semantic Similarity and Semantic Distance*
5. *Alignment with other ontologies.*

The assessment of these elements are split between logical, mathematic and opinion methods. Scoring based on these criteria will enable a thorough quantitative evaluation of the ontology. The aforementioned evaluation plan will satisfy both the axiological requirement that the ontology actually works as designed (has utility) and the more qualitative epistemological elements of the design relating to the warranty of the artefact to the organisation. The Evaluation by domain experts however is intended to find problems with the ontology construction itself, both structurally and content. This will help to determine the more traditional metrics proposed by Obrst [94] and will achieve the aim of having a comprehensive representation of the knowledge in the domain. As experts identify problems or gaps they are able to provide feedback to the designers to improve the ontology, extending it where needed, improving the semantics or making other alterations as required.

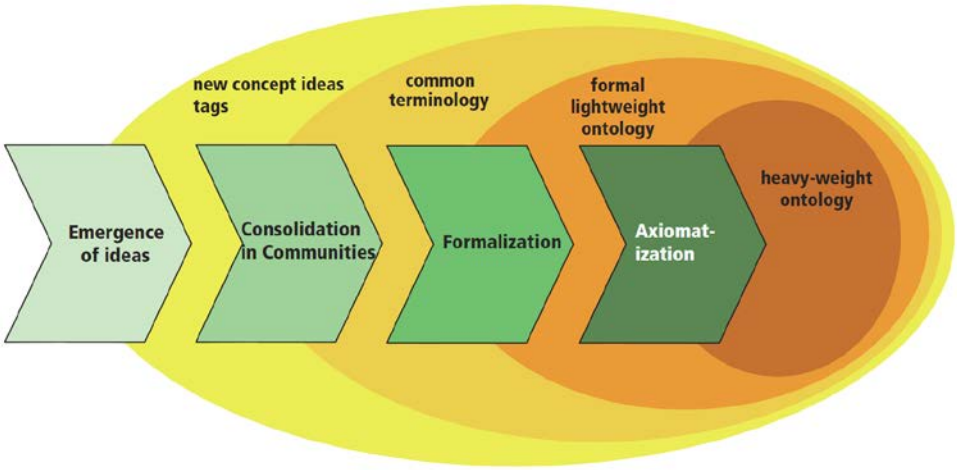

Figure 14 - Ontological maturity model

Finally, the communication element of the design will involve the implementation of the artefact into the organisation and the observation of the effects that the artefact has on the organisation. Should implementation not be possible, analysis of the realm of possibilities is also acceptable. The implementation, integration and explanation of the utility of the artefact are the key underpinning elements to this phase.

4.3 - Summary

The *Agilitology* approach produces an ontology that has been built in an exploratory manner with a minimalist approach to satisfy the usecases. This minimalist approach will eliminate superfluous information or functionality, simplifying the ontology, reducing complexity and adhering to Gruber's [91] *Clarity* Principle. Building based on integrated usecases in an iterative manner will reinforce the *coherence* of the ontology. As the conceptual usecases can be added *ad-infinitum* the ontology design process is suited to extensibility and will produce *extensible* ontologies in a manner that resembles natural growth. The *encoding bias* is the weakest element of this development approach. The selection

of an encoding prior to the completion of the design is a-typical in ontology development. If this is a cause of great concern to the developer they can elect to not encode the formalised ontology during the development. Instead, they can add each atomic usecase element to a concept tree, verb diagram or other conceptualisation tool, encoding only after all of the usecases are formalised. This is not the preferred method as it does not maximise the ability to conduct continuous testing. The trade-off between encoding bias and continuous testing was made due to the target user group of this methodology – inexperienced developers who may not have the capacity to recognise design flaws that are apparent to experienced knowledge engineers. The *Ontological Commitment* will naturally vary on a case by case basis.

The *Agilitology* approach combines the most useful principles of DSRM, *Methontology* and TDD and is capable of producing an ontology that is complaint with Gruber's principles. It will serve as a good starting point for a novice ontology developer who is developing an ontology across one or many domains in an exploratory manner. It is useful in instances where the developer is determining the substance of the problem at the same time as they are developing the solution. This approach will help to constrain the ontological scope and reduce complexity – addressing two of the biggest issues in ontology development. There is a significant potential for the automation of many elements of this approach in the future. The most relevant current research into this area is in developing ontologies based on controlled natural language competency questions [160, 161]. As the process becomes increasingly automated, the user will require less and less knowledge about the construction of ontologies – eliminating a stakeholder from the development process and reducing the complexity of developing ontologies in a cross-domain setting.

The Agilitology approach effectively addresses the research subquestion of "*What is a suitable agile, usecase centric development methodology that can be used to develop a suitable knowledge structure?"* This work has produced a methodology of ontology development that applies a design science philosophy and TDD mindset to the *Methontology* approach to ontology development to produce the *Agilitology*, the first agile ontology development methodology. This methodology will be demonstrated through the remainder of this thesis, used to construct an appropriate knowledge structure to represent cyber effects in a military simulation context.

Chapter 5 – The Cyber Effects Simulation Ontology

5.1 Introduction to the Cyber Effects Simulation Ontology

This chapter proposes a knowledge structure to answer the question "*What is a suitable knowledge structure to support the representation of cyber effects in a simulation context?".* Analysis undertaken in Section 2.4 and Section 2.5 concluded that the most suitable knowledge structure for this purpose is an Ontology.

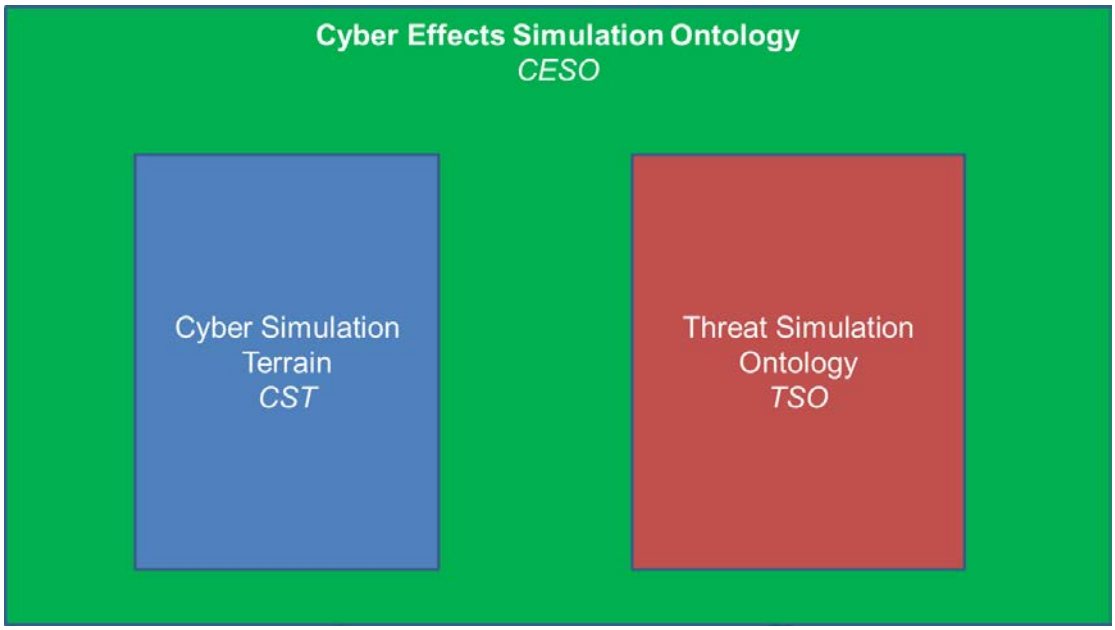

Figure 15 - Conceptual Model for the Cyber Effects Simulation Ontology

The Cyber Effects Simulation Ontology (CESO) is a knowledge representation that exists to bridge together subordinate ontologies that describe the cyber domains from discrete, ontologically committed positions. Figure 15 is an extremely high overview of this schema. The two subordinate ontologies to the CESO are the Cyber Simulation Terrain (CST) and the Threat Simulation Ontology (TSO). The CESO also links to some ontologies not in the cyber domain, leveraging elements of the Event Ontology and Organisational Ontology of a related project outside the scope of this thesis to enable its full functionality. The CESO also has a set of concepts that are used to facilitate the representation of cyber effects as an abstract concept not inherently part of either the CST or the TSO.

5.1.1 Purpose

The purpose of the CESO is to represent effectively cyber effects on military networks in a simulation context. The role of the CESO itself in this is to facilitate bridging between discrete subordinate ontologies in the Cyber Domain and to other ontologies outside the domain that support the function of the CESO. The CESO must be able to represent effectively the current state of a network and threats to develop the *Situational Awareness* of the decision maker as discussed in Section 2.3.2. It must then be able to elicit the emergent phenomena of the network, towards determining plausible, probable and preferable future states as discussed in Section 2.3.3. When complete, the CESO will be able to prove

its competency by answering a set of competency questions, generated from the analysis of a relevant usecases.

5.1.2 Development Approach

Broadly, the development of the CESO utilises the *Agilitology* process defined in Chapter 4. The approach is initiated based on the analysis of a problem in a given domain. In this context, the review of relevant literature in Chapter Two clearly illustrates the needs for a Cyber Effects Simulation knowledge structure to facilitate futures planning and resilience building. The ontology will be developed incrementally, beginning with core functionality and adding until the Ontology is capable of functioning in a comprehensive, transparent manner. Each incremental development is gated by a usecase and associated competency questions. Competency questions are used to determine when an ontology has achieved a satisfactory level of representation by using metrics of expressivity and accuracy of results [95] as judged by the developer.

5.1.3 – Specification of representation requirements and high-level functionality

The CESO must be capable of supporting the representation of cyber effects in a military simulation context. The CESO must be comprehensive and cover the appropriate elements in the domain of knowledge to maximise its utility without unnecessary verbosity. The CESO will be able to represent an interconnected cyber network with enough detail to enable the effective mapping of vulnerabilities, exploits and threats. The vulnerabilities and exploits are to be taken from existing knowledge bases and used to add granularity to the representation of these concepts within the ontology. The ontology must be able to depict accurately the links between a given attack and the effect it will cause on infrastructure.

Due to the issues discussed at length in Section 2.4 and Section 2.5 regarding ontological commitment, the need for consistent worldviews and the identified issues in Section 2.6 arising from ontologies trying to do too many incompatible tasks it is apparent that a single Ontology is not sufficient to represent the CESO. Instead, the proposed solution is based on the work examined in Section 2.6.1 and Section 2.6.2. By splitting the CESO into two sub-components - an ontological network model and a threat focused cyber ontology - the CESO itself can act as a bridging structure. Characteristically, this puts the CESO an enabling role and promotes an appropriate level of granularity to be represented as a part of a consistent worldview in the sub-ontologies, increasing their robustness and decreasing the complexity of a the knowledge structure as a whole.

5.1.4 – Implementation

Part of the proposed *Agilitology* approach requires the design to be implemented and tested. The CESO has been successfully using the Resource Description Framework (RDF) [154] Turtle (TTL) [134] syntax with basic elements of the Ontology Web Language (OWL) [155] to build the Classes, Properties and Relationships of the ontology. The choice of language is incidental to the development process and implementation of the CESO is possible in any suitable semantic web language. After building the schema, the ontology is then instantiated in RDF using a graph triple store database,

Stardog [162]. Queries are performed using the Sparql Protocol And Rdf Query Language (SPARQL) [163]. The querying is used to check that the elements instantiated into the ontology are correct and then extends toward generation of plausible future states.

5.2 – High-Level Use Case

The *Agilitology* approach requires that usecases be created to bound and guide the development of the ontology. The component usecases derive from an overarching problem usecase. The overarching problem that has been used to design the CST is shown in Figure 16. Figure 16 is taken from the paper: *System of Systems: Cyber Effects Simulation Ontology* [164]. This paper defines the need for the CESO in the context of an Offensive Support (OS) request for artillery fire against a hostile target. The focus of the CESO is the effective representation of the elements attributed to the *Virtual* layer of this usecase. The following component usecases have been developed by designing the infrastructure that would best support the activities depicted in the problem usecase above.

The usecase illustrated in figure 16 describes a scenario there a Joint Fire Team (JFT) requests artillery support. In this simplified example, a JFT Forward Observer (FO) sends a *Call For Fire* message. A Joint Fire Coordination Centre (JFCC) subsequently conducts safety checks, confirms the priority of a mission and approves the Call for Fire request. The JFCC also manages a fire mission queue of approved requests ranked by priority. The JFCC then assigns an Artillery Battery a fire mission from the queue. The battery fires a salvo to fulfil the fire mission request.

The Figure 16 usecase depicts how a Fire Mission can be conceptualised across four discrete but interrelated domains. The *Physical*, *Virtual*, *Conceptual* and *Event* domains. The Physical Domain encompasses all of the ‘tangibles’ of the model with shared attributes such as the ability to be seen, touched and interacted with. For example, a physical asset could be a soldier, a howitzer or a laptop computer. The Virtual Domain is where the CESO resides. The virtual domain contains ‘virtual assets’ that are each separately defined by their configuration. The effective representation of the virtual domain is the key development outcome for the CESO. The Conceptual Domain encompasses all of the enabling activities of the model that allow it to function in a meaningful way including processes, organisations, collections and a number of abstraction capabilities. The Event Domain is where the ‘activity’ occurs. The event domain will log the changes in the state of the assets of the physical and virtual domain and reconcile this with the conceptual domain to create a narrative of occurrences.

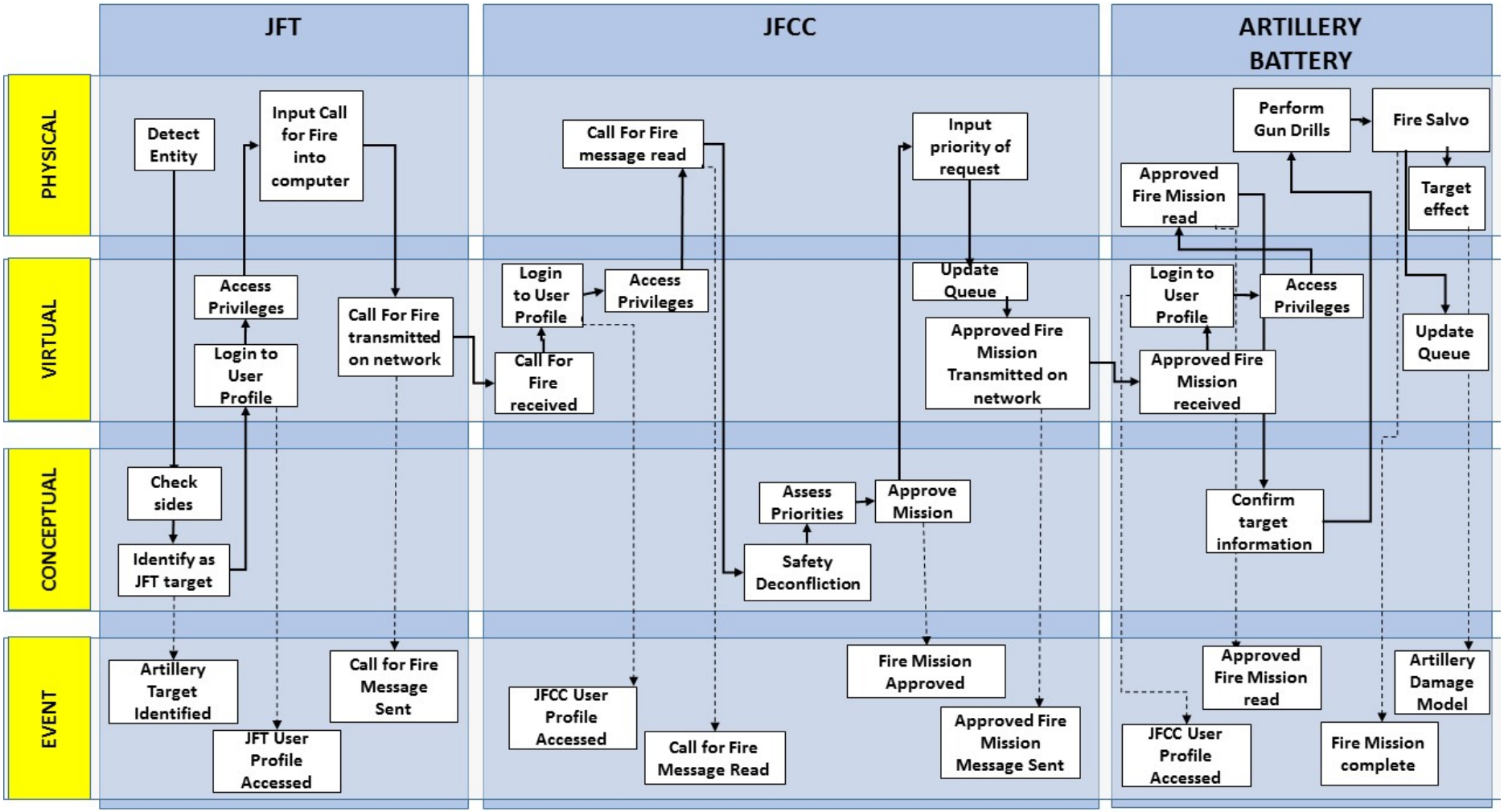

Figure 16 - Problem usecase to inform development of the CESO

The following description of the usecase depicted in Figure 16 is quoted from the *System of Systems: Cyber Effects Simulation Ontology* paper it was proposed in [164]. In this description, each concept and activity is accompanied by an indicator of the domain they reside in for the usecase. (P) = Physical Domain, (V) = Virtual Domain, (C) = Conceptual Domain and (E) = Event Domain.

"A fire mission commences after a JFT entity (P) has detected (P) another entity (P) which is recognized as a member of the opposing force (C) and is a suitable artillery target (C, E). The JFT (P) must login (P, E) on their Computer (P) through their User Account (V) with appropriate configuration settings such as Privileges (V), Decision Rights (C), Skillsets (C) and Assets (V, P). A Call For Fire message (V) is sent (E) from the JFT to the JFCC. The JFCC entity (P) can receive the message (V) after the message has navigated through a network (P) and the underlying network architecture (V). After logging into the system (P, V) the JFCC entity's allocated configuration (V, C) allows it to access the message (V), read the message (P), make a decision (C) and approve the request (V, E). This process includes an assessment of the priority of competing requests for offensive support from different JFT entities. Based on the priority of each specific target to the mission, decisions are made to allocate each fire mission to an Artillery Battery organisation (C), which consists of some Guns (P, V), managed in a Fire Mission Queue (V). The Call for Fire message (V) and Fire Mission Approval (V) are managed through a series of messages and links on physical devices such as computers (P, V), network devices (P, V) and a network architecture (V, C). The artillery battery subsequently performs a series of physical actions to fire the guns and achieve a kinetic effect on a target (P). The fire mission queue (V) is updated to reflect the completion of the assigned fire mission (V)."

The role of the CESO is to represent effectively the *Virtual* elements of the above-described usecase. Through the iterative decomposition of this usecase by the *Agilitology* ontology development method proposed in Chapter 4, this usecase will guide the development of an can represent the effects of a cyber attack in a military simulation context.

5.3 – The Cyber Simulation Terrain

The Cyber Simulation Terrain is the CESOs network infrastructure knowledge representation. This section will clearly articulate the purpose of the Cyber Simulation Terrain enumerate the requirements of the schema based on the decomposition of the problem usecase stated above and identify the novelty in the approach throughout this section.

5.3.1 – Purpose

The purpose of the Cyber Simulation Terrain is to provide the ontological network model for the CESO. The ontological network model is designed to represent the underlying infrastructure that supports ordinary business operations. The perspective of the cyber terrain is affiliation-agnostic. It represents both friendly and hostile infrastructure. In order to represent cyber attacks and the effects on 'ordinary business' the infrastructure that supports 'ordinary business' must be effectively represented.

5.3.2 – Requirements and design considerations

To facilitate the representation of ordinary business activities, an understanding of the underlying infrastructure is required. These requirements have been enumerated through a survey of the relevant literature, including related work in the area (Section 2.6.1), broader domain knowledge ([32, 145-147] and additional commercial information. Details of the applicability of these requirements are specified as part of the usecases that follow.

5.3.3 – Usecases

The following usecases have been developed through a systematic deconstruction of the problem usecase presented in Section 5.2. This process has created eight discrete areas of attention to inform the development of the Cyber Terrain Schema.

5.3.3.1 – Usecase 1: Nodes and Networking

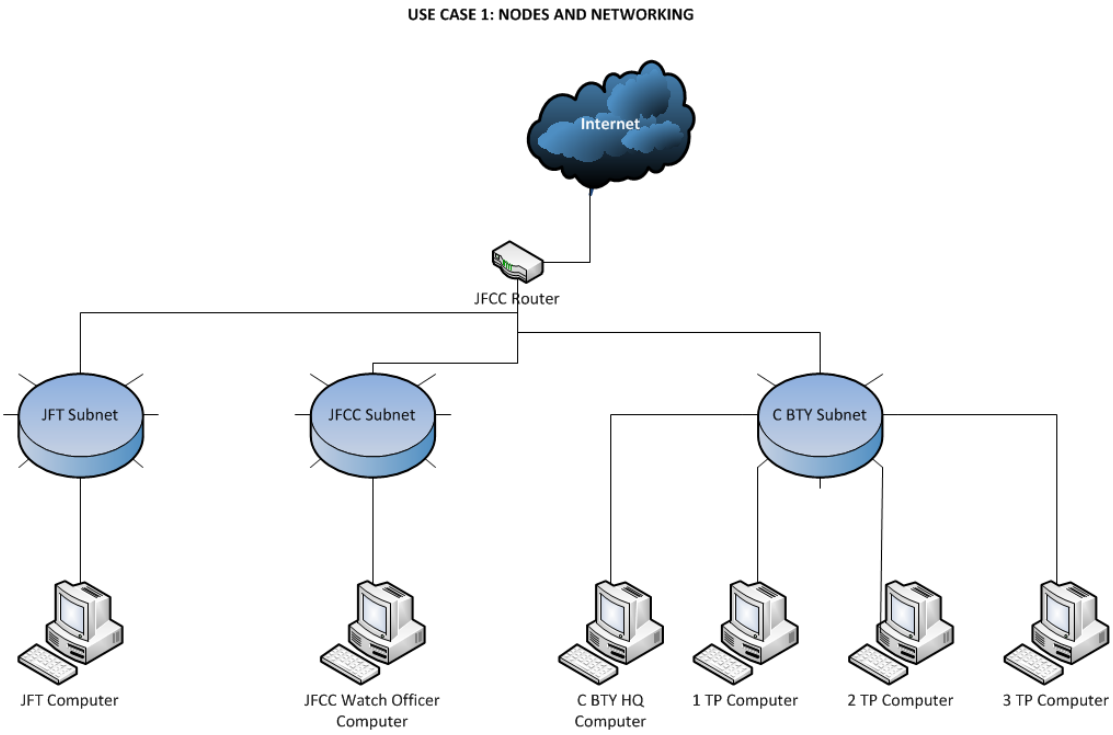

Figure 17 - USECASE 1: Nodes and Networking

The problem usecase describes three distinct groupings: the Joint Fires Team (JFT), the Joint Fires Communication Centre (JFCC) and an Artillery Battery (Arty Bty). This particular Arty Bty is C Bty from the 1337th Artillery Regiment. The JFT is responsible for *Intelligence, Surveillance, Target Acquisition and Reconnaissance* (ISTAR). JFTs are detachments that deploy forward of the main force to support land combat by directing accurate Offensive Support (OS) (such as artillery fire, air strikes or naval gunfire support). The JFT maintains a communications backlink to the JFCC. The JFCC is the C2 hub for the mission, communicating forward to the JFT and passing instructions to the Arty Bty. The Arty Bty is comprises of a headquarters and three distinct Troops. Each artillery troop has two howitzers that are used to apply fire under the direction of the JFCC to targets identified by the JFT. A

robust communications system underpins the interconnection of these three troops and their headquarters. Figure 17 illustrates this usecase.

For this usecase, the core infrastructure elements can be deduced from the required functions of each grouping. Each of the three groupings will likely be in a geographically dislocated position. Therefore, each grouping will be designated an independent subnetwork by the traditional logic of a LAN being confined to a single geographical area [146, 147]. The three subnetworks will be required to communicate and must, therefore, be connected. This usecase assumes a common WAN is shared by the three and administered from the JFCC, where the router is most likely to be located. The WAN likely has internet accessibility (or at least, a link to a strategic network) to enable the JFCC to communicate with higher headquarters. The JFT is a small, mobile unit. Therefore, it is unlikely that they will be carrying much equipment and are hence represented as possessing a single computer. The JFCC is likely to possess a large number of computers. For this usecase, only the watchkeeper's computer will be represented. An Arty Bty covers an expansive area on the ground. To facilitate cohesive action, they are grouped into troops (TP) – 1 TP, 2 TP and 3 TP. Each troop has a commander who is responsible for communicating with the Arty Bty headquarters to receive mission taskings. To support the passing of information between each of these groupings each is assigned a computer. The proliferation of computers through the command structure will network the command and control capability of the Battery Commander (BC), improving their ability to communicate and pass orders quickly and effectively to subordinate call signs.

Based on this analysis of the required infrastructure, an ontology is required to represent effectively the internet, routers, networks, subnetworks and computers depicted in the above usecase to support the mission functions detailed in the problem usecase. The schema that is implemented by the CESO to achieve this is located in Section 5.3.4.1. This usecase will be considered effectively represented when it can answer the following competency questions:

1. *Which Nodes are visible in the ontology?*
2. *Which Networks are visible?*
3. *Which Subnetwork does each node belong to?*
4. *What is the IP address of each node on the network?*

5.3.3.2 – Usecase 2: Software and Services

To complete a mission, each of the three groupings (JFT, JFCC and C Bty) must fulfil a prescribed role. Usecase 1 outlined the ISTAR role of the JFT, the C2 function of the JFCC and the OS role of C Bty. Effectively completing these tasks in a networked environment is highly likely to rely on software applications. The requirements of each role will require different application. Limits on Computation and funding arising from the *Constrained* future operating environment [18] dictates that minimal software is installed on each system. In support of the JFT, it is likely that software to assist in the targeting of hostile forces will be present, complemented by software to communicate this targeting information back to the JFCC. The JFCC will require software to effect command and control of the

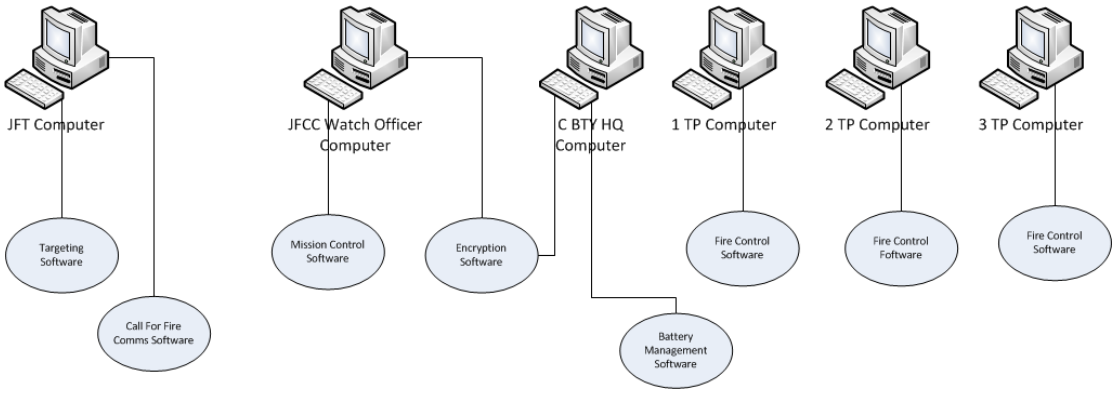

Figure 18 - USECASE 2: Software and Services

fire mission. The JFCC will also be capable of encrypted communication. For this usecase, the JFCCs encrypted link is between itself and the headquarters (HQ) of the Arty Bty. C Bty HQ needs to be able to push orders to, and receive reports from the three troops it commands. C Bty HQ will utilise battery management software to facilitate this requirement. Each troop commander will be required to translate the orders that they receive from HQ into rounds being fired out of howitzers to achieve battlefield effect. They will need to employ fire control software to perform the necessary calculations and calibrations to ensure the accurate application of fires. Figure 18 is a visual depiction of this usecase.

Competency questions arising from this usecase that the Schema implemented in Section 5.3.4.2 will be required to answer are:

5. *What software is running on which machine?*
6. *Which services are remote and what are they projecting?*

5.3.3.3 – Usecase 3: Vulnerabilities and Weaknesses

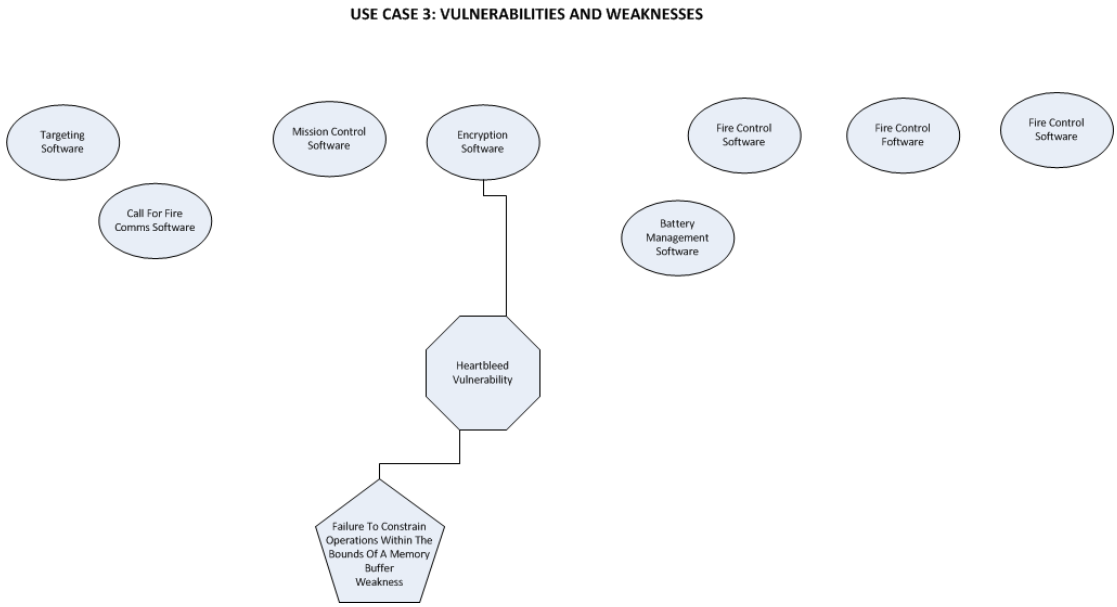

Figure 19 - USECASE 3: Vulnerabilities and Weaknesses

The problem usecase introduced in Section 5.3.2 doesn't explicitly specify any vulnerabilities requiring representation. The intended use of the CESO is to represent cyber effects. Meeting the intent of the

CESO implies a need to conduct attacks and observe the results. A prerequisite for the conduct of an attack is the presence of a targetable vulnerability. A vulnerable instance of software will need to link to the vulnerability it exhibits and also a broader categorization of vulnerabilities. To create a meaningful usecase, the *Heartbleed* vulnerability is present in the encryption software used to secure communications between the JFCC and C Bty HQ. This usecase is shown in Figure 19. Competency questions that must be answered by the Schema developed in Section 5.3.4.3 to satisfy this usecase are:

7. *Which nodes on the network are running vulnerable software?*
8. *What weakness encompasses this vulnerability?*
9. *How difficult is it for an attacker to exploit this vulnerability?*

5.3.3.4 – Usecase 4: Domains and Users

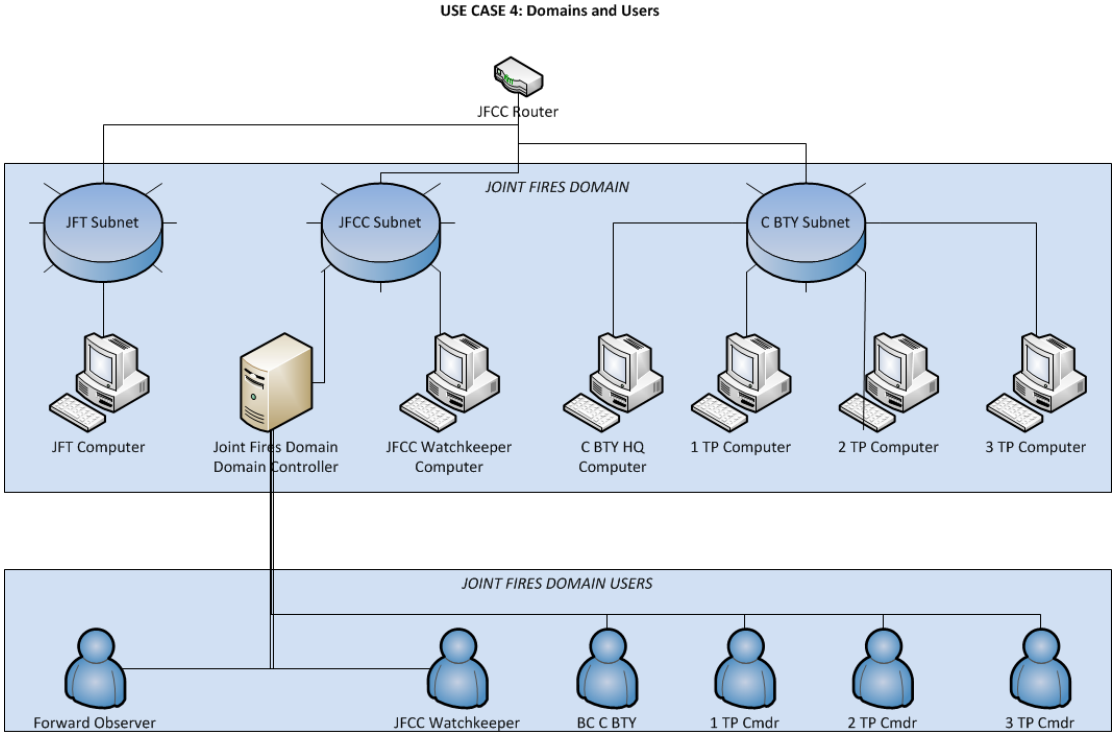

Figure 20 - USECASE 4: Domains and Users

The JFT, FJCC and Arty Bty described in the problem usecase all have unique, interdependent roles that must be fulfilled to achieve their assigned mission. Each of these roles still requires a degree of human interaction. The CESO will be required to represent the *cyber-personas* [24]of human operators through their interaction with the system. This interaction will most commonly take the form of a *User Account.* Figure 20 shows that the JFT, JFCC and Arty Bty as part of a *Joint Fires Domain* will be required to accommodate a range of user needs and specialist roles. The JFT will need a *Forward Observer* to operate the software running on the JFT computer for targeting and communication with the JFCC. The JFCC will have a *Watchkeeper,* who is responsible for monitoring communications and actioning requests for fire that are received by the centre. C Bty will have a *Battery Commander,* who controls the headquarters. Each troop in C Bty will have a *Troop Commander,* who is responsible for controlling the firing of the howitzers under their command and communicating with C Bty HQ.

Users are rarely implemented locally on enterprise networks; in the context of a recent push to adopt an enterprise architecture for military networks [165, 166] it is likely that this will be the situation in this usecase. Most user systems are implemented through a *domain* servive. The domain is a collection of organisational assets that perform a similar function, independent of their physical or logical network location. The *Joint Fires Domain* is applied to all computers represented in the usecase to this point. The *Joint Fires Domain* is likely to be administered by the JFCC as part of their Command and Control facilitation functions and hence the JFCC network will be the most likely host for the *Domain Controller*. Based on this usecase the ontology is required to represent users, domains and the associated infrastructure. In order for the schema implementation in Section 5.3.4.4 it must be able to answer the following competency questions:

10. *Which computers represented in the ontology are members of a domain?*
11. *Can a domain user access any node that is a member of the domain?*
12. *Who are all the users of a given domain?*
13 *Which users are administrators and which are normal users?*

5.3.3.5 – Usecase 5: Firewalls, Antivirus and Intrusion Detection Systems

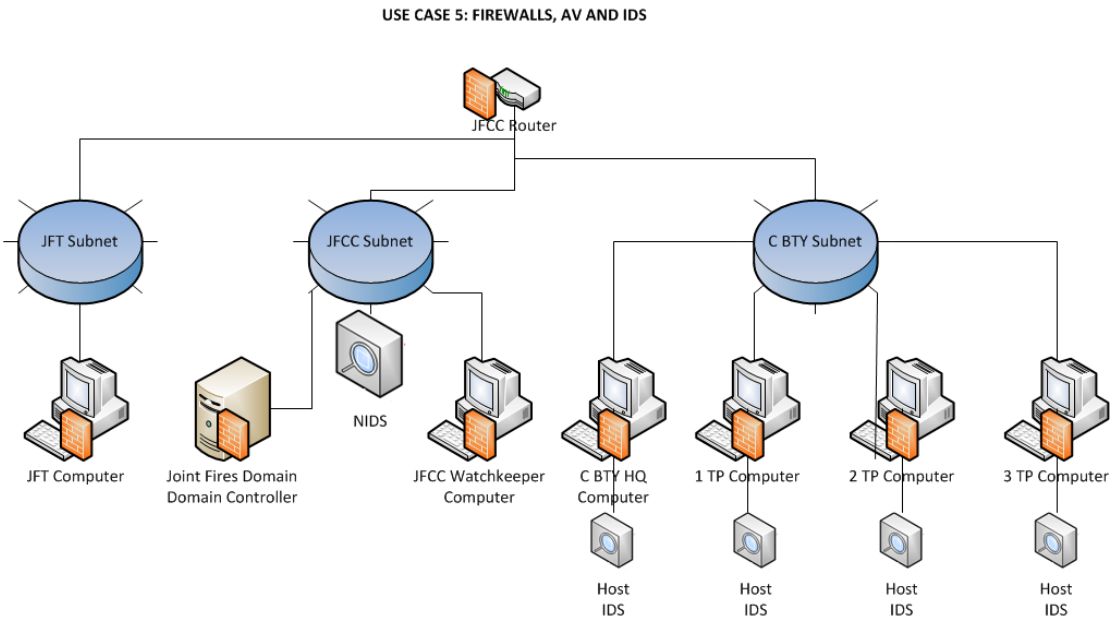

Figure 21 - USECASE 5: Firewalls, Antivirus and Intrusion Detection Systems

Protective measures such as *Firewalls, Antivirus* and *Intrusion Detection Systems* are not explicitly mentioned in the problem usecase defined in Section 5.3.2. Consistent with the analysis regarding *Enterprise Architecture* in Section 5.3.3.4 it is assumed for this usecase that there are protective measures in place on the network. Each node on the network will be protected by its own firewall, reflecting the near ubiquitous adoption of endpoint protection. The JFCC, as the C2 hub for the Joint Fires Network, is likely to be the most heavily protected and is in possession of a *Network Intrusion*

Detection System (NIDS) for this usecase. It is also probable that the JFCC would be running *Host-based Intrusion Detection Systems* (HIDS), though, for the sake of the balancing the usecase, these have been allocated to the member nodes of C Bty.

Successful representation of this usecase will require the inclusion of firewalls, HIDS and NIDS into the CST. The implementation must be capable of detecting malicious activity and performing some action in response to this. The CST representation of the firewall must also have the capacity to deny remote access to the nodes that they are protecting, filtering attempted connections to restricted ports. Figure 21 is the visual depiction of this usecase. It is implemented into the CST Schema in Section 5.3.4.5. The competency questions that the implemented ontology must be able to answer are:

14. *Which nodes are running HIDS?*
15. *Which subnetworks are running NIDS?*
16. *Can a HIDS detect a vulnerability or exploit present in the terrain?*

5.3.3.6 – Usecase 6: Data, Disks and Encryption

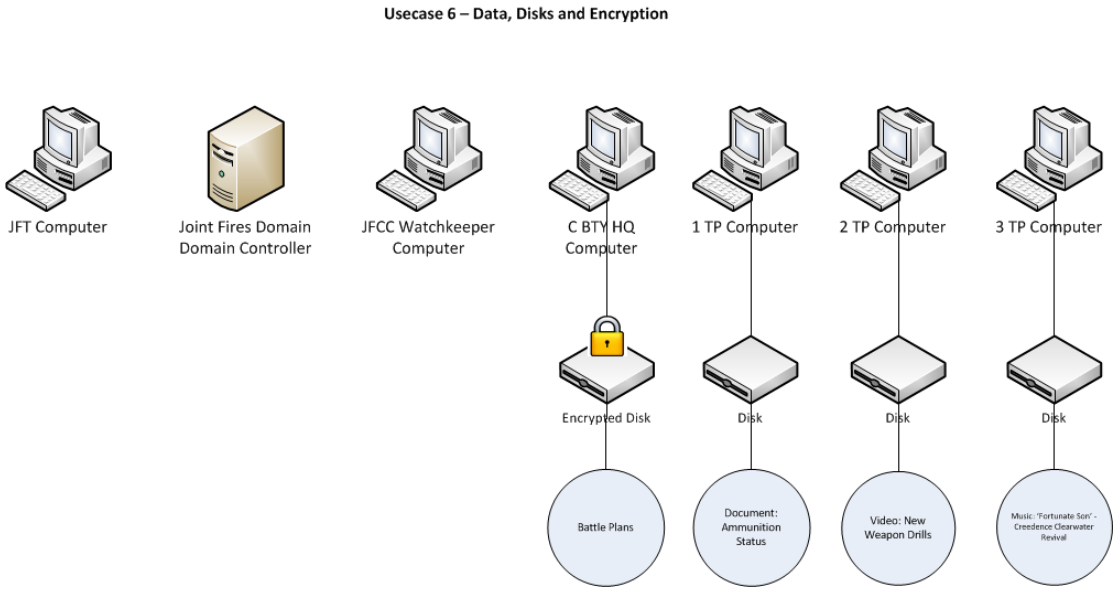

Figure 22 - USECASE 6: Data, Disks and Encryption

The creation and storage of data as shown in Figure 22 is implicit in the problem usecase described in Section 5.3.2. By creating a message to send a request for fire, the JFT is creating data. When the JFCC calculates the priority of the call for fire and performs a safety deconfliction, the JFCC creates data. As the Arty Bty fires a salvo they need to track ammunition stores, and firing adjustments – this creates data. The *Data, Disks* and *Encryption* usecase therefore requires that a disk is assigned to each computer in C Bty. The disk for the HQ computer will contain Battle Plans and the disk for 1 TP will contain a document detailing ammunition status. 2 TPs disk will contain a video file that is in use by the troop and 3 TPs disk will contain personal music that was left on there by one of the soldiers who

has used the computer. The HQ disk containing the battle plans will be encrypted. The schema implemented in Section 5.3.4.6 must be able to answer the following questions in order to progress:

17. *Which computers have storage disks?*
18. *Which disks have Data?*
19. *Which disks are encrypted?*
20. *Who owns item of data X?*

5.3.3.7 – Usecase 7: Wireless Connectivity

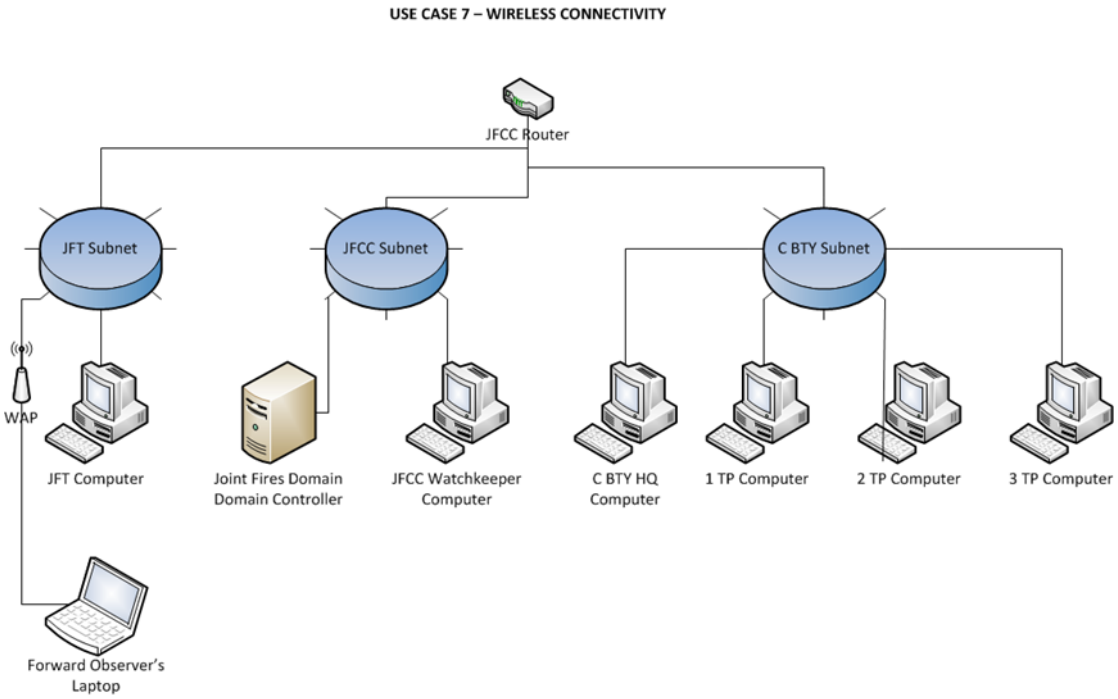

Figure 23 - USECASE 7: Wireless Connectivity

One of the activities depicted in the problem usecase located in Section 5.3.2 requires an FO to identify a target and lodge a *Call for Fire* request to the JFCC. Missions undertaken by an FO typically require them to be constantly changing location. To increase speed and stamina, an FO will aim to travel unencumbered. An FO must also have the ability to rapidly pass information back to the JFCC for actioning. Based on these criteria, it is a reasonable deduction that a wireless network connection would be used to permit freedom-of-movement for the FO and instant access to communication channels. The infrastructure required to facilitate this activity is a portable wireless-capable device for the FO, a wireless access point for the wireless device to connect to and a link from the wireless access point to a communications channel to the JFCC. For this usecase, it is assumed that the FO has a laptop computer that is connected to a wireless access point using an IEEE 802.11 wireless networking implementation. This usecase will be employing the *Wired Equivalent Privacy* (WEP) protocol for the wireless connection. This is visualised in Figure 23.

To satisfy this usecase, a wirelessly connected device should be indistinguishable from a physically connected one. In the context of network administration, tools like NMAP [130] will not return an inherently clear summary of which devices on a network are wired and which are wireless. An ontology of the cyber domain should replicate this to maintain consistency with the real world. It is not sufficient to abstract this relationship to a binary 'wireless/wired' state. The represented relationship must closely mimic reality if it is to produce accurate information about the network. Representation of this relationship must be achieved without disrupting the existing network infrastructure. Representing an intermediary '*wireless access point'* that provides connectivity to their wireless subnetwork devices through a *Wireless NIC* enables the ontology to represent additional, valuable information without disrupting the existing infrastructure. Describing wireless connectivity at this level of granularity will permit the effective representation of wireless security protocols, wireless interception attacks, interruption attacks to wireless networks and connectivity issues. The schema created to represent this usecase is located in Section 5.3.4.7. To be deemed competent, the ontology must be able to answer the following questions:

21. *Which networks are wirelessly accessible?*
22. *Does a port scan return both wired and wireless connections together?*
23. *What is the wireless security protocol that is being used to protect a given network?*

5.3.3.8 – Usecase 8: Virtualisation

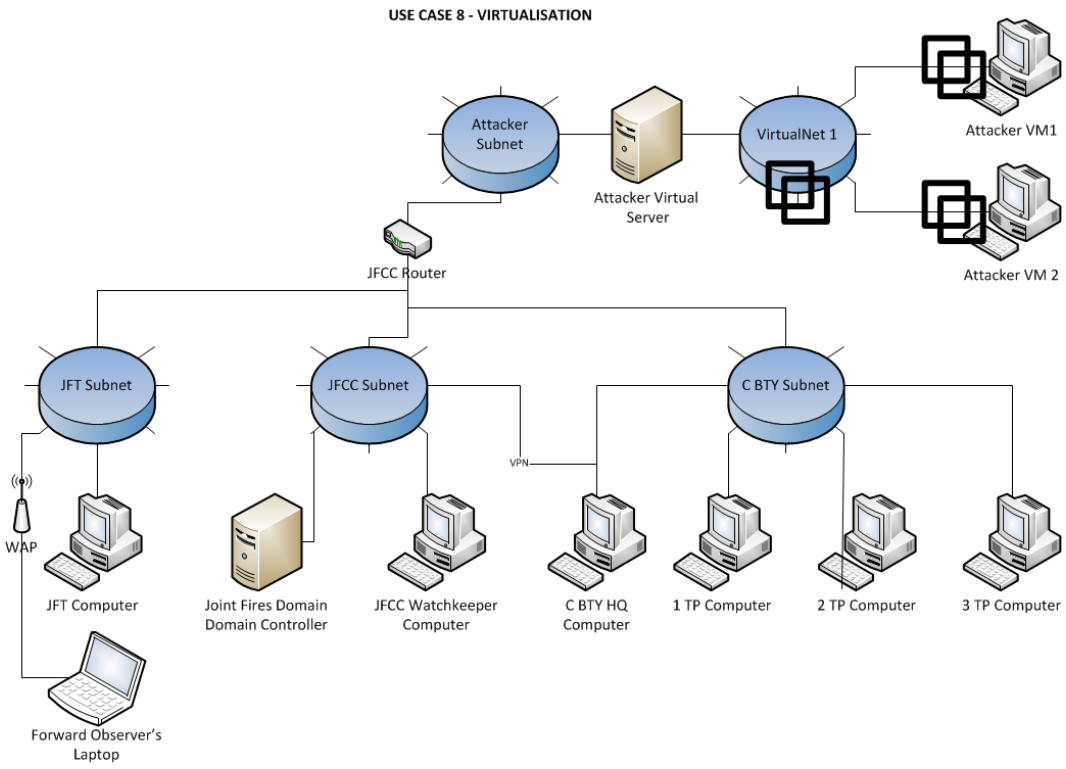

Figure 24 - USECASE 8: Virtualisation

The problem usecase defined in Section 5.3.2 does not explicitly utilise virtual machines. Virtual machines and Software Defined Networking (SDN) are increasingly significant elements of the Information Technology domain. A rise in virtualization of services to reduce risk to the physical hardware and maximise the use of computer resources is occurring. Attackers are also increasingly utilising virtual technologies to reduce risk to themselves. When the benefits of SDN intrsect with the risks posed by increasingly flexible and adaptive virtualized threat actors, the need to demonstrate the faculty of the CST to include virtualisation as a compelling usecase is established.

This usecase has specified a fourth subnetwork connected to the Joint Fires Router – a subnet dedicated to attackers. The attackers could be anyone. They could be corporate penetration testing team or professional cyber criminals or a hostile cyber warfare force. (The detail of attribution is addressed by the TSO in Section 5.4.4.) Both of the attacking nodes are virtual machines on a virtual network hosted on a physical server on the attacker's subnet. Additionally, a VPN connection has been established between the HQ C Bty computer to the JFCC Subnetwork. Figure 24 is the visualisation for this usecase. The schematic for this usecase is located in Section 5.3.4.8. To be deemed competent, the implemented ontology must be capable of answering the following questions:

24. *Which nodes on the network are virtual machines?*
25. *What are all the active VPN Connections?*
26. *Can a virtual machine be seen as part of a physical network?*

5.3.4 - CST Component Schema

The following elements of the CST schema are depicted in a pseudo-UML visualisation and accompanied with a word-picture. The visualisations depict classes (bolded terms at the top of each box), properties (the terms bulleted in the boxes) and relationships between the classes (labelled, directed arcs). The arcs are directional to encapsulate clearly the semantics of the relationships they represent. The semantic relationship between two classes is a directional from-to structure. The *Domain* or origin of the relationship connects to the *Range* or target class of the relationship. Solid lines are used to denote relationships between classes and also to represent subclassing. In each circumstance, the arc is labelled appropriately. A dashed line indicates a transitive relationship – these will be explained as they arise though the basic premise of transitivity is that if the same truth applies to successive members of a sequence it will also apply between any two members taken in order. For example, consider three objects A, B and C that are successive in a sequence:

$A > B$;
$B > C$; Therefore
$A > C$ *because* $A > B > C$.

Knowing that A is the largest object and that each successive object is smaller permits the inference that any object after A is smaller than the object that preceded it. This property is useful for describing

semantic relationships between classes. Section 5.3.4.4 and Section 5.3.4.7 contain additional information about the transitive properties represented in the schema.

5.3.4.1 – Schema 1: Nodes and Networks

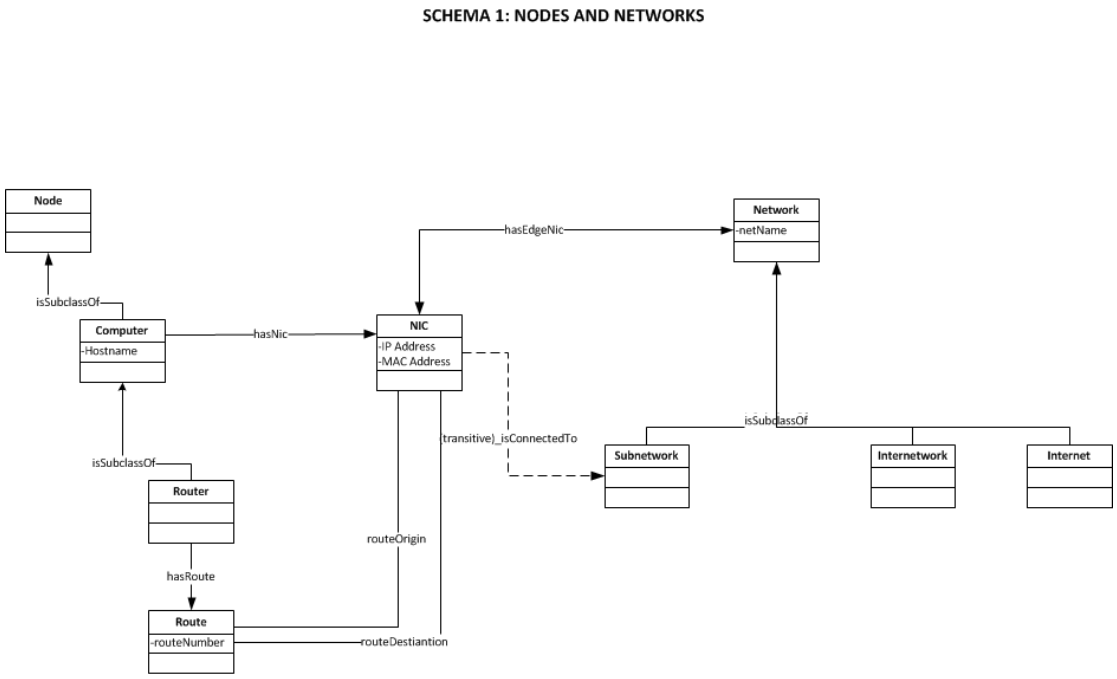

Figure 25 - SCHEMA 1: Nodes and Networking

The representation of nodes and networks is the core role of the CST. To effectively represent the concepts defined as requirements by Section 5.3.3.1 the schema depicted in Figure 25 was devised. The Virtual Terrain Project [129] inspired elements of this representation though significant alterations and improvements have been implemented to make the CST fit for the purpose of representing cyber effects in a military simulation context.

The *node* class is the superclass of the *computer* class. Node is a generic term that can allude to any network capable object. Computers have *Network Interface Controllers* (NICs). A computer can have multiple NICs. A NIC is an intermediary between computer and network that possesses the *IP address* and *MAC address* of a computer. The NIC facilitates connection to a *subnetwork.* The representation of a NIC as the connection interface to a network rather than the computer itself permits the modelling of situations where computers may have multiple NICs, hence multiple IP addresses and membership of many subnetworks. A *Subnetwork* is a subclass of a *Network*. The *Network* class is also the superclass for the *Internet* and *Internetwork* classes. The *Internet* class is used to represent the internet in the CST. It has the attributes of all other networks and hence can be used to represent individual attackers, facilitate connections to networks that are not part of the modelled LAN or WAN and also can be used to link to an attacker's network infrastructure, to permit the representation of counter-hack and active defence situations. An *Internetwork* class is a conceptual addition to the ontology that is used to facilitate the connection of *Router* objects together. A *Router* is a specialised computer that facilitates

the connection of disparate subnetworks. A *Router* will have multiple affiliated NICs (as it is a subclass of computer it inherits this relationship). These NICs represent the edge of a network and are characterised as *edgeNICs*. A *Router* has *Routes*. *Routes* define the origin *edgeNIC* and the destination *edgeNIC* of two connected *Subnetworks*. *Subnetworks* can only communicate when a *Route* exists between them. It is assumed that members of a subnet can see all other members of their subnet. Access control between computers is described in Section 5.3.4.4 and firewalling of communications between nodes in Section 5.3.4.5.

This implementation of *Nodes and Networks* is the core of the CST. Each following usecase builds on this core functionality. Implementation details of particular note from this component schema are the distinction of nodes from subnets and the use of NICs as an interface with a network rather than the computer itself. One of the major limitations on the fidelity of the Virtual Terrain family of models is the presence of *node clusters* as single entities to represent subnets [129]. In the context of high-fidelity modelling of cyber effects in a military simulation this 'node-cluster' level of abstraction is insufficient. The CST implementation is an accurate depiction of reality and hence, more suited to achieving the aims of the CESO. The use of NICs as a component of computers to connect to networks is also an improvement on existing terrain models. Current models do not permit the representation of a computer with multiple network connections and multiple IP addresses. The inability to represent multiple IP addressed per computer severely limits other models, particularly when attempting to represent the pivoting stage of an attack, a fundamental step in the attack cycle [9].

5.3.4.2 – Schema 2: Software and Services

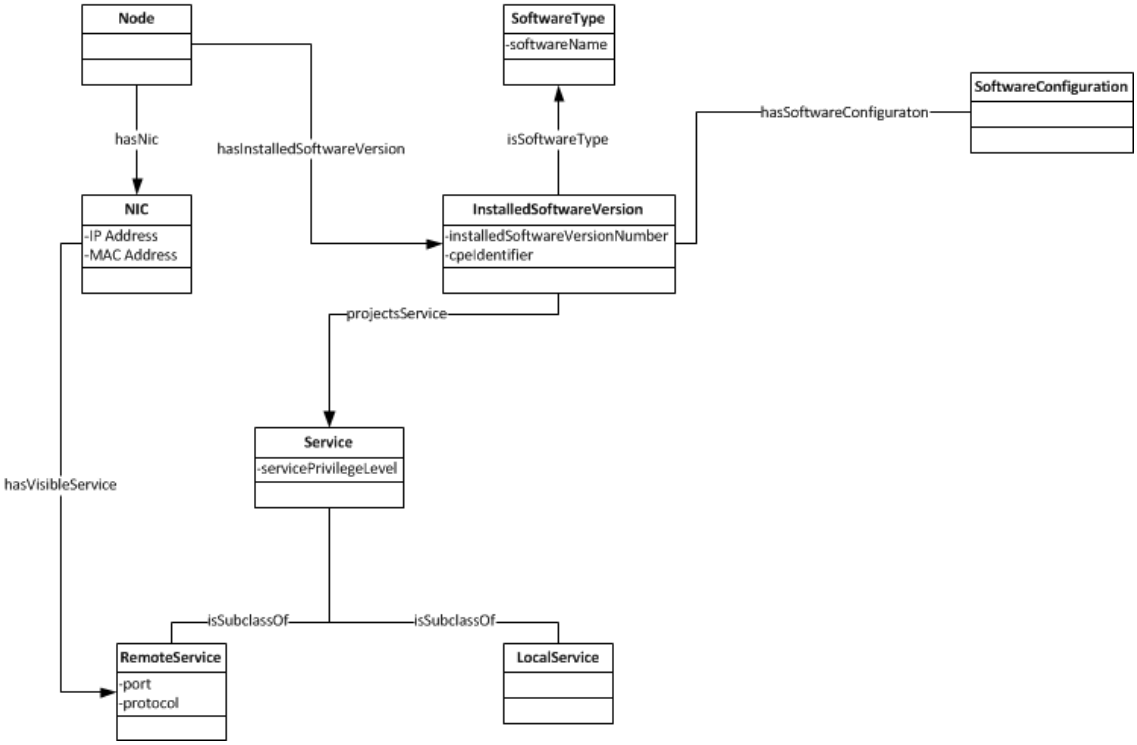

Figure 26 - SCHEMA 2: Software and Services

The representation of software in the CST is an evolution of the service tree construct from the *Virtual Terrain* family of representations. The service tree concept has been altered to give it a flatter structure that will facilitate more efficient querying. The software and services component schema introduces two major concepts to the CST – *Software* and *Services*. *Software* associates with a node as an *InstalledSoftwareVersion.* Enumerating software versions for all possible installations is difficult. To reduce the effort required to add these concepts the *Common Platform Enumeration* standard from MITRE [116] will be adopted and instantiated into the ontology. An installed version is of a *SoftwareType*. The *SoftwareType* is an informational concept, used to facilitate querying of the CST for software generically, providing richer results than can be achieved by querying by version. *Services* are projected by an *InstalledSoftwareVersion* when running on the system. *Services* are subclassed into *Local Services* and *Remote Services*. A *Local Service* runs directly on the node that hosts it and does not interact with the network. A *Remote Service* is one which interacts across the network using TCP, UDP or another transport protocol. *Remote Services* are visible from the *NIC* of the *computer* that the *InstalledSoftwareVersion* projecting them is installed on. The *InstalledSoftwareVersion* permits association with custom *Software Configurations* to specify additional information about the *InstalledSoftwareVersion* when applicable (further discussion on *Software Configuration* is in Section 5.3.4.5 .) Figure 26 is the visual depiction of the Schema.

The depiction of *Software and Services* as a flatter structure than the Virtual Terrain models is an improvement that shifts away from traditional taxonomic perceptions of software used by the Virtual Terrain family's *Service Tree* implementation [129] to an approach firmly grounded in reality. A computer is only 'aware' of the version of the software that it is currently operating. The direct association between the *node* and *InstalledSoftwareVersion* classes reflects this. Grouping by software type, usage, manufacturer or any other collection is a human structure that is imposed to enforce order. The CST permits the use of these structure with the *SoftwareType* class, the key difference between approaches being that the *SoftwareType* isn't part of the standard query chain that would represent most usage of the CST, this enhances efficiency without sacrificing human readability.

5.3.4.3 –Schema 3: Vulnerabilities and Weaknesses

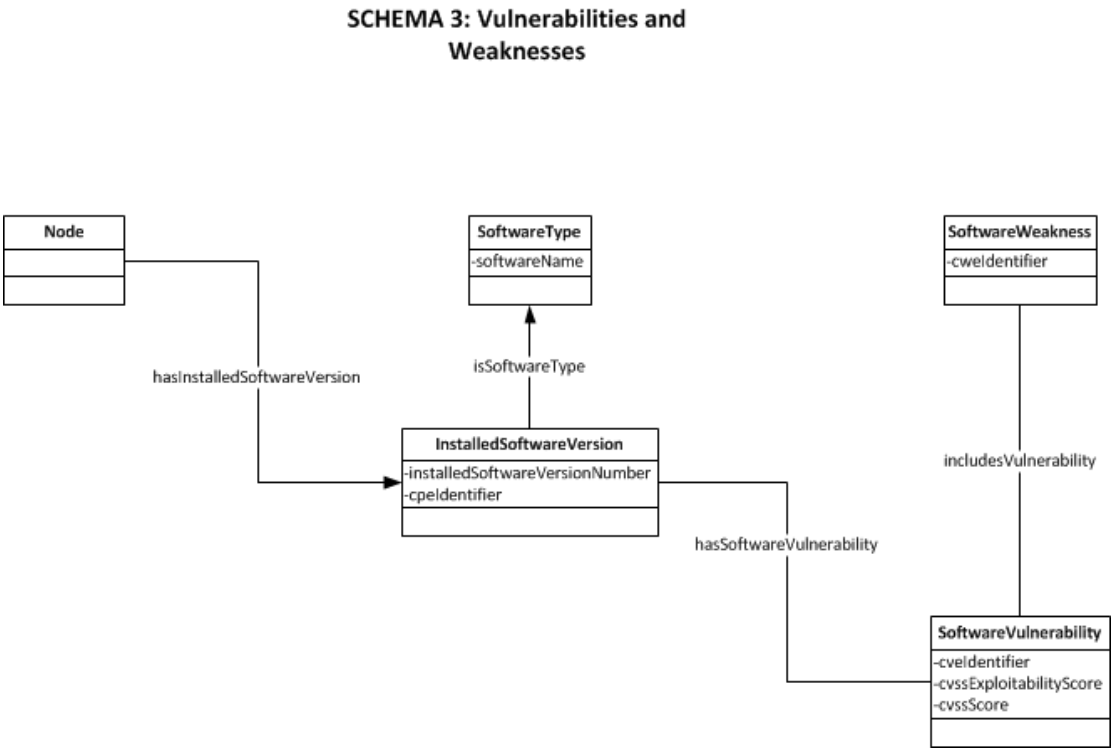

Figure 27 - SCHEMA 3: Vulnerabilities and Weaknesses

Schema 3 (depicted in Figure 27) introduces two new concepts to the CST Schema. A *Software Vulnerability* is a flaw in an *InstalledSoftwareVersion* that will permit it to be exploited by an attacker. *SoftwareVulnerabilities a*ssociate with a *Common Vulnerability and Exposure* (CVE) [118, 119] identifier. Linking to a known CVE record that will provide details about the vulnerability present in the *InstalledSoftwareVersion*. The CVE identifier will include details of two key metrics defined as properties of the *SoftwareVulnerability*. The *Common Vulnerability Scoring System* (CVSS) [167] property and the CVSS Exploitability score sub property are metrics from 1 to 10 (1 is least severe, 10 is most) that reflects the severity of the vulnerability they are associated with. The intended use of the *cvssScore* metric is as a ranking mechanism for system vulnerabilities, allowing a decision maker to assess the vulnerabilities present and plan remediation based on known severity metrics. The *cvssExploitabilityScore* is representative of the ease of exploitability. Numbers approaching 10 are easier for an attacker to exploit.

The *SoftwareWeakness* class fulfils a similar function to the *SoftwareType* class from Schema 2. It is intended to facilitate advanced querying and improve human readability of ontological outputs. The design choice to place it as an informational class removed from most functional querying follows the same logic as *SoftwareType* and will improve efficiency. The distinct link from software to vulnerability is vital for the effective functioning of the CESO. Section 5.5.2 discusses in detail the use of the vulnerability as a linking point to the *Threat Simulation Ontology*.

5.3.4.4 – Schema 4: Domains and Users

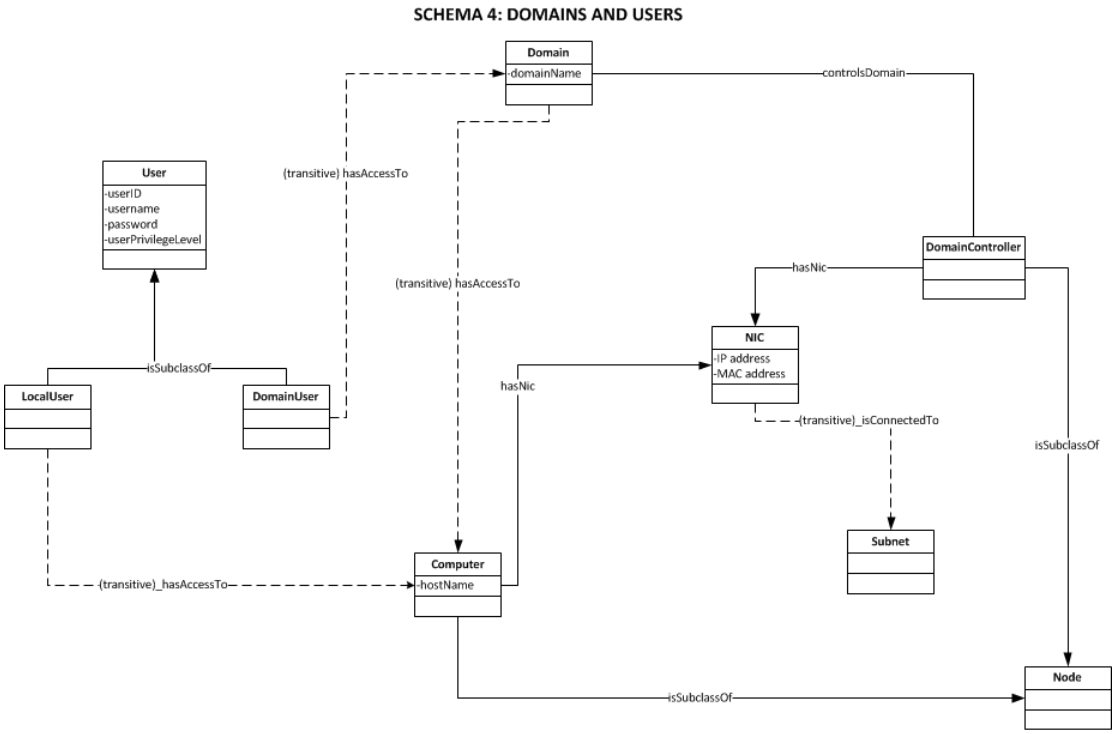

Figure 28 - SCHEMA 4: Domains and Users

Figure 28 is the visual depiction of the CSTs implementation of domains and users defined in Section 5.3.3.4 into an ontological structure. This schema introduces the *Domain, DomainController, User* and User subclasses to the CST. The *Domain* class is the core of this schema, associating users and computers together beyond geographical and topological limitations. *Nodes* are members of a *Domain* on an individual basis. A *Computer* may be a member of zero to many domains. A zero to many relationship enables the representation of scenarios where a *Computer* is a member of multiple domains simultaneously. Simultaneous *Domain* membership can then be used to represent nodes with multiple levels of classification on them (for example, an unclassified, secret and top secret domain all accessible form a single computer) The *Domain* has a *domainName* property that is used to uniquely identify the domain. A *DomainController* is a subclass of *Node* that is a member of a single subnetwork, but able to control a domain across a broader network structure.

A *User* has the properties of *userID, username, password* and *userPrivilegeLevel.* These properties enable the representation of authentication activities. The *User* class is the superclass of *LocalUser* and *DomainUser*. A *LocalUser* does not associate with a *Domain*. A *LocalUser* has access to a *Computer*. A *Computer* may or may not be a member of a domain. The *LocalUser* is only able to access *Computers* that they are explicitly granted access to. A *DomainUser* has access to a *Domain,* and through the *Domain* has access to any *Computer* on the domain. The *transitive* nature of the *hasAccessTo* relationship is the facilitator of this capability.

The fundamentals of *transitivity* are explained in Section 5.3.4. The *transitive* relationship that allows domain users to access any computer on the domain is predcated on the following logic:

LocalUser hasAccessTo Computer .
DomainUser hasAccessTo Domain ;
Domain hasAccessTo Computer ; Therefore
DomainUser hasAccessTo Computer; Because
DomainUser hasAccessTo Domain hasAccessTo Computer.

This *Transitive* relationship is implemented in the CST using the *OWL:transitiveProperty* relationship[1]. This creates a situation where the *hasAccessTo* relationship has multiple domains. The issue of multiple domains is addressed by the addition of an *OWL:disjointUnionOf* property[2]. The disjoint union property enforces that a relationship with multiple domains can only be instantiated as one of them at a time. In this context, *DisjointUnion* property enforces that a class that *hasAccessTo* another class cannot be in the *range* of A *DomainUser* and a *Domain* simultaneously.

The CST schema for *Domains* and *Users* effectively implements the requisite classes, properties and relationships to represent comprehensively *Domains* and *Users* in an ontological format. The transitive properties used to facilitate an equivocal relationship with computers for *LocalUsers* and *DomainUsers* simplify the queries requiring the elicitation of information from the CST and improving the accuracy of the results. The implementation of *Domains* on a per-node basis provides a flexible structure that enables the representation of scenarios where there are multiple domains of different restriction accessible from the same computer. This representation also allows the representation of incidents like information cross-contamination and data-spill. These capabilities are not available in any of the existing publicly available terrain models.

1 http://www.w3.org/TR/owl-ref/#TransitiveProperty-def
2 http://www.w3.org/TR/owl2-syntax/#Disjoint_Union_of_Class_Expressions

5.3.4.5 – Schema 5: Firewalls, Antivirus and Intrusion Detection Systems

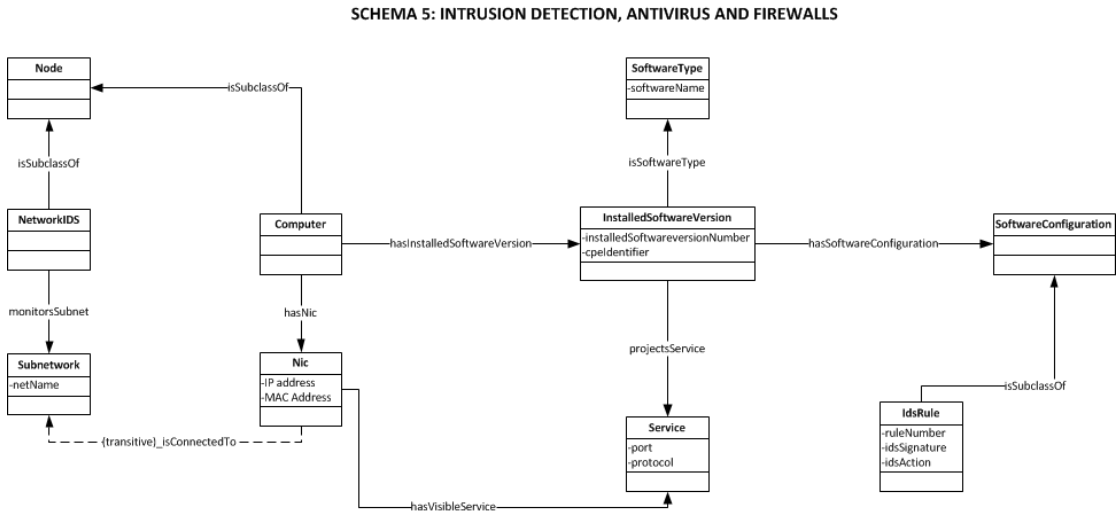

Figure 29 - SCHEMA 5: Firewalls, Antivirus and Intrusion Detection Systems

Figure 29 is a visualisation of the Schema implemented in the CST for the representation of Firewalls, Antivirus and Intrusion Detection Systems. Section 5.3.3.5 stated the requirements for the effective representation of these concepts in the CST. Resultant from this is the addition of two new classes to the CST. *NetworkIDS* as a subclass of *Computer* and *idsRule* as a subclass of *SoftwareConfiguration.* The *NetworkIDS* class is intended to monitor the traffic on a given subnetwork. The CST does not currently implement accurate modelling of network traffic and, as a result, the capabilities of this representation are limited. The *NIDS* is a special subclass of *Node* that does not connect to a *Subnetwork* through a *NIC.* The connection to the *Subnetwork* is implicit in the *monitorsSubnet* relationship. A NIDS will typically be co-located with the network gateway or switch. Network switching is an implicit activity in the CST and hence the *NIDS* is also implicitly present.

Antivirus and HIDS are implemented simultaneously in the CST. Antivirus and IDS are both software artefacts. Consequently, the are implemented as an *InstalledSoftwareVersion* with a particular SoftwareConfiguration called an *idsRule.* An idsRule is used to tune the IDS software to detect intrusion attempts. The *idsRule* achieved this through the specification of three properties: *ruleNumber, idsSignature* and *idsAction.* The *ruleNumber* is a unique identifier for each rule, allowing rulesets to be pre-built, stored and linked to as applicable. The *idsSignature* is used to detect the presence of an intrusion attempt on the system that the IDS is monitoring. This *idsSignature* will typically be specified as the *CVE* identifier [119] of the targeted vulnerability or the exploitDB identifier [168, 169] for the exploit that targets the vulnerability. Current data stores mapping the two efforts exist [170] and provide the most effective way to representthe detection intrusions on the system. An intrusion will be detected during the querying process. If a node that is being exploited is running IDS software with a rule that looks for the exploit used it will trigger an *idsAction.* An *idsAction* is the response that the IDS produces when triggered. While fully customizable, the recommended responses are *"ALERT",* "LOG", *"ACTIVATE MTNDM"* and *"IGNORE"*. Standard responses permit smoother interaction with

applications utilising the CST. The CST implements a Signature-based IDS approach. There is currently no utility gained by implementing anomaly based IDS into the CST. The CST is currently not capable of representing false alarms using the existing schema. Given the significant focus of discussion around the rates of false alarms in IDS [35-39] it is an inherent part of IDS functionality. An accurate representation of an IDS will be required to represent false alarms. However, in the CST the current implementation of an IDS will check the CVE of the software it is running against the exploitDB identifier of the exploit and see if they map to each other schematically. If they do, the IDS action is triggered. False alerts are generated by activity occurring on networks and nodes. The activity of the CESO is controlled in the linked 'Event Ontology'. To implement a false-alarm occurring on an IDS, a 'false-alarm' event type would be created in this domain which would then be used to generate false alarms in the IDS. To facilitate this representation, the CST will need to be updated to include a 'false alarm rate' property of IDS software. This number would be used in conjunction with the false alarm event by whichever application is querying the schema to effectively use the capability. As the Event Ontology is out of the scope of this work, false alarms have not been included in the representation

.

Firewalls are implemented implicitly on the CST. The design of the *RemoteService* concept dictates that only ports that are accepting connections be visible to the network. If the ports are not visible, they are not accessible. The firewall representation is a limited implementation and future work will include the enhancement of this concept to permit a more granular *Firewall* implementation.

The CST implements protective measures in three ways: Firewalls, Network IDS and Host-based IDS. The firewalls represented in the CST are implicit, and will be a priority for future work. Network IDS currently do not have network traffic to inspect and are left essentially redundant, beyond being a placeholder for a future iteration of the CST. Applications utilising the CESO can use the NIDS heuristically to modify an attractiveness of a subnetwork – though the specifics of that implementation are out of the scope of this work. The CST implements Host-based IDS as an instance of software running on a node. This is a shift from existing terrain models which either use a Boolean value to indicate whether a node is an IDS sensor [79, 80] or have sensors as independent entities that are members of networks closely associated with a particular node. The CST implementation proposed in this section is the most consistent with reality and, therefore, suitable for use in the wider CESO to represent cyber effects in a military simulation context.

5.3.4.6 – Schema 6: Data, Disks and Encryption

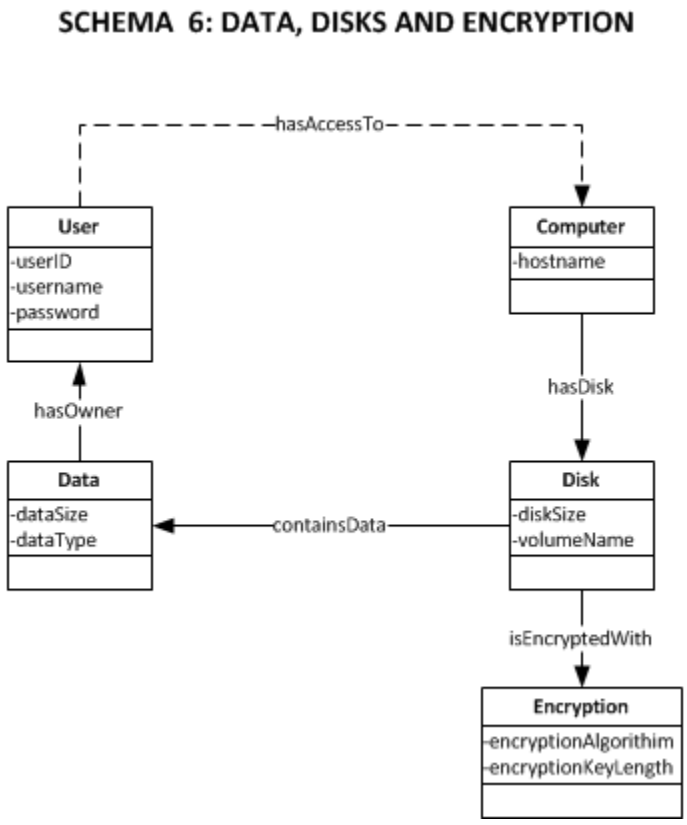

Figure 30 - SCHEMA 6: Data, Disks and Encryption

The CST implements an ontological design of the information infrastructure required to represent the effects of a cyber attack in a military simulation context. Figure 30 shows how a feature of the CST not implemented in previous terrains is the inclusion of disks storing data. This schema introduces the concepts of Data, Disks and Encryption. To effectively represent the usecase forwarded in Section 5.3.3.6, a computer can possess none to many *Disks*. The *Disk* class introduced by the CST has *diskSize* and *VolumeName* properties. A *Disk* can be *Encrypted* through an association with an *Encryption* algorithm.

The *Encryption* class has the properties of *encryptionAlgorithm* – representing the name of the algorithm use to encrypt the disk and *encryptionKeyLength* to determine the strength of the encryption. Should an application utilising the CESO or CST intend to model situations such as *brute forcing* a password, metrics including *encryptionKeyLength* become useful for calculations. *Data* are stored on *Disks* and has an *owner*. The *owner* of *Data* is a *User,* who can authenticate and read the data. Any user attempting to access the data not authenticated as the owner will not be able to read it.

The inclusion of *Disks* and *Data* improves on existing terrain models. Existing models largely do not support the representation of *Disks* and *Data* – focusing on aspects of the model that impact on *availability* and *integrity* rather than *confidentiality*. An application using the CST can use data as targets to motivate covert penetration of systems and provide targets for attackers.

5.3.4.7 – Schema 7: Wireless Connectivity

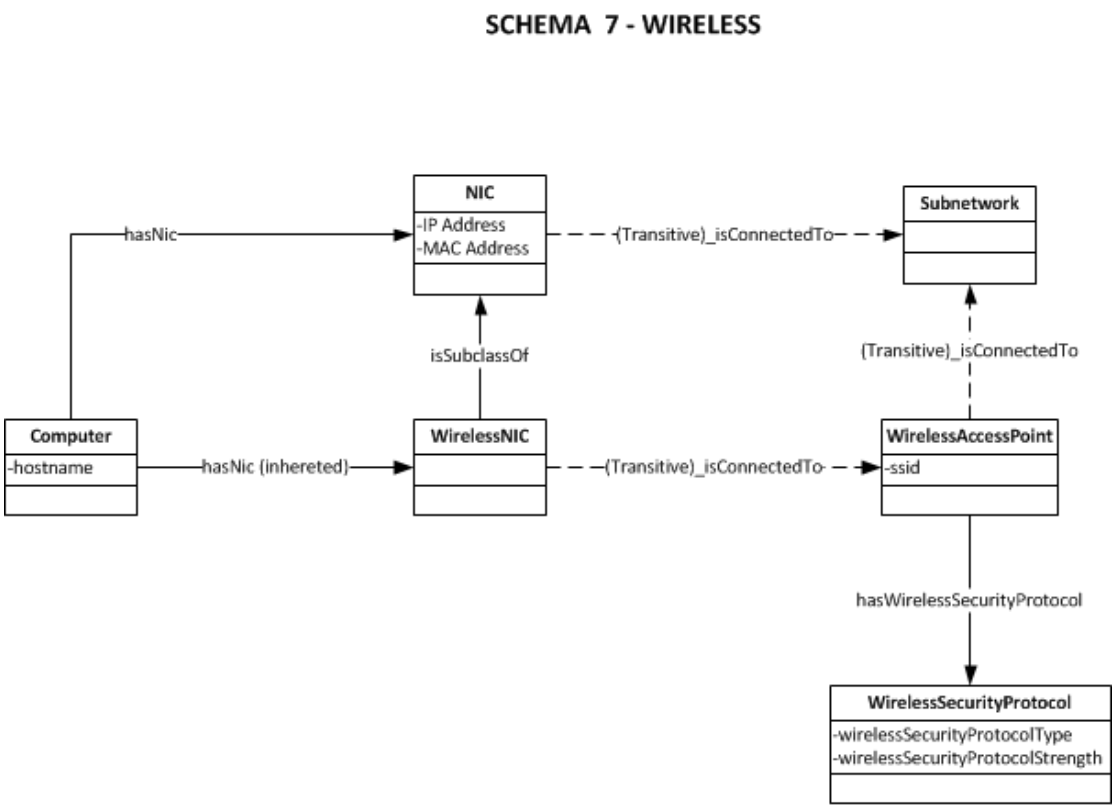

Figure 31 - SCHEMA 7: Wireless

Figure 31 depicts the schema implemented in the CST to represent effectively the Wireless Usecase described in Section 5.3.3.7. The three new classes introduced in this schema are *WirelessNIC, WirelessAccessPoint and WirelessSecurityProtocol*. WirelessNIC is a subclass of the NIC class and inherits its properties of *ipAddress and macAddress.* The *WirelessAccessPoint* is an intermediary entity that is situated between the *WirelessNic* that provides a *Computer* wireless capabilities and the *Subnetwork* of which it is a member. A *WirelessAccessPoint* is uniquely identified by its *Set Service IDentifier* (SSID). A *WirelessAccessPoint* has an associated *WirelessSecurityProtocol*. The *WirelessSecurityProtocol* is of a particular type (for example WEP, WPA2) and has an associated strength.

To facilitate the indistinguishable nature of wired and wireless connections from an attacker's perspective while maintaining the granularity offered by the inclusion of a wireless access point, a transitive relationship [3] was employed. This transitive property was implemented using the *OWL:TransitiveProperty* relationship. The transitive relationship is indicated with the prefix *"(Transitive)"* and a dashed line in the schema diagrams. The transitive relationship to enable wireless connectivity is the *isConnectedTo* relationship. This relationship asserts that:

[3] http://www.w3.org/TR/owl-ref/#TransitiveProperty-def

NIC isConnectedTO Subnetwork. And *WirelessNic* is a subclass of *NIC.*
WirelessNic isConnectedTo WirelessAccessPoint;
WirelessAccessPoint isConnectedTo Subnetwork ; Therefore
WirelessNic isConnectedTo Subnetwork Because
WirelessNic isConnectedTo WirelessAccesspoint isConnectedTo Subnetwork.

Theee potential problem of a situation where the *isConnectedTo* relationship has multiple domains is addressed by the addition of an *OWL:disjointUnionOf* property[4]. The disjoint union property enforces that a relationship with multiple domains can only be instantiated with one of them at a time.

The representation of wireless connectivity implemented in the CST maximises the granularity of representation and retains the fidelity that is sacrificed in other terrain models (for example, virtual terrain represents both wired and wireless connections as *"physical links"*[129]) to 'simplify' the representation. Implementation of the wireless connectivity in this manner permits the ontology to represent effectively detailed network configurations and attacks exclusively targeting wireless connections. These capabilities are not present in any other publicly available terrain model.

[4] http://www.w3.org/TR/owl2-syntax/#Disjoint_Union_of_Class_Expressions

5.3.4.8 – Schema 8: Virtualisation

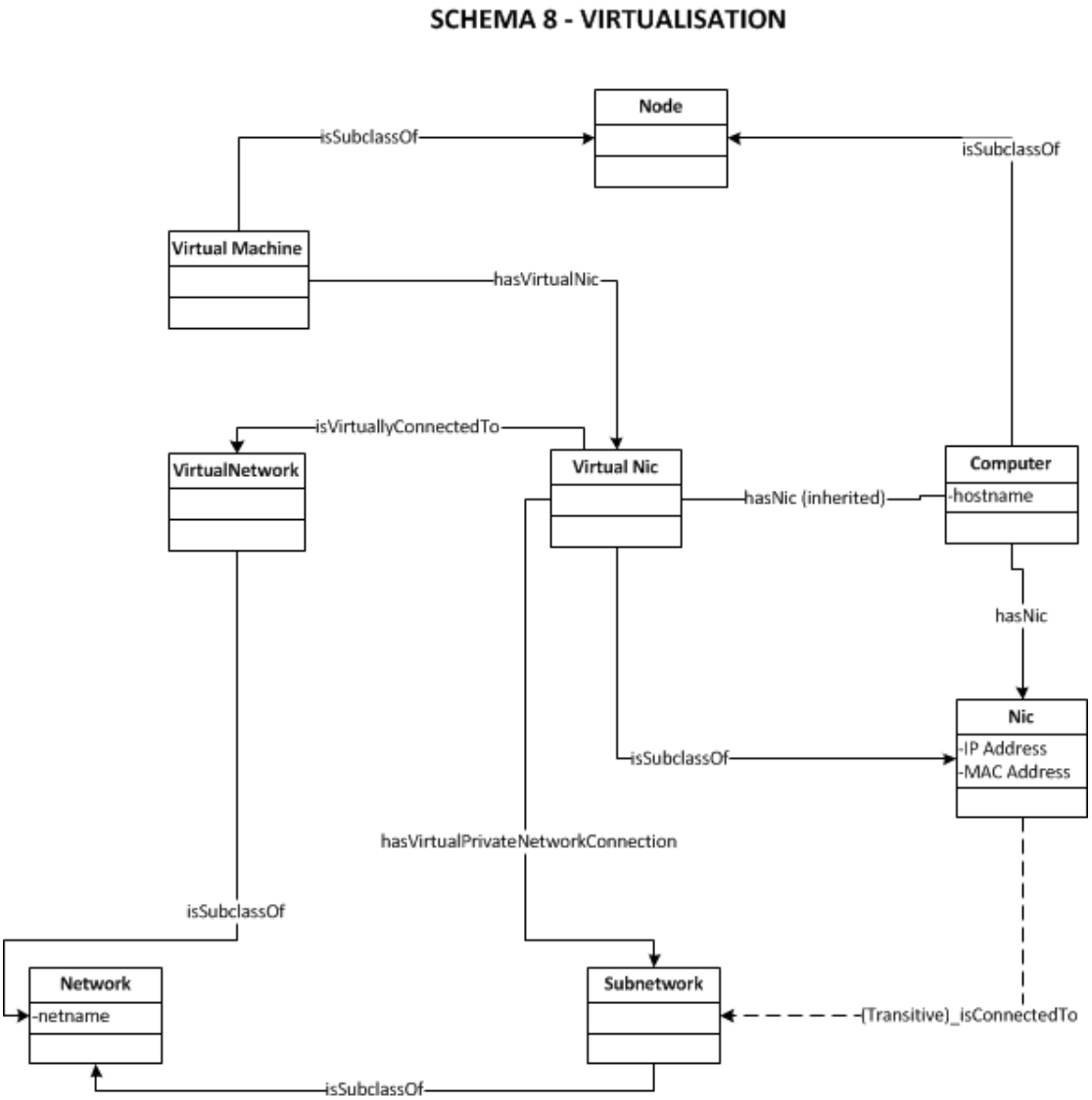

Figure 32 - SCHEMA 8: Virtualisation

The usecase from Section 5.3.3.8 establishes the requirements that *Virtual Machines, Virtual Networks and Virtual Private Network (VPN)* connections are represented in the CST. Figure 32 depicts the implemented schema. There are three new classes introduced in this schema: *VirtualMachines, VirtualNICs* and *VirtualNetworks.*

A *VirtualMachine* is a subtype of node that is only able to be associated with a *VirtualNic* to gain connectivity to a *VirtualNetwork*. To connect to a *Subnetwork,* the *VirtualMachine* must *virtuallyConnect* to a *VirtualNetwork* using a *VirtualNIC.* The physical computer that is hosting the *VirtualMachine* must also have a *VirtualNIC* associated with it so that it can connect to the network and *bridge* the connection between the virtual and physical networks. Represeting virtualised networks is one of the reasons that the representation of multiple NICs per computer as discussed in Section 5.3.4.1 is so useful. This bridging is how *VirtualMachines* are connected to networks in the real

world [5] .A *Computer* class that has a VirtualNic can use that *VirtualNic* to establish a *virtualPrivateNetworkConnection* to a geographically distinct subnetwork, assuming a path exists indirectly. This *VirtualPrivateNetworkConnection* is the equivalent to the *isConnectedTo* relationship between a *NIC* and a *Subnetwork,* it essentially just places the virtual network card on the subnetwork it is connecting to and adds the computer as a member of that Subnetwork.

The CST is the only publicly available Terrain Schema that is able to effectively represent *VirtualMachines, VirtualNetworks* and *VirtualPrivateNetworks.*

5.3.5 – CST Aggregated Schema

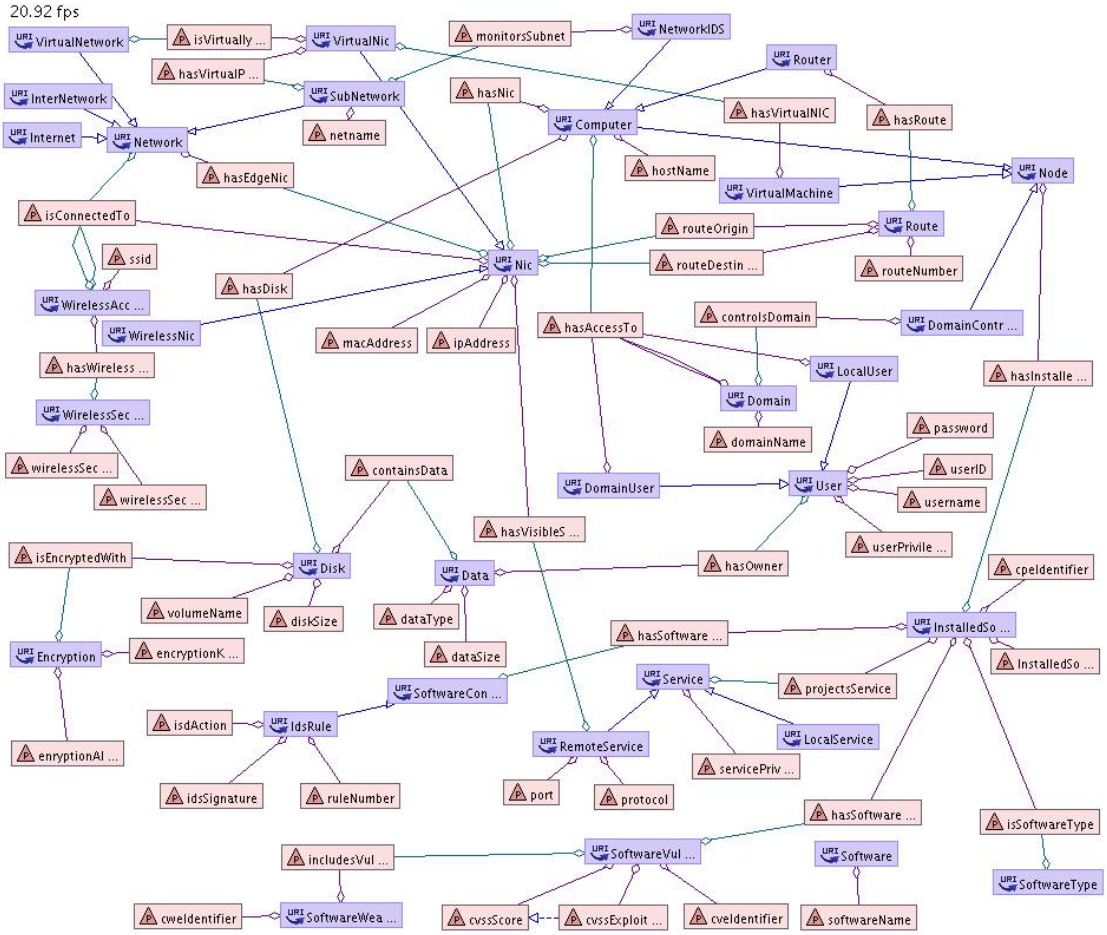

Figure 33 - Aggregated Cyber Simulation Terrain Schema

The Aggregated Schema for the CST is shown in Figure 33. It was generated by using the *easyRDF converter* web tool [171] to convert it from the TTL format to RDF/XML and then fed into *RDFgravity* [172] which can visualise the input from an RDF/XML file.

[5] http://blog.pluralsight.com/virtual-networking-101-understanding-vmware-networking

The aggregated schema is shown in Figure 33 clearly visualises the deep interconnectedness of the CST ontology. The interconnectedness and semantic strength of these connections drive the value of the ontology and make it relevant and useful for representing complex information terrains.

5.4 – The Threat Simulation Ontology

The Threat Simulation Ontology (TSO) is the module of the CESO that represents the threat elements seeking to disrupt the network infrastructure represented by the CST. It is designed to be maximally compatible with the STIX threat intelligence system [59, 173]. The reason for this similarity is to reduce the complexity of the conversion process should STIX reports be used to instantiate threat elements into the TSO in the future. To this end, where possible the semantics of STIX are preserved as are the class and property names, maximising compatibility

5.4.1 – Purpose

The purpose of the TSO is to represent the information about threats to the represented network. It is the module of the CESO that will contain information about attackers and exploits. It feeds directly into the CESO and to facilitate the generation of effects required of the CESO. The worldview taken by the TSO is one of omniscience of attacker. That is, it will be populated with all information required to represent a threat to the network and based on the stored knowledge and capabilities associated with this, be run against the network to determine where the strengths of the attackers and the weaknesses of the network overlap. Where this overlap occurs is where the CESO will be able to generate effects appropriately. Due to its differing intent from STIX all discussion of observables, indicators, reports and other elements that facilitate detection have been dropped. The TSO is not a detection tool, but a predictive one.

5.4.2 – Use Cases

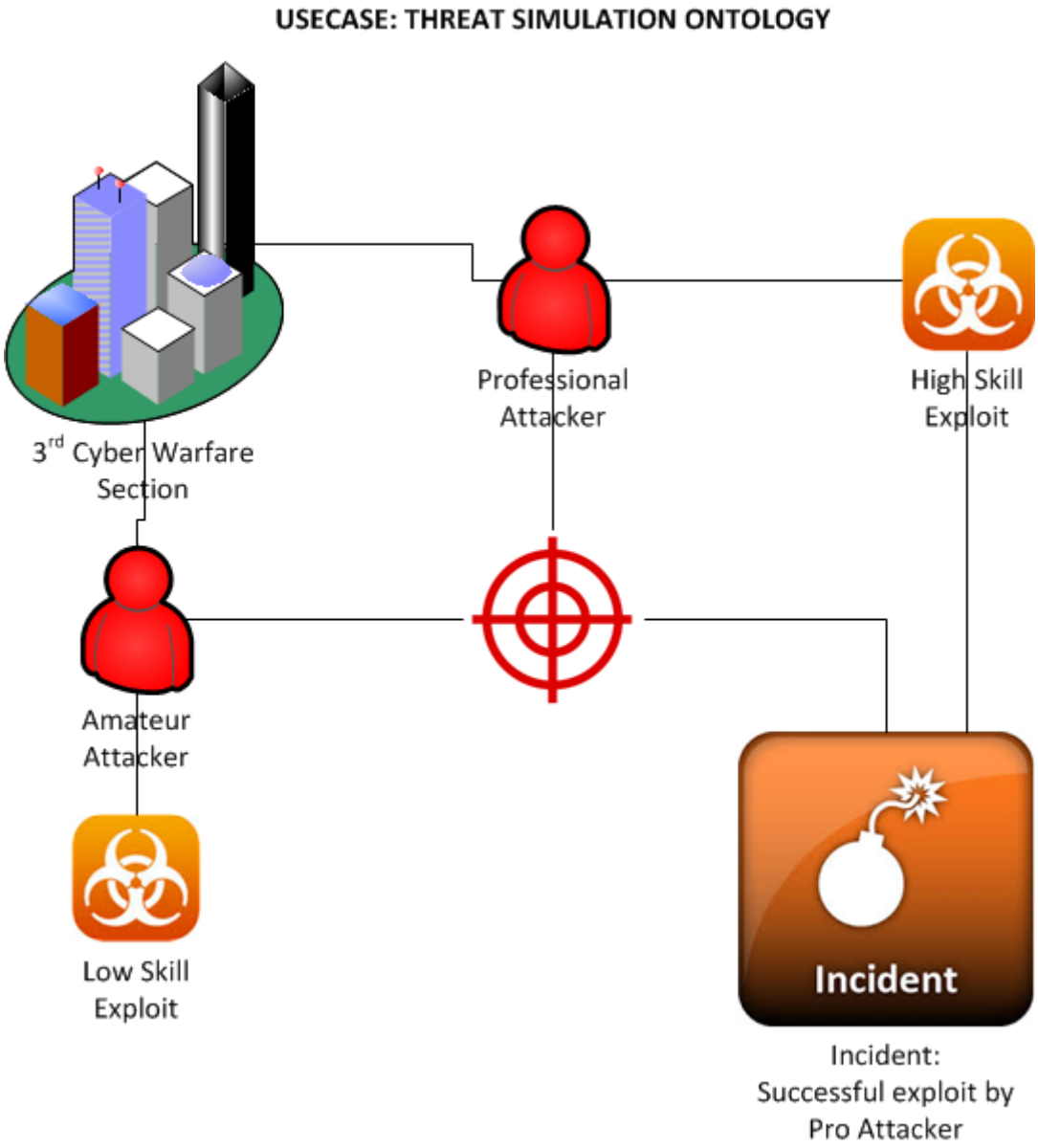

Figure 34 - TSO Usecase

The relative simplicity of the TSO (compared to the CST) has resulted in the development of a single usecase. The Threat Simulation Ontology usecase in Figure 34 is derived from the deductions in Section 5.3.4.3 regarding the purpose of the CESO and the implied capabilities associated with representing the effects of a cyber attack in a military simulation context. Achieving this end-state has led to the development of the following usecase. The usecase states that there is a campaign being conducted by the members of the 3rd Cyber Warfare Section. Two of their members are participating. One is a professionally skilled penetration tester and the other is an amateur, new to the field. Both attackers attempt to leverage exploits to compromise their target (denoted by the red crosshairs). The attacks will eventually cause an incident.

Based on this usecase the most prudent competency questions to address are:

27. *Which exploit maps to which vulnerability?*
28. *Whom are the threat actors involved in an incident?*
29. *Which organisation do the threats belong to?*
30. *What course of action is available to address an exploit?*

5.4.3 – Schema

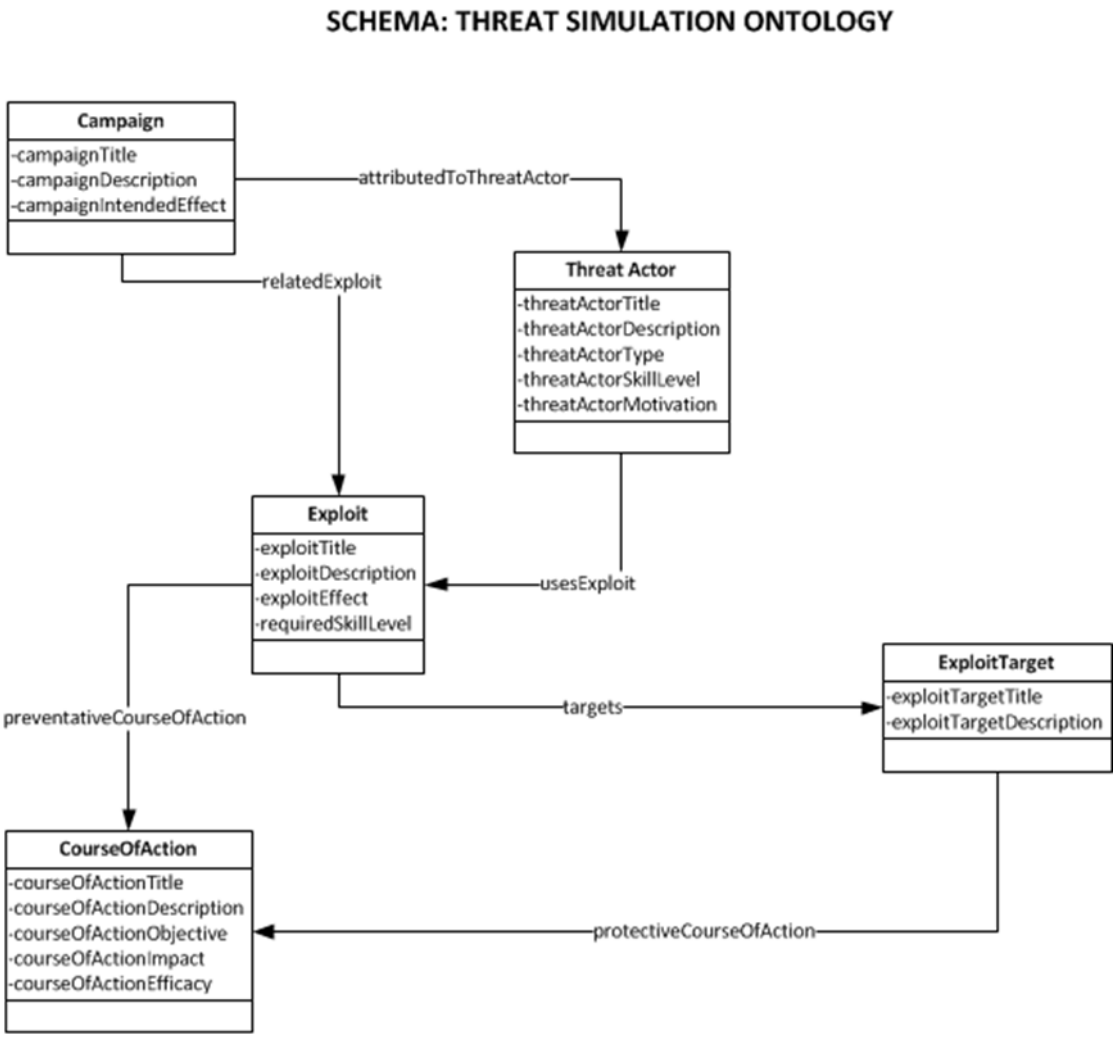

Figure 35 - Threat Simulation Ontology Schema

The schema for the TSO is comparatively much smaller and simpler than that of the Cyber Simulation Terrain. It draws on STIX for some of the core structures and semantics of the ontology. The interpretation and expansion of the ontology to shift it from a threat intelligence communication system to a threat ontology is the focus of this section. The above usecase is useful for contextualising the primary elements of the domain of knowledge that must be included in the TSO. *ThreatOrganisations, ThreatActors, Exploits, ExploitTargets and Incidents* are all important to represent in the TSO.

The schematic for the TSO is depicted in Figure 35. This visualisation shows a much smaller ontology than the CST. The concepts within the TSO all have many more properties than the CST. The increased number of properties and decreased number of concepts is consequential to the differing burden of work carried by each ontology. The only element of the TSO that directly contributes to the modelling of cyber-effects on military systems is the *Exploit* concept. The remainder of the TSO is incidental knowledge that helps to inform the decision maker and contextualise the attacks and effects that are being projected onto the CST. The above usecase is limited in its interaction with the actual conduct of an attack. It is a modular representation of the contextualising threat knowledge that is essential for informed decision making. The concept underlying this idea is that the CST is used to represent all aspects of an attack except for the final exploit. The TSO bridges across to the CST through the *ExploitTarget to Vulnerability Portkey* (Discussed in detail in Section 5.5.2) and executes the exploit. This bridge between ontologies is also available for the decision maker to query responsively, determining the identity, motives and affiliations of an attacker where viable to provide.

The *Exploit* Class is used to replace the *TTP* concept in STIX. TTP or *Tactics, Techniques and Procedures* refer broadly to the tools and processes an attacker uses to achieve their objectives. The TTP focus was too broad for the TSO and superceded by the more precise *Exploit. Exploit* defines some informational points about the exploit as properties and maintains links to *Courses of Action* that could prevent the execution of the exploit, and to the *Exploit Target* class. The *Exploit Target* class appears to be a redundant re-implementation of the *Vulnerability* concept in the CST. Though closely linked, they are two distinct concepts. The *Exploit Target* was retained in the TSO to minimise the number of inter-ontology relationships that exist in the CESO. Consideration was given to applying rhea *OWL:sameAs* rule[6] to this representation to force the equivalence of the bridging concepts between the CST and the TSO. This was not implemented due to the subtle differences in their nature. The *CST:Vulnerability* is used only to identify weakness in an *InstalledSoftwareVersion.* The *TSO:ExploitTarget* is closer to a representation of all of the metadata associated with the vulnerability. Imposing a *sameAs* rule would interfere with the representation and detract from the utility of the TSO.

The TSO uses a *Campaign* construct to group related *Exploits, Threat Actors and Threat Organisations together. ThreatOrganisations* are explained in detail in Section 5.5.1. The *threat actor* is a representation of an attacker's cyber-persona. The properties of a Threat Actor are all informational except for *SkillLevel.* A *ThreatActor's SkillLevel* is a numerical value from 1 to 10. 1 is most skilful, 10 is least skilful. The reason for the inverse scoring system is to maintain interoperability with the *CVSS Exploitability Score,* which ranks the exploitability of a given vulnerability from 10, being extremely easy to 1 – extremely hard to exploit. If an Attacker has a skill level of 1 then they can successfully employ almost any exploit

[6] http://www.w3.org/TR/owl-ref/#sameAs-def

Malware is not explicitly represented in the TSO. The reason for this is that most malware in this context is used to achieve an exploit and encapsulated in that representation. However, in the case of persistent malware (e.g. backdoors, key loggers etc.) these are implemented as an installed version of software on a system. Malware of this nature is software in its own right that must be installed (or at least present) on a system and will need to generate a service of some kind to be functional. For example, a backdoor may open a port that an attacker can connect to. Fundementally this is the same process that operates standard software. To link the 'malware' software to the TSO, it is associated with its own vulnerability linked to an exploit that reflects the malware and then to the effect that exploit can create. For example, Keylogger software is installed on a computer. The keylogger is essentially exploiting its own vulnerability (from a blue team perspective) to provide the attacker information, hence creating a confidentiality compromising 'Interception' effect.

The Course of Action is an informational relationship that links an exploit (and through the exploit target, a vulnerability) to a descriptor of how to prevent or remediate the exploit. The exploit target is the *Portkey* to the Vulnerability concept in the CST. The Exploit target is maintained in the TSO rather than linking the concepts directly to the CST to preserve the modularity of the TSO. By linking each directly, the number of inter-domain links increases and makes the ontology less independence and hence, less reusable.

The CST is linked to three other domains. The CST is contained within the CESO, but the Threat Organisation hosted in the Organisational Domain and the Incident in the Event Domain are not. These relationships will be covered in more detail in the discussion of the CESO schema.

5.5 - Cyber Effects Simulation Ontology Schema

The role of the CESO is conceptual in nature. It has two key functions: To represent cyber effects and to facilitate inter-ontology linkages. According to Obrst's classification of ontologies the CESO is a mid-level ontology [90]. Mid-level ontologies exist to facilitate the up-down interaction between low-level ontologies (such as the CST and TSO) and higher level ontologies that contain fundamental concepts like time. Mid-level ontologies also facilitate the linkages and communications between subordinate low-level ontologies. The design considerations associated with each intended function of the CESO are examined below.

5.5.1 – Representing Cyber Effects

Section 2.1.2 of the literature review discussed cyber effects and identified the *DIMFUI* framework for characterising cyber effects. Effects are not linked to a specific attack or specific infrastructure but are a broad, conceptual inference made based on the interaction of these two actions. The full characterization of an effect is reliant on a wide understanding of what caused the attack, and where it occurred in the context of associated metadata about the duration and severity of the effect. The CESO defines the cyber effects as concepts within the CESO. The *CyberEffect* class is instantiated into each of the seven effects of: *DegredationEffect, InterruptionEffect, ModificationEffect, FabricatonEffect,*

UnauthorisedUseEffect, and InterceptionEffect. These concepts themselves contain no properties or attributes. They are instances of the *CyberEffect* class belonging to the *CESO* namespace. The intent of this representation is to retain a cyber effect as a purely conceptual element of the CESO. A *CyberEffect* has two linkages to it. The first is from *Exploit* to *CyberEffect*. This relationship is intended to be an informational one – allowing decision makers to assess the likely effects that a cyber attack would have on their system as well as being able to show quickly every exploit that can achieve a particular effect during querying. The second linkage is from an *Incident* to a *CyberEffect*. This is how effects occurring as the result of an attack are represented. The effect itself has no metaproperties, all of the information for the effect is stored in the *Incident.* Within the incident, the two key meta properties are *Severity* and *Duration*. These indicate the magnitude of the effect and define a temporal element if required.

For example, these facilitate the accurate representation of the spectrum of degradation from slight slowing through a severely degraded effect to the point that it totally denies access to an asset. The duration is for effects with a temporal element – like interruption where access to an asset may only be unavailable for several minutes – representing a forced restart or a power loss. The centralisation of all active (and previous) effects into the event domain as incidents ensures a consistent representation of effects. It also permits effective logging of effects as they occur (and cease) and enables an analyst to track patterns and relations (for example, if an unauthorised use effect always follows an interception effect the induction may be that the enemy is targeting user credentials and using them to access the system.

5.5.2 – Portkey Relationships

Section 2.5.1 discussed the concept of o*ntology-bridging* as a method of linking disparate ontologies. In the context of the CESO these *bridged* relationships are referred to as *Portkeys*. The word *Portkey* originates from the *Harry* Potter fictional realm and refers to "*an object enchanted to instantly bring anyone touching it to a specific location"*[7]. It is used in this thesis as a to describe an *Ontology Bridge* (See Section 2.5.1) that connects one ontology to another through a common object. *Portkeys* are formalised inter-ontology relationship controlled by the CESO. The term has been adopted after exposure through colloquial use in the Australian Research community In the CESO the syntactic definition of a *Portkey* is:

ceso:portkey_<namespaceFrom>-<namespaceTo>_<ClassFrom>-<ClassTo>

For example, to define the link between an incident and the effect it causes the following *Portkey* is defined:

ceso:portkey_event-ceso_Incident-CyberEffect

[7] http://harrypotter.wikia.com/wiki/Portkey

The implementation of a rigid naming convention is to maximise the human readability of the CESO *Portkey* relationships which will rapidly become confusing as the multiple sub-ontologies interact and overlap, particularly into the future as the CESO grows and is expanded.

The CESO links to external ontologies. These are the Event and the Organisational Ontologies. These two Ontologies are being developed as part of ongoing work in simulating tactical land combat in a broader context that is out of the scope of this thesis. The contribution of this thesis to these two areas is the development of the *Incident* class of event to reflect a cyber attack as an event using the STIX lexicon and the addition of the *Threat Organisation* Class and subclasses to *Organisation* to flesh out the abstracted version of Threat Actor collective in STIX.

In the Event Domain, the incident class is associated with an *Exploit*, a *Node* and a *Course of Action*. The metadata continued within the incident class itself is the unique ID of the incident (for logging purposes), the duration of the incident (to facilitate effects like interruption with a temporal element) and a severity rating from 1 to 10 (where ten is most critical and 1 is least severe). The severity metric is used to support effects like Degradation where there is a spectrum of effects contained within it from slow at the non-severe end to total denial or destruction at the critical end. The associated course of action will be used to represent feedback from technicians about how to resolve an incident. The Event domain is used for more than the tracking of incidents. It handles the monitoring of all events that change the state of the network represented in the CST. This includes tracking when sessional connections are made between nodes on a network when an effect is present etc. An intended expansion of the *Event* domain is towards tracking the status of *MTNDM* described in work into DVT [81] The event domain does this by tracking the current states of IP addresses and ports. When an *MTNDM* is triggered the event domain will record the change in state of the relevant IP addresses and ports, achieving this much more efficiently than the method proposed in DVT [81] which required a reduction in fidelity as well as dynamically reforming the entire terrain through the use of terrain manager agents.

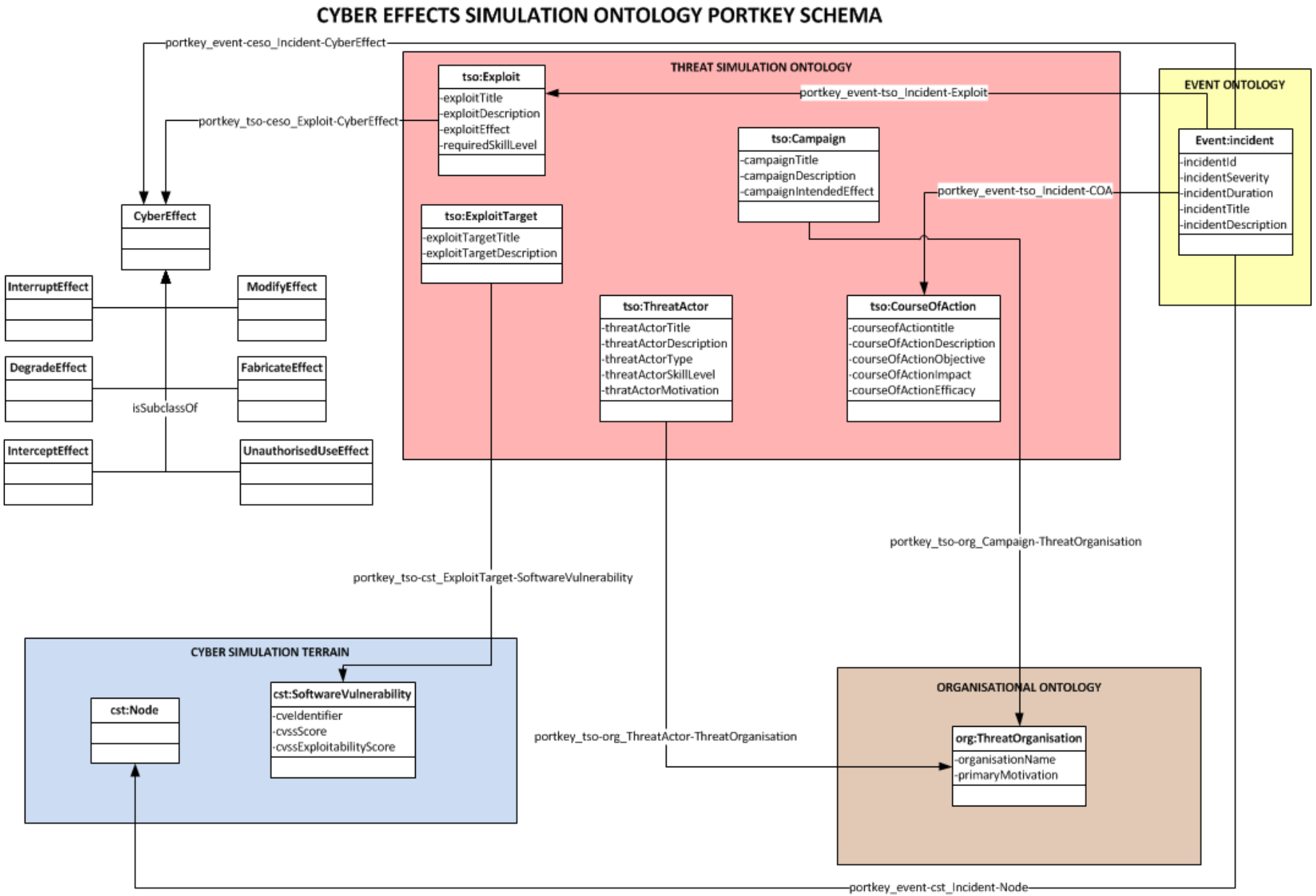

Figure 36 - Cyber Effects Simulation Ontology *Portkey* Schema

5.5.3 – Summary of the CESO Schema

The high-level architecture of the CESO is in Figure 36. It depicts the CESO and the sub-ontologies that it interacts with. The green shading denotes the boundaries of the CESO. Contained within the CESO are the two sub-ontologies – the ontological network model called *The Cyber Simulation Terrain* (Blue) and the threat-centric ontology called *The Threat Simulation Ontology* (Red). The CESO also contains its own concepts, denoted by the white circles. These represent the Super and Subclasses of the Cyber Effects that were determined to be suitable for inclusion in Section 2.1.4.3. The grey area denotes the collection of all linked ontologies that enable the functionality of the CESO without being an integral part of it. These are the Event Ontology (Yellow) and the Organisational Ontology (Brown). Contained within each of the sub-ontologies are black circles that represent the *Portkey* concepts in the sub-terrains. All of the *Portkey* relationships are stored centrally in the CESO to ensure a consistent representation of inter-domain communication and preserve the ontological commitment of the subordinate domains. Note that the *Portkeys* represented are not the only classes in the sub-ontologies, the others are not shown for the sake of space.

Based on the design decision to use the CESO as a mid-level ontology to facilitate communication between the subordinate ontologies and house the cyber effects there is only one key competency question for functionality that needs to be answered:

31. What effect has an exploit had on the system it is targeting?

5.6 - Evaluation

5.6.1 – Evaluation approach

The evaluation approach employed by the CESO is that outlined in the *Agilitology* process, discussed in Section 4.2.2. This three phase approach of integrated usecase assessment, usecase testing and finally review and appraisal by clients and domain experts will ensure that a robust ontology is generated and developed into the future.

5.6.2 – Evaluation Usecase

5.6.2.1 – Introduction to the Usecase

The evaluation usecase is an extension of the one initially proposed in Section 5.2 that accounts for the additional functionality that was determined as logical extensions of the initial problem as part of the development process, particularly during the development of the CST. For this reason, elements of virtualization, Wireless, Data, disks, encryption and defensive measures will be added in. The proposed final usecase is shown in Figure 37. Section 5.6.2.4 will describe the competency questions that the Ontology must be able to answer about the usecase and Section 5.4.2.5 will describe the results and provide discussion.

5.6.2.2 – Usecase Narrative

The usecase that is presented in the following section is built around the competency questions that have been extracted from the incremental usecases throughout this chapter. Combined, they present a compelling usecase to determine the effectiveness of the CESO for representing cyber attacks in a military simulation context. The core structure is from the problem usecase defined in Section 5.2 with the evolutionary capabilities of each usecase added until the usecase was completed. The usecase has been encoded and will be made available on GitHub for review[8] .

[8] https://github.com/AustralianCentreforCyberSecurity

5.6.2.3 – Usecase Visualisation.

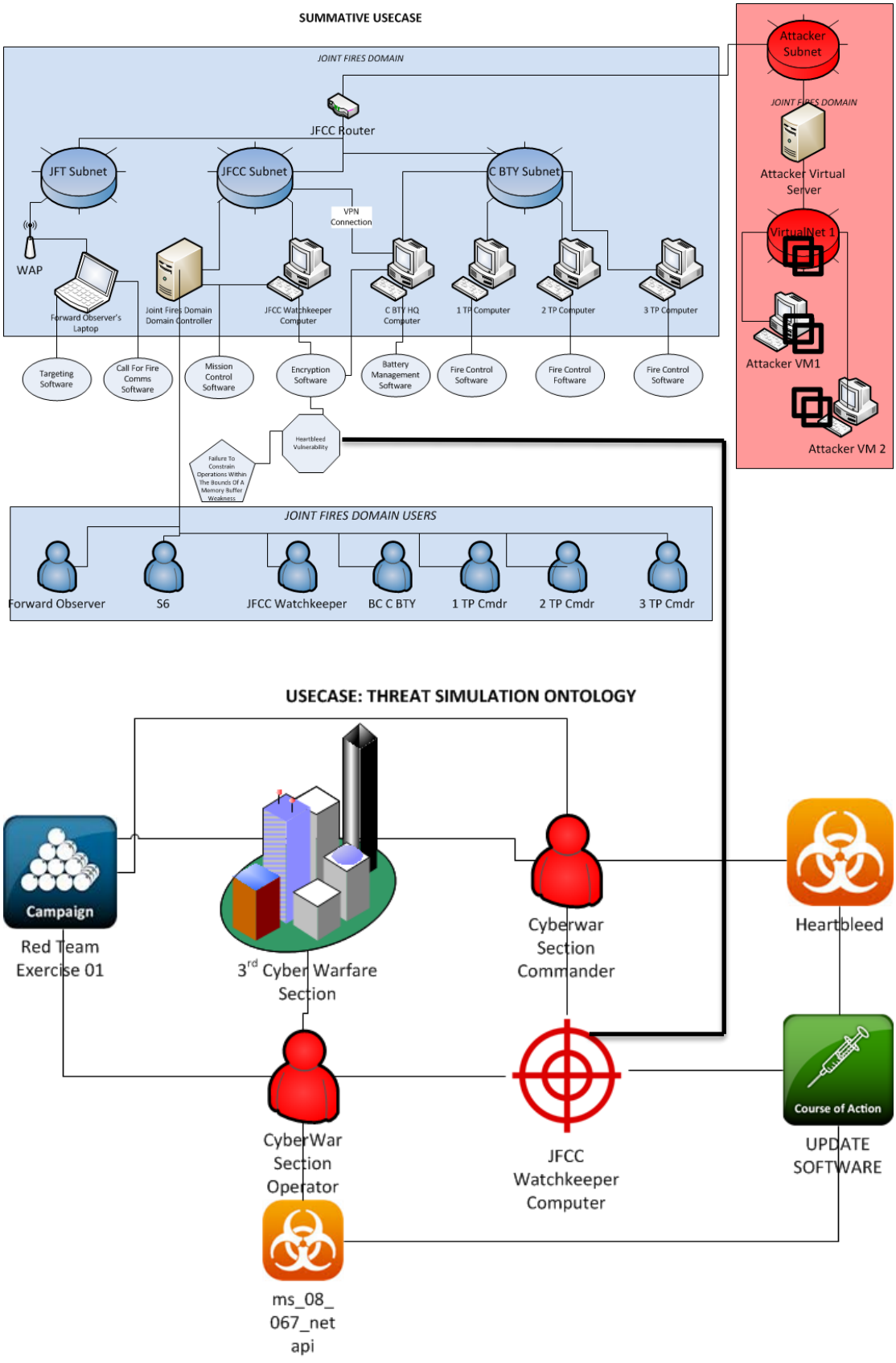

Figure 37 - Aggregate usecase for the evaluation of the Cyber Effects Simulation Ontology

5.6.2.4 Summary of Competency Questions to Be Answered.

Thirty-one competency questions have been identified through chapter five to determine the basic functionality of the Cyber Effects Simulation Ontology. The questions are as follows:

1. Which Nodes are visible in the ontology?
2. Which Networks are visible?
3. Which Subnetwork does each node belong to?
4. What is the IP address of each node on the network?
5. What software is running on which machine?
6. Which services are remote and what are they projecting?
7. Which nodes on the network are running vulnerable software?
8. What weakness encompasses this vulnerability?
9. How difficult is it for an attacker to exploit this vulnerability?
10. Which computers represented in the ontology are members of a domain?
11. Can a domain user access any node that is a member of the domain?
12. Who are all the users of a given domain?
13. Which users are administrators and that are normal users?
14. Which nodes are running HIDS?
15. Which subnetworks are running NIDS?
16. Can a HIDS detect a vulnerability or exploit present in the terrain?
17. Which computers have storage disks?
18. Which disks have Data?
19. Which disks are encrypted
20. Who owns piece of data X?
21. Which networks are wirelessly accessible?
22. Does a port scan return both wired and wireless connections together?
23. What is the wireless security protocol that is being used to protect a given network?
24. Which nodes on the network are virtual machines?
25. What are all the active VPN Connections?
26. Can a virtual machine be seen as part of a physical network?
27. Which exploit maps to which vulnerability?
28. Who are the threat actors involved in an incident?
29. Which organisation do the threats belong to?
30. What course of action is available to address an exploit?
31. What effect has an exploit had on the system it is targeting?

5.6.2.5 - Results of Competency Questions.

The competency questions listed in Section 5.6.2.4 describe the required functionality of the CESO. In the next section of the thesis, Competency questions are tested against the collective usecase by implementing the usecase in RDF and combining it into a graph database. Once in the graph database

the SPARQL query language is used to ask each question individually. SPARQL queries follow a simple format:

SELECT <object> FROM <database> WHERE
{
    ?Subject ?Predicate ?Object .
}

The query is entered in this format to the graph database and a result is returned. For example, the SPARQL Query for Competency Question 1: *Which Nodes are visible in the ontology?* Will read as:

*SELECT ?**Node** FROM **Evalusecase_DB** WHERE*
{
    *?**Node rdf:type cst:Node** .*
}

This query is searching the *Evalusecase_DB* database for any object of the type *Node.* It returns the following results:

Node
usecase:JointFiresRouter
usecase:JointFiresDomain
usecase:JointFiresDomainController
usecase:JFT_Computer
usecase:JFCC_Computer
usecase:C_BTY_HQ_Computer
usecase:C_BTY_1TP_Computer
usecase:C_BTY_2TP_Computer
usecase:C_BTY_3TP_Computer
usecase:JFT_Laptop
usecase:NIDS_JFCCNET
usecase:AttackerVirtualServer
usecase:Attacker1_VM
usecase:Attacker2_VM

More complicated queries can utilise variable chains to elicit more complex results, for example Competency Question 7 asks: *Which nodes on the network are running vulnerable software?* This is implemented in SPARQL as:

SELECT ***?Node ?VulnerableSoftware*** *FROM* ***Evalusecase_DB*** *WHERE*

{

?Node cst:hasInstalledSoftwareVersion ?VulnerableSoftware .

?VulnerableSoftware cst:hasVulnerability ?Vulnerability .

}

This produces the resulting information from the instantiated Ontology:

Node	VulnerableSoftware
usecase:JFCC_Computer	usecase:OpenSSL1_0_1
usecase:C_BTY_HQ_Computer	usecase:OpenSSL1_0_1
usecase:C_BTY_HQ_Computer	usecase:AdobeReader8_1_3

Appendix A lists each of the 31 competency questions and the results from running the queries. Of these 31 questions, the ontology provided positive results for all of them and permitted the continuing development of the ontology. The remaining 29 SPARQL queries, as with the rest of the Ontology is available on the CESO GitHub repository for recreation of these tests.

5.6.3 – Evaluation by Client / Domain Experts.

Evaluation by Clients and domain experts will be undertaken simultaneously and continuously. This will begin with the release of the CESO as open source for public review and criticism. The public availability of the CESO will enable domain experts in the areas of cyber security, knowledge engineering, ontological engineering, and situational awareness to access, review it, improve or use it as they see fit. The motivation to open-source the code produced in the development of this thesis under the GNU General Public License v3[9] is driven by the essays of the notable free software activist Richard Stallman [174], who has highlighted a number of benefits associated with the free and open sharing of software.

5.6.4 – Results

This chapter introduced the Cyber Effects Simulation Ontology as a mid-level ontological framework that facilitates the interaction of its component sub-terrains: the Cyber Simulation Terrain and the Threat Simulation Ontology to represent the effects of cyber attacks on a computer network. The ontology has been designed to be transparent, comprehensive and accurate. Transparency has been

[9] http://choosealicense.com/licenses/gpl-3.0/

achieved by developing a semantically strong ontology that is easily readable and interpretable by humans as well as machines. The Transparency is further enhanced by the public release of the RDF schema and example use-cases to promote academic review and enhancement into the future. The ontology is comprehensive in the context of the usecases it is required to represent. By incrementally developing the ontology using the *Agilitology* method the Cyber Effects Simulation Ontology can facilitate accurate representation of all related concepts to the component usecase. The requirements of each component usecase were developed by synthesizing the decomposition of the problem use case with existing work in representing the area and industry implementations of the represented technology. Each usecase produced some competency questions that were used for incremental testing of each component schema and finally as a comprehensive test to ensure that the ontology is able to meet client specifications. The formal review and endorsement by domain experts to validate the ontology will be achieved gradually, beginning with this thesis, followed with the public release of the schema for public review. A conference paper defining the CST has been submitted to for peer review towards establishing peer credibility of this work too.

This chapter has detailed the development of a comprehensive ontology intended for representing cyber effects in a clear, transparent, comprehensive and accurate manner, utilising the *Agilitology* approach to bound scope and maximise the relevance of included concepts. The Ontology has been able to satisfy all questions developed through the usecase development phase of this chapter to prove competence. It is publicly available for public examination, adoption and improvement to further demonstrate its robustness. The CESO is capable representing with granular detail several elements not seen previously in publicly available Cyber Effects ontologies. Representing Virtual Machines, Wireless Devices, VPN, Data, Disks and threat organisations are all examples of additional capability beyond current implementations. All of these elements have been achieved while producing a robust, semantically strong, compliant ontology that leverages ontology bridging using *Portkeys* to overcome the problems associated with representing two fundamentally differing perspectives (attacker and defender) in the same ontology structure. The CESO, CST, TSO and formal definition of *Portkeys* are all valuable, novel contributions made to the fields of Cyber Security and Ontological Engineering by this chapter.

Chapter 6 – Conclusions and Future Work

6.1 – Summary of Research

This thesis has sought to determine a solution to the research question:

"How can we effectively develop a method of representing cyber effects in a simulated military environment that promotes transparency, comprehensiveness and understanding?"

This is a large, complex problem that was decomposed into five sub-questions and answered through the research and development undertaken for this thesis. A summary of the approaches to these questions and the results that were produced are detailed in the following subsections.

Subquestion 1:

What is the current state of knowledge representation for the field of cyber security? Is it comprehensive and does it promote transparency of representation? Does it allow the effective modelling of cyber effects in a military context?

Before developing a method of representing cyber effects in a simulated environment that promotes transparency, comprehensiveness and understanding it has been necessary to gain an understanding of the current state of the field. The reason that this is required is to prevent the reinvention of existing work or contributing to the crowding of an over-developed field. Section 2.4 foregrounds some important principles about knowledge representation that must be considered in the context of cyber security knowledge representation. The most significant conclusion is that no knowledge representation is perfect. Errors, bias and encoding limitations are unavoidable. To combat this, we are required to commit to a specific worldview to minimise the ontological commitment of a knowledge representation, reduce the complexity and reduce the chance or impact of error. The section concludes that an ontological knowledge structure is most appropriate because of their extensibility, modularity, semantic strength and machine readability. Section 2.5.3 highlights how ontologies are suited to representing the complex interrelationships of the systems, software, hardware, vulnerabilities, exploits and actors in the domain. The survey of existing ontologies in Section 2.6 concludes that many of the existing ontologies try and encompass too much and compromise their ontological commitment, weakening them too much to be useful. Many of the other existing general-purpose ontologies are focused on audit support and are glorified dictionaries of standards, controls and audit rules that aren't capable of modelling cyber effects. It highlights that the wiser approach is to separate ontological models into discrete structures with limited worldviews, bridging between them with what are described here as 'Portkey'. Section 2.6.1 explores the existing terrain models designed to support cyber effect simulations. It concluded that all existing models lack openness, are not granular enough in their representation and are not inclusive enough of what needs to be included in the domain. Section 2.6.2 examines current threat simulation ontologies. The conclusions from this examination are that existing threat ontologies are either old and opaque or focused too heavily on threat intelligence and aren't directly applicable to the problem of representing cyber effects in a simulated environment.

None of the existing ontological models examined in Section 2.6 are capable of effectively supporting the bridging between discrete sub-ontologies that the conclusions of Section 2.4 specifies as necessary.

Based on the analysis of the current state of knowledge representation for the field of cyber security we can conclude that there is a requirement for the knowledge structure to be an ontological structure. A survey of existing cyber security ontologies has concluded that most general-use cyber security ontologies are unsuitable for the purpose of representing cyber effects in a military context. The most effective approach is to bridge together disparate ontologies that achieve related but independent functions of modelling the network infrastructure and modelling the threats through the use of a central effects ontology. The existing 'terrain' ontologies lack transparency, electing not to publish their schema for academic scrutiny. They also lack granularity in representation and completeness of included concepts in the domain. The Threat ontologies surveyed were either lacking transparency and tied to large, commercial products or focused very heavily on Threat Intelligence and hence have a worldview unsuitable for inclusion. There were no ontologies surveyed that would be suitable for bridging between the others and facilitating the representation of effects in a simulation environment.

Based on the above analysis, the deduced conclusion is that the current state of knowledge representation in the field of cyber security is inadequate for the representation of cyber effects in a simulation context. The existing representations lack transparency, comprehensiveness, interoperability, extensiveness and modularity. Due to the lack of a suitable existing ontology (and in the context of the need to build resilience to ensure the future capabilities of military forces discussed in Section 2.3) there is a clear need for a comprehensive, transparent, extensible, modular and granular ontology to be developed that is capable of representing cyber effects in the context of simulation

Subquestion 2:

What are the areas, fields and disciplines that a comprehensive, effects-focused knowledge structure should encompass or represent?

The thorough analysis of the literature conducted in Chapter Two and produced a broad understanding of the requirements of a comprehensive, effects-based knowledge structure to represent effectively cyber effects in a military simulation context. These requirements were driven by the detailed analysis of the current military need for such a knowledge representation structure and the context that they would be operating in. Specific requirements were deduced through the application of the *Agilitology* to the creation of incremental usecases from a core, high-level problem usecase. The high-level usecase defined in Section 5.2 contained a number of problems that required addressing. Foremost of these challenges was the requirement to create a generic network structure that is able to represent the required use case and then be re-applied. To create realistic representation requirements from the problems in the high-level usecase the existing work into cyber ontologies, terrain models and threat intelligence frameworks needed to be considered in addition to wider domain knowledge. Once the requirements to effectively build the underlying representation of the usecase were enumerated they were implemented schematically.

Resulting from the use-case approach, the fields, areas and disciplines that an effective effects focussed cyber ontology needed to encompass were defined as the usecases. The required representations were determined to be:

1. Nodes and Networks
2. Software and Services
3. Vulnerabilities and Weaknesses
4. Domains and Users
5. Firewalls, Antivirus and Intrusion Detection Systems
6. Data, Disks and Encryption
7. Wireless Connectivity
8. Virtualisation
9. Cyber Attacks and attribution
10. Cyber Effects

These ten key areas of representation were implemented collectively into the aggregated usecase described in Section 5.6.2. The ability to effectively represent these ten areas was the competency benchmark for the CESO. The results of the testing carried out in Section 5.6 clearly demonstrate that the CESO is competent and is able to represent effectively the requisite areas, fields and disciplines to effectively represent the effects of cyber attacks in a military simulation context.

Subquestion 3:

What is a suitable agile, usecase-centric development methodology that can be used to develop a suitable knowledge structure?

The summary of existing research and development methodologies in Section 3.4 concludes that there is no suitable existing methodology. Chapter 4 defines the Agilitology approach, the first usecase centric ontology development approach. The Agilitology approach is built on the philosophical worldview of the design science methodology and applies the principles of agile software development approaches like Test Driven Development to the Methontology ontology development process. The proposed Agilitology approach will make ontology development accessible to non-experts who have a problem that a knowledge structure is suitable to solve. In addition to being a purely developmental tool it is useful for helping decompose the problem and address it in manageable segments, when the habit for ontology design seems to be approaching it as a single whole-of-domain problem. This also facilitates an improvement to the collaboration possible in ontology development. The modularity of the approach means that discrete ontologies segments can be allocated to developers and the probability of them working coherently when combined is increased, due to an awareness of the overarching usecase that their particular representation problem is drawn from. The *Agilitology* approach produces a naturally modular, extensible ontology due to its design process. The ontology is assessed continually throughout – each discrete usecase is gated. Until the developer is satisfied that the ontology segment can satisfy their usecase it isn't added to the larger usecase. In addition to the

informal testing, a formal testing and development phase at the end requires the ontology to satisfy a complex usecase before undergoing evaluation by domain experts and clients to determine its suitability and effectiveness for solving the problem it is developed for.
The Agilitology approach is used in this thesis to develop the Cyber Effects Simulation Ontology and its subordinate Ontologies. The implementation of this approach to successfully solve a real problem reflects on the validity of the *Agilitology* ontology development methodology to develop an effective knowledge structure to represent cyber effects in a military simulation context. This approach can also be expanded to develop ontologies for other problems outside the realm of cyber security.

Subquestion 4:
What is a suitable knowledge structure to support the representation of cyber effects in a simulation context?
The literature review in Chapter 2, particularly Section 2.4 and Section 2.5 identify that an ontology is the best choice of knowledge structure for the effective representation of cyber effects in a military simulation context. The review of existing Ontological solutions in Section 2.6 discovered that none of them were suitable for this task due to scoping or purpose incompatibility. Section 2.6.1 and Section 2.6.2 looked toward using modular ontological solutions collaboratively to solve this problem, examining ontological network models and threat representation ontologies respectively. The findings were twofold; first – the existing terrain models and threat ontologies were unsuitable for the task of representing cyber effects in a military simulation context and second, that even if they were capable of this there is no existing framework to unify the two.

Out of this gap in knowledge came the need for the Cyber Effects Simulation Ontology (CESO). The CESO is the unifying framework between its two component ontologies, disparate ontologies with minor supporting functionality and the cyber effects themselves. The component ontologies developed for the CESO are the Cyber Simulation Terrain (CST) and the Threat Simulation Ontology (TSO). The CST is based on the Virtual Terrain family of ontological network models though due to the lack of open source schema and academic analysis there were not used in more than an 'inspirational' manner, extended heavily in the CST. The TSO draws from the development into threat intelligence and adopts the core functionality and Semantics of the Structured Treat Information eXpression (STIX) in order to maximise interoperability and set the conditions for importing STIX reports to allow for the rapid assessment of new threats against the networks as they arise.

The CESO provides a robust, transparent knowledge structure that is suitable for the representation of cyber effects in a military context. The transparent nature of the ontology will allow analysts who are attempting to build resilience through enhancing their situational awareness and projecting plausible, probable and preferable futures to validate the results of simulations that use this knowledge structure by checking the results against the underlying knowledge that defined them. The modular design of the CESO positions it for significant revision and expansion into the future so that it can move and evolve as the cyber warfare paradigm does. The decision to release the CESO open source will permit

academic evaluation of the ontology as well as permitting the future development and use of the ontology by those working on this field into the future.

Subquestion 5:

How are the results of the knowledge representation validated; what is a suitable collection of relevant usecases to facilitate this evaluation?

The results of the knowledge representation are evaluated in three phases by the Agilitology process defined in Chapter 4. The first phase is the integrated usecase testing that occurs during development. The gated nature of the Agilitology approach requires the designer to test each ontology segment against its discrete component usecase and be satisfied with the results before incorporating it onto the larger ontology. The second phase of the evaluation requires that the ontology is evaluated against a collective usecase with client-approved competency questions to test its full functionality. When this is complete, the ontology must be accepted by the clients as solving their problem and deemed as suitable by domain experts on formal review.

The CESO was developed with incremental test usecases. Section 5.6 defines a suitable problem-centric usecase that tests each of the major components of the ontology. The querying of the instantiated usecase was able to return positive results for each competency question asked of the usecase. Now, the CESO is to be formally evaluated by domain experts in two waves. First, the assessment of this thesis and second is the public release of the schema for review, public comment and academic evaluation. This will be extended through the availability of the CESO for future use and improvement. Adoption for future use will indicate the degree of acceptance that the CESO receives from the community.

6.2 – Future Work

There is substantial potential for future work emerging out of this thesis. The summary of future work will be split into three areas, future work for the *Agilitology*, Future work for the CESO and potential applications of the CESO.

6.2.1 – Future work for Agilitology

The *Agilitology* is a step towards a more agile future for ontology development. One of the underlying problems for Ontology development is the requirement to bring experts from multiple domains together and formalise their knowledge into an ontological structure. The steepest learning point of ontology development is learning how to encode the ontology. The decomposition of a domain of knowledge into Entities, Relationships and Attributes is a relatively trivial challenge compared to that. Therefore, an area of future development for the Agilitology is to remove the most complex element of the ontology creation process – the encoding into a formal knowledge description language. There is currently work being conducted into the development of technology that is able to intuitively develop ontologies through the analysis of Controlled Natural Language Competency Questions [160, 161].

Application of this technology to the Agilitology approach would mean that the process is truly accessible to novice ontology and that the client could conduct all aspects of development, without need for intervention of an ontology developer.

6.2.2 – Future Work for CESO

There are some areas of future work for the CESO. First of these is the automation of the instantiation process. This is envisioned as occurring in two phases. The first is the conversion of the existing CVE, CWE, CPE, NVD, OSVDB and Exploit-DB into CESO-formatted Vulnerabilities, Weaknesses, Software, and Exploits into pre-existing graph triplestore databases to facilitate rapid association and querying. This is also applicable for the automatic conversion of STIX formatted threat information to be imported into the CESO for use in the representation activities.

Additionally, the automated collection of network information has been touched on by several research efforts [69, 75, 76, 129] as being of key importance to making the tools usable. The vision for the CESO is that it is able to develop and deploy networks of collector daemons onto networks that can automatically report back useful information as it becomes available. These daemons could range from being as simple as monitoring the ARP requests on a network to the complex networks of collector Daemons described in work by DSTO [136] [175]. The automation is equally applicable to the TSO. In order to minimise the time it takes to determine if an emerging threat poses a danger to the network being protected an automated tool could be developed to leverage the semantic interoperability of the TSO with STIX and instantly provide new threat information to the CESO as the threat intelligence arrives. Automatic threat assessment of the network would be a valuable tool to security officers and network administrators who may know of the existence of a new threat but lack and understanding of the potential impact that it will have on their network. As soon as the information is available they are provided with a report of its ability to interfere with the network and likely effects that this would have.

The CESO can also be furthered by adding additional metrics enhance critical path analysis or mission planning algorithms in order to triage and prioritise systems when attempting to defend a network in a resource constrained environment. Developing a GUI to assist in the creation and management of the CESO and interpretation of the generated data in a similar vein done into work by Visual Analytics for Cyber Red Teaming [176] would greatly assist analysts with designing, managing and interpreting data produced by the CESO.

6.2.3 – Potential applications of the CESO

There are numerous potential applications of the CESO. The first, intended one is as the underpinning cyber-knowledge structure to support simulation of cyber effects on tactical land combat.[164]. It can, however be used in a number of other settings including network vulnerability assessment, network change management, intrusion detection, impact evaluation and decision support.

The conduct of vulnerability assessment by the CESO relies on a full and accurate representation of the target network being represented by the CST and an expansive level of information about existing threats and exploits contained in the TSO. Based on this, the TSO is run against the CST to attempt to

enumerate all possible vulnerabilities, exploit and COAs to remediate these exploits on the network. This will also permit the situation were a new zero-day vulnerability is discovered. With a fully updated network knowledge store, a query for all instances of the vulnerable software can be run and returned rapidly, allowing decision makers to rapidly identify and prioritise patching of the system or the application of mitigations to reduce the risk of exploit.

Network Change management in this context means the ability to detect rogue or masquerading devices on a network by establishing a known 'good state' which is stored in the ontological model and then using the dispersed net of sensors discussed in Section 6.1.2 to compare the current state of the network to this known good state. The ideas from the original virtual terrain can be extended through the CESO as well to enhance intrusion detection capabilities. By mapping all known IDS alerts to the TSO and then through that to the CST and applying the threat intelligence kill-chain logic of STIX an effective IDS is possible.

Detailed impact assessment is possible by integrating the CESO with a mission mapping approach to create a paired mission impact assessment system such as the cyber-Argus framework [68] or the mission impact work done by MITRE [29-31]. This links to the potential to leverage an ontological towards conducting cyber Battle Damage Assessment [23, 31, 177-179] to determine the effect and impact of a cyber attack.

Finally, the CESO could be implemented as a decision support tool for red teams (and automated red teams). Assuming that their toolsets used to conduct the penetration test are able to backload all data to a central data store, the CESO would be capable of facilitating 'what-next' scenarios. This could be achieved based on the currently known state of the network and the detailed mappings of software, vulnerabilities and exploits that the network has – pushing suggested courses of action to the red teams.

6.2.4 – Summary of Future Work

There is a rich body of future work emerging from the research conducted as part of this thesis. The release of the CESO into the public domain under the GNU v3 General Public License aims to assist in the facilitation of this research and the development of the fields of knowledge representation, ontological engineering, cyber situational awareness and cyber security.

6.3 – Conclusion

This thesis set out to answer the question *"How can we effectively develop a method of representing cyber effects in a simulated military environment that promotes transparency, comprehensiveness and understanding?"* This question emerges from a context of cyber capable military forces preparing to move into the future operating environment and wage war in a highly networked digitised environment, very vulnerable to disruption by cyber attack. Planning for a future where an attack has occurred and a force will be required to *fight through* in an information-degraded environment drives

the need to develop resilience. Resilience can be enhanced by planning for plausible and probable futures. The importance of this planning for continuing military effectiveness drives the imperative behind this research question "*How can we effectively develop a method of representing cyber effects in a simulated environment that promotes transparency, comprehensiveness and understanding?"* There is a paradox between the necessity to predict futures to prepare for the future and the inherent inaccuracies that the abstraction process to create the models that make results untrustworthy. Trust in the predicted futures can be gained by decision makers by establishing a transparent knowledge structure that will permit decision makers ensure that the future that they are preparing for is indeed based off the present state of the systems. An ontological structure was determined to be the most suitable for this knowledge representation. The development of this ontological structure was not suited to any existing development process, so the *Agilitology* approach was developed to facilitate usecase based agile development of ontologies. This approach produced the Cyber Effects Simulation Ontology, a modular, extensible, transparent knowledge structure that is highly suitable for use in representing cyber effects in a simulated military context. It also produced two component sub-ontologies: the Cyber Simulation Terrain for representing computer network infrastructure and the Threat Simulation Ontology for the Representation of cyber attacks and threat actors. The CESO as the overarching framework for these two sub-ontologies can determine the cyber effects active on a system. Agilitology usecase testing and is now being released for public critique and review, the open source release is intended to facilitate this and also share the product of this thesis with the exceptionally fertile grounds for future work established in this thesis.

To answer the question *"How can we effectively develop a method of representing cyber effects in a simulated environment that promotes transparency, comprehensiveness and understanding?"* we now know that by developing an ontology that uses an independent cyber-terrain model and a threat simulation model under a collective linking and bridging of effects concept we are able to effectively represent the effects of a cyber attack in a military context in a way that promotes transparency, comprehensiveness and understanding. Being able to accurately and transparently represent these effects will permit the use of simulation and modelling tools to enumerate futures based on this representation. The enumeration of futures will provide input into situational awareness, resilience and developing preparedness. Preparedness and resilience are the key attributes of a military force that will continue to operate effectively in the future in an increasingly contested cyber domain.

Bibliography

[1] O. Corcho, M. Fernandez-Lopez, and A. Gomez-Perez, "Ontological engineering: What are ontologies and how can we build them?," in *Semantic Web Services*, ed Nueva York: Premier Reference Source, 2007, pp. 44-70.

[2] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of management information systems,* vol. 24, pp. 45-77, 2007.

[3] D. S. Janzen and H. Saiedian, "Does test-driven development really improve software design quality?," *Software, IEEE,* vol. 25, pp. 77-84, 2008.

[4] M. m. Herman. (2015, 29 July). *Flaskr: Intro to Flask, Test Driven Development (TDD), and jQuery* [Git Repository]. Available: https://github.com/mjhea0/flaskr-tdd

[5] M. Scott, "Operating in a Degraded Information Environment," *Australian Defence Force Journal,* pp. 112-119, 2013.

[6] S. Boey, P. Dortmans, and J. Nicholson, "Forward 2035: DSTO Foresight Study," Defence Science and Technology Organisation, Canberra, ACT2014.

[7] MCAFEE, "Net Losses: Estimating the Global Cost of Cybercrime," IntelSecurity, Santa Clara, California2014.

[8] Kaspersky Labs. (2015, 21 October). *The Targeted CyberAttacks Logbook*. Available: https://apt.securelist.com/

[9] M. I. Centre, "Apt1: Exposing one of china's cyber espionage units," Mandiant2013.

[10] A. News, "China blamed after blueprints stolen in major cyber attack on Canberra Headqarters," in *ABC News*, ed: ABC News, 2013.

[11] D. Kushner, "The real story of stuxnet," *IEEE Spectrum,* vol. 50, pp. 48-53, 2013.

[12] Kaspersky Lab Global Research and Analysis Team, "Energetic Bear - Crouching Yeti," Kaspersky, Moscow, Russia2014.

[13] Kaspersky Lab Global Research and Analysis Team, "The Desert Falcons Targeted Attacks," Kaspersky Labs, Moscow, Russia2015.

[14] K. Finklea, M. Christensen, E. Fischer, S. Lawrence, and C. Theohary, "Cyber Intrusion into U.S. Office of Personnell Management: In Brief," Congressional Research Service, Wachington DC R44111, 17 July 2015.

[15] Federal Bureau of Investigation National Press Office. (2014, 21 October). *Update on Sony Investigation*. Available: https://www.fbi.gov/news/pressrel/press-releases/update-on-sony-investigation

[16] D. Regalado, N. Villeneuve, and J. S. Railton, "Behind The Syrian Conflict's Digital Front Lines," Fire Eye, Milpitas, California, Threat Intelligence ReportFebruary 2015 2015.

[17] E. Nakashima, "A list of US weapons designs and technologies compromised by hackers," in *The Washington Post*, ed: Weymouth, Katherine, 2013.

[18] Directorate of Future Land Warfare, "The Future land Warfare Report," D. o. Defence, Ed., ed. Canberra, ACT: Australian Army, 2014, p. 26.

[19] J. H. Irvine and S. Schwarzbach, "New Technologies and the World Ahead: The Top 20 Plus 5," *The Futurist,* vol. 45, pp. 119-234, 2011.

[20] S. Hajkowicz, *Global megatrends: Seven Patterns of Change Shaping Our Future*, 1 ed. Clayton South, VIC: CSIRO Publishing, 2015.

[21] Modernisaton and Strategic Planning Branch - Australian Army Headquarters, "The Fundamentals of Land Power," Department of Defence, Ed., ed. Canberra, ACT: Australian Army, 2014, p. 60.

[22] Directorate of Future Land Warfare, "Adaptive Campaigning: Future Land Operating Concept," Department of Defence, Ed., ed. Canberra, ACT: Australian Army, 2012, p. 100.

[23] United States Strategic Command, "The Cyber Warfare Lexicon: A Language to support the development, testing, planning and employment of cyber weapons and other modern warfare capabilities.," Department of Defence, Ed., 1.7.6 ed. Washington DC: Department of Defence, 2009, p. 45.

[24] Joint Staff Director of Operations, "Joint Publicaton 3-12: Cyberspace Operations," Joint Chiefs of Staff, Ed., ed. Suffolk, VA, 2013.

[25] D. Kostadinov, "The Cyber Exploitation Life Cycle," in *INFOSEC Institute: General Security* vol. 2015, ed: INFOSEC Institute, 2013.

[26] Deloitte, "Cyber Espionage: The Harsh Reality of Advanced Security Threats," Deliotte Develpment LLC2011 2011.
[27] B. Binde, R. McRee, and T. J. O'Connor, "Assessing outbound traffic to uncover advanced persistent threat," SANS Institute22 May 2011 2011.
[28] MITRE Corportation, "An Active Defense Strategy for Cyber," vol. 12-3352, MITRE, Ed., ed: MITRE Corporation,, 2012, pp. 1-2.
[29] S. Musman, M. Tanner, A. Temin, E. Elsaesser, and L. Loren, "Computing the impact of cyber attacks on complex missions," in *Systems Conference (SysCon), 2011 IEEE International*, 2011, pp. 46-51.
[30] S. Musman and A. Temin, "A Cyber Mission Impact assessment tool," in *Technologies for Homeland Security (HST), 2015 IEEE International Symposium on*, 2015, pp. 1-7.
[31] S. Musman, A. Temin, M. Tanner, D. Fox, and B. Pridemore, "Evaluating the impact of cyber attacks on missions," in *Proceedings of the 5th International Conference on Information Warfare and Security*, 2010, pp. 446-456.
[32] W. Stallings and L. Brown, *Computer Security - Principles and Practice*, Second Edition ed. Upper Saddle River, NJ: Prentice Hall, 2012.
[33] S. Marshall, "Tactical Cohesion," in *Men Against Fire: the problem of battle command in future war*, ed New York: William Morrow, 1947, pp. 123-137.
[34] Chief Information Officer of the United States Army, "U.S. Army - Network Security Enterprise Reference Architecture," Department of the Army, Ed., ed: United States Army, 2014, p. 42.
[35] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for unix processes," in *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, 1996, pp. 120-128.
[36] D. E. Denning, "An intrusion-detection model," *Software Engineering, IEEE Transactions on,* pp. 222-232, 1987.
[37] S. A. Hofmeyr and S. Forrest, "Architecture for an artificial immune system," *Evolutionary computation,* vol. 8, pp. 443-473, 2000.
[38] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of computer security,* vol. 6, pp. 151-180, 1998.
[39] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns," 2013.
[40] Bit9. (2015, October 18). *Bit9 and Carbon Black Endpoint Protection Solution*. Available: https://www.bit9.com/solutions/bit9-carbon-black-solution/
[41] FireEye. (2015, 18 October). *Endpoint Security*. Available: https://www.fireeye.com/products/hx-endpoint-security-products.html
[42] Bromium. (2015, 18 October). *Endpoint Security Products Overview*. Available: http://www.bromium.com/products.html
[43] MITRE Corporation, "A New Cyber Defense Playbook," MITRE Corporation,2012.
[44] SentinelOne. (2015, 18 October). *Next Generation Endpoint Protection*. Available: http://www.sentinelone.com/#
[45] Australian Signals Directorate and Defence Signals Directorate, "Top four mitigation strategies to protect your ICT system," Cyber Security Operations Centre, Ed., ed. Canberra, ACT: Department of Defence,, 2012, p. 2.
[46] H. Goldman, R. McQuaid, and J. Picciotto, "Cyber resilience for mission assurance," in *2011 IEEE International Conference on Technologies for Homeland Security (HST)*, 2011, pp. 236-241.
[47] US Air Force Space Command, "The United States Air Force Blueprint for Cyberspace," ed: Colorado Springs, CO, 2009.
[48] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors: The Journal of the Human Factors and Ergonomics Society,* vol. 37, pp. 32-64, 1995.
[49] P. Barford, M. Dacier, T. G. Dietterich, M. Fredrikson, J. Giffin, S. Jajodia*, et al.*, "Cyber SA: Situational awareness for cyber defense," in *Cyber Situational Awareness*, ed: Springer, 2010, pp. 3-13.
[50] H. Goldman, "Building secure, resilient architectures for cyber mission assurance," *The MITRE Corporation,* 2010.
[51] Symantec, "Symantec Data Loss Prevention Solution Data Sheet," Symantec, Mountain View, CA2015.
[52] RSA, "RSA Data Loss Prevention Data Sheet," RSA, USA2013.

[53] D. Shackleford and S. Northcutt, "Who's Using Cyberthreat Intelligence and How?," SANS Institute, Atlanta, USA2015.
[54] FireEye, "FireEye Advanced Threat Intelligence Plus Datasheet," FireEye, Ed., 1 ed. Milpitas, California: FireEye, 2015.
[55] S. Fitch and M. Muckin, "Defendable Architectures: Achieving Cyber Security By Designing For Intelligence Driven Defence," Lockheed Martin Corporation2015.
[56] MITRE Corporation, "Cyber Information Sharing Models: An Overview," Mitre Corporation,2012.
[57] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research,* vol. 1, p. 80, 2011.
[58] MITRE Corporation, "Cyber Observable eXpression (CybOX) - A Structured Language For Cyber Observables," M. Corporation, Ed., ed: MITRE Corporation, p. 2.
[59] S. Barnum, "Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX™)," *MITRE Corporation,* p. 11, 2012.
[60] MANDIANT, "openIOC White Paper," MANDIANT.
[61] R. Danyliw, J. Meijer, and Y. Demchenko, "RFC 5070: the incident object description exchange format," *Internet Engineering Task Force (IETF),* 2007.
[62] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, "Computer Security Incident Handling Guide," in *NIST Special Publication*, 2 ed. Gaithersburg, Maryland: National Institute for Standards and Technology, 2012, p. 71.
[63] S. Mitropoulos, D. Patsos, and C. Douligeris, "On Incident Handling and Response: A state-of-the-art approach," *Computers & Security,* vol. 25, pp. 351-370, 2006.
[64] A. Tolk, "Challenges of Combat Modelling and Distributed Siumlation," in *Engineering Principles of Combat Modelling and Distributed Simulation*, A. Tolk, Ed., ed. Hoboken, New Jersey: John Wiley and Sons, 2012, pp. 1-22.
[65] A. F. Machado, A. B. Barreto, and E. T. Yano, "Architecture for cyber defense simulator in military applications," presented at the 18th Annual Command & Control Research & Technology Symposium (ICCRTS), Alexandria, Virginia, USA, 2013.
[66] A. F. Machado and E. T. Yano, "Conceptual Architecture for Obtaining Cyber Situational Awareness," DTIC Document2014.
[67] A. Barros Barreto, P. C. Costa, and E. T. Yano, "A Semantic Approach to Evaluate the Impact of Cyber Actions on the Physical Domain," presented at the The Seventh International Conference on Semantic Technology for Intelligence Defence and Security, George mason University, Fiarfax, Virginia, 2012.
[68] A. Barros Barreto, P. Costa, and M. Hieb, "Cyber-Argus: Modelling C2 impacts of Cyber Attacks," in *19th International Command And Control Research and Technology Symposium*, Alexandria, Virginia, USA, 2014.
[69] G. Jakobson, "Mission cyber security situation assessment using impact dependency graphs," in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, 2011, pp. 1-8.
[70] G. Jakobson, "Extending situation modeling with inference of plausible future cyber situations," in *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on*, 2011, pp. 48-55.
[71] G. Jakobson, "Mission-centricity in cyber security: Architecting cyber attack resilient missions," presented at the Cyber Conflict (CyCon), 2013 5th International Conference on, Talinn, Estonia, 2013.
[72] M. Sudit, A. Stotz, M. Holender, W. Tagliaferri, and K. Canarelli, "Measuring situational awareness and resolving inherent high-level fusion obstacles," in *Defense and Security Symposium*, 2006, pp. 624205-624205-9.
[73] M. Sudit, A. Stotz, and M. Holender, "Situational awareness of a coordinated cyber attack," in *Defense and Security*, 2005, pp. 114-129.
[74] J. Holsopple, S. J. Yang, and M. Sudit, "TANDI: Threat assessment of network data and information," in *Defense and Security Symposium*, 2006, pp. 62420O-62420O-12.
[75] J. Holsopple and S. J. Yang, "FuSIA: Future situation and impact awareness," in *Information Fusion, 2008 11th International Conference on*, 2008, pp. 1-8.
[76] J. R. Goodall, A. D'Amico, and J. K. Kopylec, "Camus: Automatically mapping cyber assets to missions and users," in *Military Communications Conference, 2009. MILCOM 2009. IEEE*, 2009, pp. 1-7.

[77] A. D'Amico, L. Buchanan, J. Goodall, and P. Walczak, "Mission impact of cyber events: scenarios and ontology to express the relationships between cyber assets, missions and users," in *Proceedings of 5th International Conference on Information Warfare and Security*, 2010, pp. 8-9.

[78] S. J. Yang, A. Stotz, J. Holsopple, M. Sudit, and M. Kuhl, "High level information fusion for tracking and projection of multistage cyber attacks," *Information Fusion,* vol. 10, pp. 107-121, 2009.

[79] S. Moskal, D. Kreider, L. Hays, B. Wheeler, S. J. Yang, and M. Kuhl, "Simulating attack behaviors in enterprise networks," in *Communications and Network Security (CNS), 2013 IEEE Conference on*, 2013, pp. 359-360.

[80] S. Moskal, B. Wheeler, D. Kreider, M. E. Kuhl, and S. J. Yang, "Context model fusion for multistage network attack simulation," in *Military Communications Conference (MILCOM), 2014 IEEE*, Baltimore, MD, 2014, pp. 158-163.

[81] B. F. Wheeler, "A Computer Network Model for the Evaluation of Moving Target Network Defense Mechanisms," Master of Science (Computer Engineering) Masters, Rochester Institute Of Technology, Rochester, 2014.

[82] D. Kreider, "A Guidance Template for Attack Sequence Specification in Cyber Attack Simulation," Master of Science (Industrial Engineering) Masters, Department of Industrial and Systems Engineering, Rochester Institute of Technology, Rochester, 2015.

[83] R. Davis, H. Shrobe, and P. Szolovits, "What is a knowledge representation?," *AI magazine,* vol. 14, p. 17, 1993.

[84] G. Caldarelli and M. Catanzaro, *Networks: A Very Short Introduction*. Oxford, UK: Oxford University Press, 2012.

[85] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the atmospheric sciences,* vol. 20, pp. 130-141, 1963.

[86] R. H. Kewley and M. Wood, "Federated Simulation for Systems of Systems Engineering," in *Engineering Principles of Combat Modelling and Distributed Simulation*, A. Tolk, Ed., ed Hoboken, New Jersey: John Wiley and Sons, 2012, pp. 765-810.

[87] D. Ormrod, "A 'Wicked Problem' - Predicting SoS behaviour in Tactical Land Combat with Compromised C4ISR," presented at the 9th International Conference on System of Systems Engineering (SOSE), Adelaide, Australia, 2014.

[88] V. Raskin, C. F. Hempelmann, K. E. Triezenberg, and S. Nirenburg, "Ontology in information security: a useful theoretical foundation and methodological tool," in *Proceedings of the 2001 workshop on New security paradigms*, 2001, pp. 53-59.

[89] J. Undercoffer, J. Pinkston, A. Joshi, and T. Finin, "A target-centric ontology for intrusion detection," in *18th International Joint Conference on Artificial Intelligence*, 2004, pp. 9-15.

[90] L. Obrst, "Ontological Architectures," in *Theory and Applications of Ontology : Computer Applications*, R. Poli, M. Healy, and A. Kameas, Eds., ed. Dordrecht: Springer, 2010, pp. 27-66.

[91] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International journal of human-computer studies,* vol. 43, pp. 907-928, 1995.

[92] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," *The knowledge engineering review,* vol. 11, pp. 93-136, 1996.

[93] R. Neches, R. E. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator*, et al.*, "Enabling technology for knowledge sharing," *AI magazine,* vol. 12, p. 36, 1991.

[94] L. Obrst, W. Ceusters, I. Mani, S. Ray, and B. Smith, "The evaluation of ontologies," in *Semantic Web*, ed: Springer, 2007, pp. 139-158.

[95] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide To Developing Your First Ontology," Stanford Knowledge Systems Laboratory, Stanford2001.

[96] D. Dou, D. McDermott, and P. Qi, "Ontology translation on the semantic web," in *Journal on data semantics II*, ed: Springer, 2005, pp. 35-57.

[97] T. Gruber, "A Translation Approach To Portable Ontology Specifications," *Knowledge Acquisition,* vol. 5, pp. 199-220, 1993.

[98] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy*, et al.*, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 601-610.

[99] A. Singhal, "Introducing the knowledge graph: things, not strings," in *Official Google Blog, May* vol. 2015, ed: Google inc., 2012.

[100] J. Hendler, "Beyond OWL: Challenges for Ontologies on the Semantic Web," ed. Rensselaer Polytechnic Institute: Tetherless World Constellation, 2015, p. 44.
[101] D. Lenat, "Hal's Legacy: 2001's Computer as Dream and Reality. From 2001 to 2001: Common Sense and the Mind of HAL," *Cycorp, Inc.,: http://www. cyc. com/cyc/technology/halslegacy. html,* 2006.
[102] D. B. Lenat, "CYC: A large-scale investment in knowledge infrastructure," *Communications of the ACM,* vol. 38, pp. 33-38, 1995.
[103] C. Deaton, B. Shepard, C. Klein, C. Mayans, B. Summers, A. Brusseau, *et al.*, "The comprehensive terrorism knowledge base in cyc," in *Proceedings of the 2005 International Conference on Intelligence Analysis, McLean, Virginia*, 2005.
[104] D. Lenat, M. Witbrock, D. Baxter, E. Blackstone, C. Deaton, D. Schneider, *et al.*, "Harnessing Cyc to answer clinical researchers' ad hoc queries," *AI Magazine,* vol. 31, pp. 13-32, 2010.
[105] C. Matuszek, J. Cabral, M. J. Witbrock, and J. DeOliveira, "An Introduction to the Syntax and Content of Cyc," in *AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, 2006, pp. 44-49.
[106] B. Shepard, C. Matuszek, C. B. Fraser, W. Wechtenhiser, D. Crabbe, Z. Güngördü, *et al.*, "A Knowledge-based approach to network security: applying Cyc in the domain of network risk assessment," 2005.
[107] W. D. Hillis, "Aristotle (the knowledge web)," *Edge Foundation, Inc,* vol. 138, 2004.
[108] O. Corcho, M. Fernández-López, A. Gómez-Pérez, and A. López-Cima, "Building legal ontologies with METHONTOLOGY and WebODE," in *Law and the semantic web*, ed: Springer, 2005, pp. 142-157.
[109] M. F. López, A. Gómez-Pérez, J. P. Sierra, and A. P. Sierra, "Building a chemical ontology using methontology and the ontology design environment," *IEEE intelligent Systems,* pp. 37-46, 1999.
[110] A. Sawsaa and J. Lu, "Building Information Science ontology (OIS) with Methontology and Protégé," *Journal of Internet Technology and Secured Transactions (JITST),* vol. 1, 2012.
[111] A. Souag, C. Salinesi, R. Mazo, and I. Comyn-Wattiau, "A security ontology for security requirements elicitation," in *Engineering Secure Software and Systems*, ed: Springer, 2015, pp. 157-177.
[112] E. Prestes, S. R. Fiorini, and J. Carbonera, "Core Ontology for Robotics and Automation," presented at the Proceedings of the First Standardized Knowledge Representaion and Ontologies for Robotics and Automation, Chicago, Illinois, 2014.
[113] N. Schuurman and A. Leszczynski, "Ontologies for Bioinformatics," *Bioinformatics and Biology Insights,* vol. 2, pp. 187-200, 03/12 2008.
[114] B. Tsoumas and D. Gritzalis, "Towards an ontology-based security management," in *20th International Conference on Advanced Information Networking and Applications, 2006.* , 2006, pp. 985-992.
[115] A. Oltramari, L. F. Cranor, R. J. Walls, and P. McDaniel, "Building An Ontology Of Cyber Security," presented at the The Ninth International Conference for Semantic Technology for Intelligence, Defence and Security, George Mason University, Fairfax, Virginia, United States of America, 2014.
[116] M. Corp. (2015, 18 October). *The Common Platform Enumeration Specification*. Available: https://cpe.mitre.org/specification/
[117] M. Corp. (2015, 18 October). *The Common Weakness Enumeration*. Available: https://cwe.mitre.org/
[118] R. Martin. (2001) Managing vulnerabilities in networked systems. *IEEE Computer Society Computer Magazine*. 32-38. Available: https://cve.mitre.org/docs/docs-2001/CVEarticleIEEEcomputer.pdf
[119] MITRE. (2015, 26 Jun 2015). *Common Vulnerabilities and Exposures Database*. Available: https://cve.mitre.org/
[120] T. Takahashi, H. Fujiwara, and Y. Kadobayashi, "Building ontology of cybersecurity operational information," in *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, 2010, p. 79.
[121] T. Takahashi, Y. Kadobayashi, and H. Fujiwara, "Ontological approach toward cybersecurity in cloud computing," in *Proceedings of the 3rd international conference on Security of information and networks*, 2010, pp. 100-109.
[122] L. Obrst, P. Chase, and R. Markeloff, "Developing an Ontology of the Cyber Security Domain," in *The 7th International Conference on Semantic Technology for Intelligence,*

Defense, and Security, Geroge Mason University, Fairfax Virginia, United States of America, 2012, pp. 49-56.
[123] Mitre Corporation. (2015, 26 October). *Malware Attribute Characterisation and Enumeration*. Available: https://maec.mitre.org/
[124] I. Moskowitz and M. Kang, "An Insecurity Flow model," in *New Security Paradigms Workshop*, Langdale, Cumbria, United Kingdom, 1997, pp. 61-74.
[125] C. Phillips and L. P. Swiler, "A graph-based system for network-vulnerability analysis," presented at the Proceedings of the 1998 workshop on New security paradigms, Charlottesville, Virginia, USA, 1998.
[126] L. P. Swiler, C. Phillips, D. Ellis, and S. Chakerian, "Computer-attack graph generation tool," in *DISCEX'01. Proceedings DARPA Information Survivability Conference Exposition II, 2001.* , 2001, pp. 307-321.
[127] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. M. Wing, "Automated generation and analysis of attack graphs," in *Security and privacy, 2002. Proceedings. 2002 IEEE Symposium on*, 2002, pp. 273-284.
[128] D. Fava, J. Holsopple, S. J. Yang, and B. Argauer, "Terrain and behavior modeling for projecting multistage cyber attacks," in *10th International Conference on Information Fusion, 2007* 2007, pp. 1-7.
[129] J. Holsopple, S. Yang, and B. Argauer, "Virtual terrain: a security-based representation of a computer network," in *SPIE Defense and Security Symposium*, 2008, pp. 69730E-69730E-10.
[130] G. Lyon. (2015, 07 October). *NMAP: The Network Mapper*. Available: https://nmap.org/
[131] (2015, 07 October). *Nessus Vulnerability Scanner*. Available: http://www.tenable.com/products/nessus-vulnerability-scanner
[132] B. J. Argauer and S. J. Yang, "VTAC: Virtual terrain assisted impact assessment for cyber attacks," in *SPIE Defense and Security Symposium*, 2008, pp. 69730F-69730F-12.
[133] B. Argauer, "VTAC: Virtual Terrain Assisted Impact Assessment for Cyber Attacks," Master of Science (Computer Engineering) Masters, Department of Computer Engineering, Rochester Institute of Technology, Rochester, 2007.
[134] Resource Description Framework Working Group. (2014, 18 October). *RDF 1.1 Turtle* [W3C Recommendation]. Available: http://www.w3.org/TR/turtle/
[135] M. Davidson and C. Schmidt, "TAXII Overview," The MITRE Corporation2014.
[136] D. Grove, A. Murray, D. Gerhardy, B. Turnbull, T. Tobin, and C. Moir, "An overview of the parallax BattleMind v1. 5 for computer network defence," in *Proceedings of the Eleventh Australasian Information Security Conference-Volume 138*, 2013, pp. 31-37.
[137] MITRE Corporation. (2015, 07 Jun 15). *Support for STIX*. Available: http://stixproject.github.io/supporters/
[138] A. Sykosch and M. Wubbeling, "STIX 2 IDS," 2015.
[139] M. Hammell, "The Role of ThreatExchange," in *Spam Fighting @ Scale*, ed: @ Scale, 2015, p. 27:23.
[140] Facebook. (2015, 26 October 15). *Facebook ThreatExchange GitHub Repository*. Available: https://github.com/facebook/ThreatExchange/
[141] Facebook. (2015, 26 October). *Facebook ThreatExhange API Reference*. Available: https://developers.facebook.com/docs/threat-exchange/reference/apis/v2.5
[142] CYCORP, "CycSecure Backgrounder," Cycorp, Austin, Texas05 November 2001 2001.
[143] P. Gruba and J. Zobel, *How to Write a Better Minor Thesis*. Carlton, Victoria: Melbourne University Publishing, 2014.
[144] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, 4 ed. Thousand Oaks, California: Sage, 2014.
[145] $(ISC)^2$, *Official $(ISC)^2$ Guide to the CISSP CBK*, Fourth Edition ed. Boca Raton, FL: CRC Press, 2015.
[146] V. Bagad and R. Dhotre, *Computer Networks*. Pune, India: Technical Publications Pune, 2010.
[147] A. Tanenbaum and D. Wetherall, *Computer Networks*, Fifth Edition ed. Boston, Massachusetts: Prentice Hall, 2011.
[148] K. Beck, *Test-driven development: by example*: Addison-Wesley Professional, 2003.
[149] G. Dodig-Crnkovic, "Scientific Methods in Computer Science," in *Proceedings of the Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden, Skövde, Suecia*, 2002, pp. 126-130.
[150] C. Vidal, "What is a worldview?," *De wetenschappen en het creatieve aspect van de werkelijkheid,* 2008.

[151] R. Cole, S. Purao, M. Rossi, and M. Sein, "Being proactive: where action research meets design research," *ICIS 2005 Proceedings,* p. 27, 2005.
[152] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, "Methontology: from ontological art towards ontological engineering," presented at the 1997 AAAI Spring Symposium on Ontological Engineering, 1997.
[153] IEEE, "IEEE Standard for Developing a Software Project Life Cycle Process," in *IEEE Std 1074-2006 (Revision of IEEE Std 1074-1997)*, ed. New York, NY: IEEE Computer Society, 2006, pp. 1-104.
[154] Resource Description Framework Working Group. (2014, 18 October). *Resource Description Framework*. Available: https://www.w3.org/RDF/
[155] Web Ontology Language Working Group, "Web Ontology Language," ed: World Wide Web Consortium, 2012.
[156] Google Inc. (2015, 03 Nov 15). *Freebase: A community-curated database of well-known people, places, and things*. Available: https://www.freebase.com/
[157] R. H. von Alan, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly,* vol. 28, pp. 75-105, 2004.
[158] S. Braun, A. P. Schmidt, A. Walter, G. Nagypal, and V. Zacharias, "Ontology Maturing: a Collaborative Web 2.0 Approach to Ontology Engineering," *Ckc,* vol. 273, 2007.
[159] A. Schmidt, "Knowledge maturing and the continuity of context as a unifying concept for knowledge management and e-learning," in *Proceedings of I-Know*, 2005, pp. 122-136.
[160] Y. Ren, A. Parvizi, C. Mellish, J. Z. Pan, K. Van Deemter, and R. Stevens, "Towards competency question-driven ontology authoring," in *11th International European Semantic Web Conference*, Anissaras, Crete, Greece, 2014, pp. 752-767.
[161] C. Sousa, A. L. Soares, C. Pereira, and S. Moniz, "Establishing Conceptual Commitments in the Development of Ontologies through Competency Questions and Conceptual Graphs," in *On the Move to Meaningful Internet Systems: OTM 2014 Workshops*, 2014, pp. 626-635.
[162] C. Inc. (2015, 18 Oct). *Stardog*. Available: http://stardog.com/
[163] SPARQL Protocol and RDF Query Language Working Group. (2015, 18 October). *SPARQL Protocol and RDF Query Language 1.1 Recommendation*. Available: http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/
[164] D. Ormrod, B. Turnbull, and K. O'Sullivan, "Systems of Systems: Cyber Effects Simulation Ontology," presented at the TBC, TBC, 2015.
[165] Chief Information Officer - United States Army, "Deployed Tactical Network Guidance (Appendix D to Guidance for End State for Army Enterprise Network Architecture)," Department of the Army, Ed., ed: United States Army, 2012, p. 50.
[166] Chief Information Officer of the United States Army, "LandWarNet2020 and Beyond Enterprise Architecture," Department of the Army, Ed., ed: United States Army,, 2013, p. 23.
[167] Forum of Incident Response and Security Teams CVSS Special Interest Group. (2015, 18 October). *Common Vulnerability Scoring System Version 3.0 Specification Document*. Available: https://www.first.org/cvss/cvss-v30-specification-v1.7.pdf
[168] Offensive Security. (2015, 23 June). *The Exploit Database*. Available: https://www.exploit-db.com/
[169] Offensive Security. Offensive Security ExploitDatabase [Online]. Available: https://www.exploit-db.com/
[170] MITRE Corporation. (2015, 31 Oct). *CVE Reference Map for Exploit DB*. Available: https://cve.mitre.org/data/refs/refmap/source-EXPLOIT-DB.html
[171] N. Humfrey. (2015, 18 October). *easyRDF Converter*. Available: http://www.easyrdf.org/converter
[172] S. Goyal and R. Westenthaler. (2004, 14 October). *Rdf Gravity (rdf graph visualization tool)*. Available: http://semweb.salzburgresearch.at/apps/rdf-gravity/
[173] Homeland Security Systems Engineering and Development Institute, "Structured Threat Information Expression Overview," ed: MITRE Corporation, 2014.
[174] R. Stallman, *Free Software, Free Society: selected essays of Richard M. Stallman*. Boston, MA: Free Software Foundation, 2002.
[175] C. Moir and J. Dean, "A Machine Learning approach to Generic Entity Resolution in support of Cyber Situation Awareness," in *Proceedings of the 38th Australasian Computer Science Conference (ACSC 2015)*, 2015, p. 30.
[176] J. Yuen, B. Turnbull, and J. Hernandez, "Visual Analytics for Cyber Red Teaming," presented at the Paper In Press, 2015.

[177] R. A. Martino, "Leveraging traditional battle damage assessment procedures to measure effects from a computer network attack," DTIC Document2011.
[178] R. Ostler, "Defensive cyber battle damage assessment through attack methodology modeling," DTIC Document2011.
[179] N. Rose, "Shaping the Future Battlespace: Offensive Cyber Warfare Tools for the Planner," *Australian Army Journal,* vol. 10, pp. 53-68, 2013.

Appendix A – Results of Competency Questions

```
#     _____                          _                          ____ _
#    / ____|                        | |                        / __ \(_)
#   | |     ___  _ __ ___  _ __   ___| |_ ___ _ __   ___ _   _  | |  | |/ ___
#   | |    / _ \| '_ ` _ \| '_ \ / _ \ __/ _ \ '_ \ / __| | | | | |  | | / __|
#   | |___| (_) | | | | | | |_) |  __/ ||  __/ | | | (__| |_| | | |__| | \__ \
#    \_____\___/|_| |_| |_| .__/ \___|\__\___|_| |_|\___|\__, |  \___\_\ |___/
#                         | |                             __/ |
#
```

1. Which Nodes are visible in the ontology?
2. Which Networks are visible?
3. Which Subnetwork does each node belong to?
4. What is the IP address of each node on the network?
5. What software is funning on which machine?
6. Which services are remote and what are they projecting?
7. Which nodes on the network are running vulnerable software?
8. What weakness encompasses this vulnerability?
9. How difficult is it for an attacker to exploit this vulnerability?
10. Which computers represented in the ontology are members of a domain?
11. Can a domain user access any node that is a member of the domain?
12. Who are all the users of a given domain?
13. Which users are administrators and which are normal users?
14. Which nodes are running HIDS?
15. Which subnetworks are running NIDS?
16. Can a HIDS detect a vulnerability or exploit present in the terrain?
17. Which computers have storage disks?
18. Which disks have Data?
19. Which disks are encrypted
20. Who owns piece of data X?
21. Which networks are wirelessly accessibele?
22. Does a portscan return both wired and wireless connections together?
23. What is the wireless security protocol that is being used to protect a given network?
24. Which nodes on the network are virtual machines?
25. What are all the active VPN Connections?
26. Can a virtual machine be seen as part of a physical network?
27. Which exploit maps to which vulnerability?
28. Who are the threat actors involved in an incident?
29. Which organisation do the threats belong to?
30. What course of action is available to address an exploit?
31. What effect has an exploit had on the system it is targeting?

#Note - All queries below are formatted for use by the stardog query engine. They are still using SPARQL.

1.
./stardog query -r Evalusecase_DB "SELECT ?Node WHERE {?Node rdf:type cst:Node}"

Node
usecase:JointFiresRouter
usecase:JointFiresDomain
usecase:JointFiresDomainController
usecase:JFT_Computer
usecase:JFCC_Computer
usecase:C_BTY_HQ_Computer
usecase:C_BTY_1TP_Computer
usecase:C_BTY_2TP_Computer
usecase:C_BTY_3TP_Computer
usecase:JFT_Laptop
usecase:NIDS_JFCCNET
usecase:AttackerVirtualServer
usecase:Attacker1_VM
usecase:Attacker2_VM

#N.b. - inferrencing must be enabled for this query to work. Note how it shows all nodes inclusing VMs, the NIDS and the DC

2. Which Networks are visible?
./stardog query -r Evalusecase_DB "SELECT ?Network WHERE {?Network rdf:type cst:Network}"

Network
usecase:JFCCNET
usecase:JFTNET
usecase:C_BTYNET
usecase:AttackerNet
usecase:JFTWAP
usecase:AttackerNET
usecase:AttackerVirtualNetwork

#Note: Shows virtualNet too - an refine by asking for just subnets

3. Which Subnetwork does each node belong to?

```
./stardog query -r Evalusecase_DB "SELECT ?Network ?Node WHERE {?Network rdf:type cst:Network . ?Node cst:hasNic ?Nic . ?Nic
cst:isConnectedTo ?Network}"
+--------------------+-----------------------------------+
|       Network      |                Node               |
+--------------------+-----------------------------------+
| usecase:JFTNET     | usecase:JointFiresRouter          |
| usecase:JFCCNET    | usecase:JointFiresRouter          |
| usecase:C_BTYNET   | usecase:JointFiresRouter          |
| usecase:AttackerNet| usecase:JointFiresRouter          |
| usecase:JFCCNET    | usecase:JointFiresDomainController|
| usecase:JFTWAP     | usecase:JFT_Laptop                |
| usecase:JFCCNET    | usecase:JFCC_Computer             |
| usecase:C_BTYNET   | usecase:C_BTY_HQ_Computer         |
| usecase:C_BTYNET   | usecase:C_BTY_1TP_Computer        |
| usecase:C_BTYNET   | usecase:C_BTY_2TP_Computer        |
| usecase:C_BTYNET   | usecase:C_BTY_3TP_Computer        |
| usecase:AttackerNET| usecase:AttackerVirtualServer     |
+--------------------+-----------------------------------+
```

4. What is the IP address of each node on the network?

```
./stardog query -r Evalusecase_DB "SELECT ?Network ?Node ?IPAddress WHERE {?Network rdf:type cst:Network . ?Node cst:hasNic ?Nic . ?Nic
cst:isConnectedTo ?Network . ?Nic cst:ipAddress ?IPAddress} ORDER BY ASC(?IPAddress)"
+--------------------+-----------------------------------+----------------+
|       Network      |                Node               |   IPAddress    |
+--------------------+-----------------------------------+----------------+
| usecase:JFTWAP     | usecase:JFT_Laptop                | "10.10.0.1"    |
| usecase:JFTNET     | usecase:JointFiresRouter          | "10.10.0.200"  |
| usecase:JFCCNET    | usecase:JFCC_Computer             | "10.10.1.1"    |
| usecase:JFCCNET    | usecase:JointFiresDomainController| "10.10.1.100"  |
| usecase:JFCCNET    | usecase:JointFiresRouter          | "10.10.1.200"  |
| usecase:C_BTYNET   | usecase:C_BTY_1TP_Computer        | "10.10.3.1"    |
| usecase:C_BTYNET   | usecase:C_BTY_2TP_Computer        | "10.10.3.2"    |
| usecase:C_BTYNET   | usecase:JointFiresRouter          | "10.10.3.200"  |
| usecase:C_BTYNET   | usecase:C_BTY_3TP_Computer        | "10.10.3.3"    |
| usecase:C_BTYNET   | usecase:C_BTY_HQ_Computer         | "10.10.3.9"    |
| usecase:AttackerNET| usecase:AttackerVirtualServer     | "10.10.66.10"  |
| usecase:AttackerNet| usecase:JointFiresRouter          | "10.10.66.200" |
+--------------------+-----------------------------------+----------------+
#Note how we can see them ordered by subnet IAW the laptop. Also not how JFT laptop appears as part of the search for networks! and how
it's not there twice because of disjoint property.

###
```

5. What software is running on which machine?

```
./stardog query -r Evalusecase_DB "SELECT ?Software ?Computer WHERE {?Computer cst:hasInstalledSoftwareVersion ?Software }"
+-----------------------------------------+----------------------------+
|                Software                 |          Computer          |
+-----------------------------------------+----------------------------+
| usecase:MissionControlSoftwareV3_0      | usecase:JFCC_Computer      |
| usecase:OpenSSL1_0_1                    | usecase:JFCC_Computer      |
| usecase:OpenSSL1_0_1                    | usecase:C_BTY_HQ_Computer  |
| usecase:BatteryManagementSoftwareV16_8_8 | usecase:C_BTY_HQ_Computer  |
| usecase:FireControlSoftwareV2_1         | usecase:C_BTY_1TP_Computer |
| usecase:HIDS1_1_1                       | usecase:C_BTY_1TP_Computer |
| usecase:FireControlSoftwareV2_1         | usecase:C_BTY_2TP_Computer |
| usecase:HIDS1_1_1                       | usecase:C_BTY_2TP_Computer |
| usecase:FireControlSoftwareV2_1         | usecase:C_BTY_3TP_Computer |
| usecase:HIDS1_1_1                       | usecase:C_BTY_3TP_Computer |
| usecase:CallForFireCommsSoftwareV5_7    | usecase:JFT_Laptop         |
| usecase:TargetingSoftwareV1_5_5         | usecase:JFT_Laptop         |
| usecase:NIDS1_1_1                       | usecase:NIDS_JFCCNET       |
+-----------------------------------------+----------------------------+
```

6. Which services are remote and what are they projecting?

```
./stardog query -r Evalusecase_DB "SELECT ?Computer ?RemoteService ?Port ?Protocol  WHERE {?Computer cst:hasInstalledSoftwareVersion ?
Software . ?Software cst:projectsService ?RemoteService . ?RemoteService cst:port ?Port . ?RemoteService cst:protocol ?Protocol}"
+---------------------------+--------------------------+----------+-----------+
|          Computer         |       RemoteService      |   Port   | Protocol  |
+---------------------------+--------------------------+----------+-----------+
| usecase:JFT_Laptop        | usecase:CallForFireComms | "8766"   | "CFFC"    |
| usecase:JFCC_Computer     | usecase:MissionController | "7331"   | "MsnC"    |
| usecase:JFCC_Computer     | usecase:SSL              | "443"    | "SSL"     |
| usecase:C_BTY_HQ_Computer | usecase:SSL              | "443"    | "SSL"     |
| usecase:C_BTY_HQ_Computer | usecase:BatteryManager   | "8765"   | "BTYMan"  |
| usecase:C_BTY_1TP_Computer | usecase:FireController   | "161616" | "FireMan" |
| usecase:C_BTY_2TP_Computer | usecase:FireController   | "161616" | "FireMan" |
| usecase:C_BTY_3TP_Computer | usecase:FireController   | "161616" | "FireMan" |
+---------------------------+--------------------------+----------+-----------+
```

7. Which nodes on the network are running vulnerable software?

```
./stardog query Evalusecase_DB "SELECT ?Node ?VulnerableSoftware WHERE {?VulnerableSoftware cst:hasVulnerability ?Vulnerability . ?Node
cst:hasInstalledSoftwareVersion ?VulnerableSoftware}"
+--------------------------+--------------------------+
|           Node           |    VulnerableSoftware    |
+--------------------------+--------------------------+
| usecase:JFCC_Computer    | usecase:OpenSSL1_0_1     |
| usecase:C_BTY_HQ_Computer | usecase:OpenSSL1_0_1     |
| usecase:C_BTY_HQ_Computer | usecase:AdobeReader8_1_3 |
+--------------------------+--------------------------+
```

8. What weakness encompasses this vulnerability?

```
./stardog query Evalusecase_DB "SELECT ?VulnerableSoftware ?Vulnerability ?Weakness WHERE {?VulnerableSoftware cst:hasVulnerability ?
Vulnerability . ?Weakness cst:includesVulnerability ?Vulnerability}"

+--------------------------+----------------------------+-------------------------------------------------------------------------+
|    VulnerableSoftware    |        Vulnerability       |                                Weakness                                 |
+--------------------------+----------------------------+-------------------------------------------------------------------------+
| usecase:OpenSSL1_0_1     | usecase:Heartbleed         | usecase:FailureToConstrainOperationsWithinTheBoundsOfAMemeoryBuffer |
| usecase:AdobeReader8_1_3 | usecase:AdobeBufferOverflow | usecase:InputValidation                                                 |
+--------------------------+----------------------------+-------------------------------------------------------------------------
```

9. How difficult is it for an attacker to exploit this vulnerability?

```
./stardog query Evalusecase_DB "SELECT ?Node ?VulnerableSoftware ?Vulnerability ?ExploitabilityScore WHERE {?Node
cst:hasInstalledSoftwareVersion ?VulnerableSoftware .?VulnerableSoftware cst:hasVulnerability ?Vulnerability . ?Vulnerability
cst:cvssExploitabilityScore ?ExploitabilityScore}"
+--------------------------+----------------------+--------------------+----------------------+
|           Node           |  VulnerableSoftware  |   Vulnerability    | ExploitabilityScore |
+--------------------------+----------------------+--------------------+----------------------+
| usecase:JFCC_Computer    | usecase:OpenSSL1_0_1 | usecase:Heartbleed | "10"                 |
| usecase:C_BTY_HQ_Computer | usecase:OpenSSL1_0_1 | usecase:Heartbleed | "10"                 |
+--------------------------+----------------------+--------------------+----------------------+
```

10. Which computers represented in the ontology are members of a domain?

```
./stardog query Evalusecase_DB "SELECT ?Domain ?Computer WHERE {?Domain rdf:type cst:Domain . ?Domain cst:hasAccessTo ?Computer}"
+--------------------------+-------------------------------------+
|          Domain          |              Computer               |
+--------------------------+-------------------------------------+
| usecase:JointFiresDomain | usecase:JointFiresDomainController |
| usecase:JointFiresDomain | usecase:JFT_Computer                |
| usecase:JointFiresDomain | usecase:JFCC_Computer               |
| usecase:JointFiresDomain | usecase:C_BTY_HQ_Computer           |
| usecase:JointFiresDomain | usecase:C_BTY_1TP_Computer          |
| usecase:JointFiresDomain | usecase:C_BTY_2TP_Computer          |
| usecase:JointFiresDomain | usecase:C_BTY_3TP_Computer          |
+--------------------------+-------------------------------------+
```

11. Can a domain user access any node that is a member of the domain?

```
./stardog query Evalusecase_DB "SELECT ?Domain ?Computer ?User WHERE {?Domain rdf:type cst:Domain . ?Domain cst:hasAccessTo ?Computer . ?
User cst:hasAccessTo ?Domain}"
+--------------------------+-------------------------------------+-----------------------------+
|          Domain          |              Computer               |            User             |
+--------------------------+-------------------------------------+-----------------------------+
| usecase:JointFiresDomain | usecase:JointFiresDomainController | usecase:ForwardObserver     |
| usecase:JointFiresDomain | usecase:JointFiresDomainController | usecase:JointFirecontroller |
| usecase:JointFiresDomain | usecase:JointFiresDomainController | usecase:BCChraliebattery    |
| usecase:JointFiresDomain | usecase:JointFiresDomainController | usecase:1Troopcommander     |
| usecase:JointFiresDomain | usecase:JointFiresDomainController | usecase:2Troopcommander     |
| usecase:JointFiresDomain | usecase:JointFiresDomainController | usecase:3Troopcommander     |
| usecase:JointFiresDomain | usecase:JointFiresDomainController | usecase:S6Admin             |
| usecase:JointFiresDomain | usecase:JFT_Computer                | usecase:ForwardObserver     |
| usecase:JointFiresDomain | usecase:JFT_Computer                | usecase:JointFirecontroller |
| usecase:JointFiresDomain | usecase:JFT_Computer                | usecase:BCChraliebattery    |
| usecase:JointFiresDomain | usecase:JFT_Computer                | usecase:1Troopcommander     |
| usecase:JointFiresDomain | usecase:JFT_Computer                | usecase:2Troopcommander     |
| usecase:JointFiresDomain | usecase:JFT_Computer                | usecase:3Troopcommander     |
| usecase:JointFiresDomain | usecase:JFT_Computer                | usecase:S6Admin             |
| usecase:JointFiresDomain | usecase:JFCC_Computer               | usecase:ForwardObserver     |
```

```
| usecase:JointFiresDomain | usecase:JFCC_Computer             | usecase:JointFirecontroller |
| usecase:JointFiresDomain | usecase:JFCC_Computer             | usecase:BCChraliebattery    |
| usecase:JointFiresDomain | usecase:JFCC_Computer             | usecase:1Troopcommander     |
| usecase:JointFiresDomain | usecase:JFCC_Computer             | usecase:2Troopcommander     |
| usecase:JointFiresDomain | usecase:JFCC_Computer             | usecase:3Troopcommander     |
| usecase:JointFiresDomain | usecase:JFCC_Computer             | usecase:S6Admin             |
| usecase:JointFiresDomain | usecase:C_BTY_HQ_Computer         | usecase:ForwardObserver     |
| usecase:JointFiresDomain | usecase:C_BTY_HQ_Computer         | usecase:JointFirecontroller |
| usecase:JointFiresDomain | usecase:C_BTY_HQ_Computer         | usecase:BCChraliebattery    |
| usecase:JointFiresDomain | usecase:C_BTY_HQ_Computer         | usecase:1Troopcommander     |
| usecase:JointFiresDomain | usecase:C_BTY_HQ_Computer         | usecase:2Troopcommander     |
| usecase:JointFiresDomain | usecase:C_BTY_HQ_Computer         | usecase:3Troopcommander     |
| usecase:JointFiresDomain | usecase:C_BTY_HQ_Computer         | usecase:S6Admin             |
| usecase:JointFiresDomain | usecase:C_BTY_1TP_Computer        | usecase:ForwardObserver     |
| usecase:JointFiresDomain | usecase:C_BTY_1TP_Computer        | usecase:JointFirecontroller |
| usecase:JointFiresDomain | usecase:C_BTY_1TP_Computer        | usecase:BCChraliebattery    |
| usecase:JointFiresDomain | usecase:C_BTY_1TP_Computer        | usecase:1Troopcommander     |
| usecase:JointFiresDomain | usecase:C_BTY_1TP_Computer        | usecase:2Troopcommander     |
| usecase:JointFiresDomain | usecase:C_BTY_1TP_Computer        | usecase:3Troopcommander     |
| usecase:JointFiresDomain | usecase:C_BTY_1TP_Computer        | usecase:S6Admin             |
| usecase:JointFiresDomain | usecase:C_BTY_2TP_Computer        | usecase:ForwardObserver     |
| usecase:JointFiresDomain | usecase:C_BTY_2TP_Computer        | usecase:JointFirecontroller |
| usecase:JointFiresDomain | usecase:C_BTY_2TP_Computer        | usecase:BCChraliebattery    |
| usecase:JointFiresDomain | usecase:C_BTY_2TP_Computer        | usecase:1Troopcommander     |
| usecase:JointFiresDomain | usecase:C_BTY_2TP_Computer        | usecase:2Troopcommander     |
| usecase:JointFiresDomain | usecase:C_BTY_2TP_Computer        | usecase:3Troopcommander     |
| usecase:JointFiresDomain | usecase:C_BTY_2TP_Computer        | usecase:S6Admin             |
| usecase:JointFiresDomain | usecase:C_BTY_3TP_Computer        | usecase:ForwardObserver     |
| usecase:JointFiresDomain | usecase:C_BTY_3TP_Computer        | usecase:JointFirecontroller |
| usecase:JointFiresDomain | usecase:C_BTY_3TP_Computer        | usecase:BCChraliebattery    |
| usecase:JointFiresDomain | usecase:C_BTY_3TP_Computer        | usecase:1Troopcommander     |
| usecase:JointFiresDomain | usecase:C_BTY_3TP_Computer        | usecase:2Troopcommander     |
| usecase:JointFiresDomain | usecase:C_BTY_3TP_Computer        | usecase:3Troopcommander     |
| usecase:JointFiresDomain | usecase:C_BTY_3TP_Computer        | usecase:S6Admin             |
+--------------------------+-----------------------------------+-----------------------------+
```

12. Who are all the users of a given domain?

```
./stardog query Evalusecase_DB "SELECT ?Domain ?User WHERE {?Domain rdf:type cst:Domain .?User cst:hasAccessTo ?Domain}"
+--------------------------+-----------------------------+
|          Domain          |            User             |
+--------------------------+-----------------------------+
| usecase:JointFiresDomain | usecase:ForwardObserver     |
| usecase:JointFiresDomain | usecase:JointFirecontroller |
| usecase:JointFiresDomain | usecase:BCChraliebattery    |
| usecase:JointFiresDomain | usecase:1Troopcommander     |
| usecase:JointFiresDomain | usecase:2Troopcommander     |
| usecase:JointFiresDomain | usecase:3Troopcommander     |
| usecase:JointFiresDomain | usecase:S6Admin             |
+--------------------------+-----------------------------+
```

13. Which users are administrators and which are normal users?

```
./stardog query Evalusecase_DB "SELECT ?Domain ?User ?PrivilegeLevel WHERE {?Domain rdf:type cst:Domain .?User cst:hasAccessTo ?Domain . ?
User cst:userPrivilegeLevel ?PrivilegeLevel}"
+--------------------------+-----------------------------+-----------------+
|          Domain          |            User             | PrivilegeLevel  |
+--------------------------+-----------------------------+-----------------+
| usecase:JointFiresDomain | usecase:ForwardObserver     | "USER"          |
| usecase:JointFiresDomain | usecase:JointFirecontroller | "USER"          |
| usecase:JointFiresDomain | usecase:BCChraliebattery    | "USER"          |
| usecase:JointFiresDomain | usecase:1Troopcommander     | "USER"          |
| usecase:JointFiresDomain | usecase:2Troopcommander     | "USER"          |
| usecase:JointFiresDomain | usecase:3Troopcommander     | "USER"          |
| usecase:JointFiresDomain | usecase:S6Admin             | "ADMINISTRATOR" |
+--------------------------+-----------------------------+-----------------+
```

14. Which nodes are running HIDS?

```
./stardog query -r Evalusecase_DB "SELECT ?Node ?Software WHERE {?Node cst:hasInstalledSoftwareVersion ?Software . ?Software
cst:isSoftwareType usecase:HIDS}"
+----------------------------+-------------------+
|            Node            |     Software      |
+----------------------------+-------------------+
| usecase:C_BTY_HQ_Computer  | usecase:HIDS1_1_1 |
| usecase:C_BTY_1TP_Computer | usecase:HIDS1_1_1 |
| usecase:C_BTY_2TP_Computer | usecase:HIDS1_1_1 |
| usecase:C_BTY_3TP_Computer | usecase:HIDS1_1_1 |
+----------------------------+-------------------+
```

15. Which subnetworks are running NIDS?

```
./stardog query -r Evalusecase_DB "SELECT ?Subnetwork ?NIDS WHERE {?NIDS cst:monitorsSubnet ?Subnetwork}"
+-----------------+----------------------+
|   Subnetwork    |         NIDS         |
+-----------------+----------------------+
| usecase:JFCCNET | usecase:NIDS_JFCCNET |
+-----------------+----------------------+
```

16. Can a HIDS detect a vulnerability or exploit present in the terrain?

```
./stardog query -r Evalusecase_DB "SELECT ?SoftwareCompromised ?Vulnerability ?Rule ?Signature ?Action  WHERE {usecase:C_BTY_HQ_Computer
cst:hasInstalledSoftwareVersion ?IDS . ?IDS cst:hasSoftwareConfiguration ?Rule . usecase:C_BTY_HQ_Computer
cst:hasInstalledSoftwareVersion ?SoftwareCompromised . ?SoftwareCompromised cst:hasVulnerability ?Vulnerability . ?Vulnerability
cst:cveIdentifier ?Signature . ?Rule cst:idsSignature ?Signature . ?Rule cst:idsAction ?Action}"

+--------------------------+----------------------------+-----------------------------------+-----------------+---------+
|   SoftwareCompromised    |        Vulnerability       |                Rule               |    Signature    | Action  |
+--------------------------+----------------------------+-----------------------------------+-----------------+---------+
| usecase:AdobeReader8_1_3 | usecase:AdobeBufferOverflow | usecase:AdobeBufferOverflowWarning | "CVE-2009-0927" | "ALERT" |
+--------------------------+----------------------------+-----------------------------------+-----------------+---------+
```

###

17. Which computers have storage disks?

```
./stardog query -r Evalusecase_DB "SELECT ?Computer ?Disk WHERE {?Computer cst:hasDisk ?Disk}"
+-------------------------------+------------------+
|            Computer           |       Disk       |
+-------------------------------+------------------+
| usecase:C_BTY_HQ_Computer     | usecase:HQ_Disk  |
| usecase:C_BTY_2TP_Computer    | usecase:2TP_Disk |
| usecase:C_BTY_3TP_Computer    | usecase:3TP_Disk |
| usecase:C_BTY_1TP_Computer_Nic | usecase:1TP_Disk |
+-------------------------------+------------------+
```

18. Which disks have Data?

```
./stardog query -r Evalusecase_DB "SELECT ?Computer ?Disk ?DataInfo ?Data WHERE {?Computer cst:hasDisk ?Disk . ?Disk cst:hasData ?Data . ?
Data cst:dataType ?DataInfo}"
+-------------------------------+------------------+-----------------------------------------------------------+------------------+
|            Computer           |       Disk       |                          DataInfo                         |       Data       |
+-------------------------------+------------------+-----------------------------------------------------------+------------------+
| usecase:C_BTY_HQ_Computer     | usecase:HQ_Disk  | "Battle Plans"                                            | usecase:HQ_Data  |
| usecase:C_BTY_1TP_Computer_Nic | usecase:1TP_Disk | "Ammunition State"                                        | usecase:1TP_Data |
| usecase:C_BTY_2TP_Computer    | usecase:2TP_Disk | "Music: Fortunate Son - Creedence Clearwater Revivial" | usecase:3TP_Data |
| usecase:C_BTY_3TP_Computer    | usecase:3TP_Disk | "Music: Fortunate Son - Creedence Clearwater Revivial" | usecase:3TP_Data |
+-------------------------------+------------------+-----------------------------------------------------------+------------------+
```

19. Which disks are encrypted

```
./stardog query -r Evalusecase_DB "SELECT ?Computer ?Disk ?Encryption WHERE {?Computer cst:hasDisk ?Disk . ?Disk cst:isEncryptedWith ?
Encryption}"
+---------------------------+-----------------+-------------+
|         Computer          |      Disk       | Encryption  |
+---------------------------+-----------------+-------------+
| usecase:C_BTY_HQ_Computer | usecase:HQ_Disk | usecase:AES |
+---------------------------+-----------------+-------------+
```

20. Who owns piece of data X?

```
./stardog query -r Evalusecase_DB "SELECT ?Computer ?Disk ?Owner WHERE {?Computer cst:hasDisk ?Disk . ?Disk cst:hasData ?Data . ?Data
cst:hasOwner ?Owner}"
+-------------------------------+------------------+--------------------------+
|            Computer           |       Disk       |          Owner           |
+-------------------------------+------------------+--------------------------+
| usecase:C_BTY_HQ_Computer     | usecase:HQ_Disk  | usecase:BCChraliebattery |
| usecase:C_BTY_1TP_Computer_Nic | usecase:1TP_Disk | usecase:1Troopcommander  |
| usecase:C_BTY_2TP_Computer    | usecase:2TP_Disk | usecase:3Troopcommander  |
| usecase:C_BTY_3TP_Computer    | usecase:3TP_Disk | usecase:3Troopcommander  |
+-------------------------------+------------------+--------------------------+
```

21. Which networks are wirelessly accessibele?

```
./stardog query -r Evalusecase_DB "SELECT ?WAP ?SSID {?WAP rdf:type cst:WirelessAccessPoint . ?WAP cst:ssid ?SSID}"
+----------------+------------+
|      WAP       |    SSID    |
+----------------+------------+
| usecase:JFTWAP | "JFT_WIFI" |
+----------------+------------+
```

22. Does a portscan return both wired and wireless connections together?

```
./stardog query -r Evalusecase_DB "SELECT ?Computer WHERE {?Computer rdf:type cst:Computer}"
+-----------------------------------+
|              Computer             |
+-----------------------------------+
| usecase:JointFiresRouter          |
| usecase:JointFiresDomain          |
| usecase:JointFiresDomainController |
| usecase:JFT_Computer              |
| usecase:JFCC_Computer             |
| usecase:C_BTY_HQ_Computer         |
| usecase:C_BTY_1TP_Computer        |
| usecase:C_BTY_2TP_Computer        |
| usecase:C_BTY_3TP_Computer        |
| usecase:JFT_Laptop                |
| usecase:C_BTY_1TP_Computer_Nic    |
| usecase:AttackerVirtualServer     |
+-----------------------------------+
```

23. What is the wireless security protocol that is being used to protect a given network?

```
./stardog query -r Evalusecase_DB "SELECT ?WAP ?SSID ?SecurityProtocol {?WAP rdf:type cst:WirelessAccessPoint . ?WAP cst:ssid ?SSID . ?WAP
cst:hasWirelessSecurityProtocol ?SecurityProtocol}"
+----------------+------------+------------------+
|      WAP       |    SSID    | SecurityProtocol |
+----------------+------------+------------------+
| usecase:JFTWAP | "JFT_WIFI" | usecase:WEP      |
+----------------+------------+------------------+
```

24. Which nodes on the network are virtual machines?

```
./stardog query -r Evalusecase_DB "SELECT ?VM WHERE {?VM rdf:type cst:VirtualMachine}"
+----------------------+
|          VM          |
+----------------------+
| usecase:Attacker1_VM |
| usecase:Attacker2_VM |
+----------------------+
```

25. What are all the active VPN Connections?

```
./stardog query -r Evalusecase_DB "SELECT ?Computer ?Network WHERE {?Computer cst:hasNic ?VirtualNic . ?VirtualNic
cst:hasVirtualPrivateNetworkConnection ?Network}"
+---------------------------+-----------------+
|         Computer          |     Network     |
+---------------------------+-----------------+
| usecase:C_BTY_HQ_Computer | usecase:JFCCNET |
+---------------------------+-----------------+
```

26. Can a virtual machine be seen as part of a physical network?

```
./stardog query -r Evalusecase_DB "SELECT ?VM ?VMNET ?Computer ?Network WHERE {?VM cst:hasVirtualNic ?VNIC . ?VNIC
cst:isVirtuallyConnectedTo ?VMNET . ?Computer cst:hasNic ?VNIC2 . ?VNIC2 cst:isVirtuallyConnectedTo ?VMNET . ?Computer cst:hasNic ?NIC . ?
NIC cst:isConnectedTo ?Network}"
+---------------------+-------------------------------+-------------------------------+---------------------+
|          VM         |             VMNET             |           Computer            |       Network       |
+---------------------+-------------------------------+-------------------------------+---------------------+
| usecase:Attacker2_VM | usecase:AttackerVirtualNetwork | usecase:AttackerVirtualServer | usecase:AttackerNET |
| usecase:Attacker1_VM | usecase:AttackerVirtualNetwork | usecase:AttackerVirtualServer | usecase:AttackerNET |
+---------------------+-------------------------------+-------------------------------+---------------------+
```

27. Which exploit maps to which vulnerability?

```
./stardog query -r Evalusecase_DB "SELECT ?Computer ?Vulnerability ?Exploit WHERE {?Computer cst:hasInstalledSoftwareVersion ?Software . ?
Software cst:hasVulnerability ?Vulnerability . ?Vulnerability rdf:type ?SoftwareVulnerability . ?ExploitTarget ceso:portkey_tso-
cst_ExploitTarget-SoftwareVulnerability ?Vulnerability . ?Exploit tso:targets ?ExploitTarget}"
+---------------------------+-------------------+---------------------------+
|         Computer          |   Vulnerability   |          Exploit          |
+---------------------------+-------------------+---------------------------+
| usecase:JFCC_Computer     | usecase:Heartbleed | usecase:HeartbleedExploit |
| usecase:C_BTY_HQ_Computer | usecase:Heartbleed | usecase:HeartbleedExploit |
+---------------------------+-------------------+---------------------------+
```

28. Who are the threat actors involved in an incident?

```
./stardog query -r Evalusecase_DB "SELECT ?Incident ?ThreatActor WHERE {?Incident rdf:type event:Incident . ?Incident ceso:portkey_event-
tso_Incident-Exploit ?Exploit . ?ThreatActor tso:usesExploit ?Exploit}"
+-----------------------------------+-------------------+
|             Incident              |    ThreatActor    |
+-----------------------------------+-------------------+
| usecase:HeartbleedExploitedIncident | usecase:Attacker1 |
| usecase:HeartbleedExploitedIncident | usecase:Attacker2 |
+-----------------------------------+-------------------+
```

29. Which organisation do the threats belong to?

```
./stardog query -r Evalusecase_DB "SELECT ?ThreatActor ?ThreatOrganisation WHERE {?ThreatActor ceso:portkey_tso-org_ThreatActor-
ThreatOrganisation ?ThreatOrganisation}"
+-------------------+--------------------------------+
|    ThreatActor    |       ThreatOrganisation       |
+-------------------+--------------------------------+
| usecase:Attacker1 | usecase:3rdCyberWarfareSection |
| usecase:Attacker2 | usecase:3rdCyberWarfareSection |
+-------------------+--------------------------------+
```

30. What course of action is available to address an exploit?

```
./stardog query -r Evalusecase_DB "SELECT ?Exploit ?CourseOfAction WHERE {?Exploit tso:preventativeCourseOfAction ?CourseOfAction}"
+--------------------------+----------------------+
|          Exploit         |    CourseOfAction    |
+--------------------------+----------------------+
| usecase:HeartbleedExploit | usecase:UpdateOpenSSL |
+--------------------------+----------------------+
```

31. What Effect has an Attack had?

```
./stardog query -r Evalusecase_DB "SELECT ?Exploit ?CyberEffect WHERE {?Exploit ceso:portkey_tso-ceso_Exploit-CyberEffect ?CyberEffect}"
+--------------------------+---------------------+
|          Exploit         |     CyberEffect     |
+--------------------------+---------------------+
| usecase:HeartbleedExploit | ceso:InterceptEffect |
+--------------------------+---------------------+
```