A common core-based cyber ontology in support of cross-domain situational awareness

Brian Donohue, Mark Jensen, Alexander Cox, Ron Rudnicki

Brian Donohue, Mark Jensen, Alexander P. Cox, Ron Rudnicki, "A common core-based cyber ontology in support of cross-domain situational awareness," Proc. SPIE 10635, Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IX, 106350F (4 May 2018); doi: 10.1117/12.2307719

Event: SPIE Defense + Security, 2018, Orlando, FL, United States

A Common Core-Based Cyber Ontology in Support of Cross-Domain Situational Awareness

Brian Donohue*[a], Mark Jensen[a], Alexander P. Cox[a], Ronald Rudnicki[a]
[a]CUBRC, Inc., 4455 Genesee Street, Buffalo, NY, USA 14225

ABSTRACT

Awareness of mission environment and events is impeded by data heterogeneity and lack of integration among data sources across diverse domains. In this paper, we present C3O, an RDF/OWL-based cyber ontology which provides a representation of cyber assets and events, to which existing XML-based cyber models (STIX, CybOX) can be mapped. C3O is unique in that it is designed as an extension of Basic Formal Ontology (BFO) and the Common Core Ontologies (CCO), which renders it automatically interoperable with a host of existing BFO- and CCO-based domain ontologies for land, sea, air, planning, operations, and sensor data.

Keywords: Cyber security, situational awareness, ontology, Basic Formal Ontology

1. INTRODUCTION

Cyber assets play increasingly critical roles in mission environments, including mission management, data mining, and decision support. Thus, mission success is predicated on the ability to protect cyber assets through anticipating, preventing, and counteracting cyber threat activity, and to react efficiently in the face of compromised cyber assets. This requires, in turn, that cyber data be ingested and exploited in a way which captures not only the characteristics of a threat to some cyber asset, but also the relationship of those threats to a mission whose success depends upon the health and security of that asset.

This paper presents the Common Core Cyber Ontology (C3O), an ontology for the cyber domain which is designed to align cyber data with data from other domains, and which can thus be used to support multi-domain situational awareness. In Section 2, we describe ontologies and our approach to ontology design. In Section 3, we discuss related work in cyber taxonomies and ontology development. In Section 4, we discuss the higher-level semantic framework on which C3O is based (namely, Basic Formal Ontology and Common Core Ontologies). In Section 5, we provide an overview of some of the main content of C3O. In Section 6, we present two use cases. In Section 7, we discuss future work.

2. ONTOLOGIES AND MODULAR DEVELOPMENT

An ontology is a formal representation of categories of entities and their relationships, which can be implemented in a machine-processable language. For example, an ontology of mammals would be designed to capture a hierarchy of mammal classes, additional facts about those classes (called class axioms), individuals that belong to those classes, and inferences about those individuals. For example:

Assertions
Canis lupus familiaris subClassOf Mammal
Every Mammal hasPart some Neocortex
Fido instanceOf *Canis lupus familiaris*

Inferences
Fido instanceOf Mammal
Fido hasPart Neocortex_1

Because ontologies are focused on representing the relationships between entities, they are typically designed to structure data in directed graphs. Class axioms can then be understood as rules for inferring extensions of the graph. For example, the assertions above can be captured in the following graph structure:

Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IX,
edited by Michael A. Kolodny, Dietrich M. Wiegmann, Tien Pham, Proc. of SPIE Vol. 10635,
106350F · © 2018 SPIE · CCC code: 0277-786X/18/$18 · doi: 10.1117/12.2307719

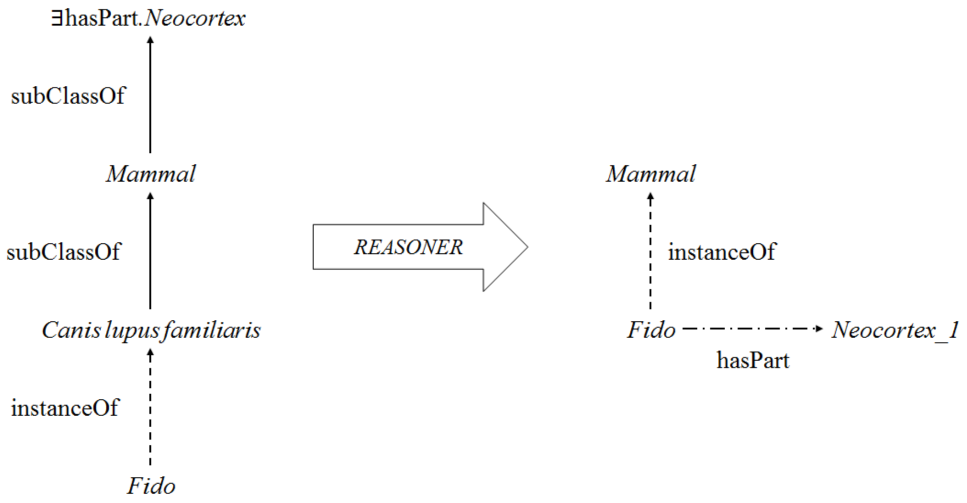

Figure 1. An example of reasoning with ontologies

Ontologies have been proposed in a wide array of domains as a way to enhance data expressivity, clarify ambiguities in data, promote data reusability, discover relationships within or across datasets, and fuse heterogeneous information sources. Accordingly, ontologies are often designed to provide a unified representation of entities in a domain. In many cases, however, there is also need for a unified representation of entities from different domains. Thus, some ontologies have been designed to specify a set of generic, domain-neutral categories and relationships, which can serve as a common upper-level semantic framework for multiple domain ontologies [5, 11, 1]. Specifically, we can distinguish three types of ontologies based on their levels of granularity:

- A *top-level ontology* is an ontology that represents only highly generic categories of entity (e.g., object, quality, function, process) and their relationships to each other (e.g., has_part).
- A *mid-level ontology* is an ontology that represents relatively general categories common to many domains of interest (e.g., person, act of communication, country). Such categories often require further specialization to be useful for data modeling.
- A *domain-level ontology* is an ontology that represents categories that are of interest to a more limited number of domains (e.g., intelligence analyst role, portion of ammonium nitrate, or watercraft registration).

C3O is designed to be a domain-level ontology whose categories and relationships extend from the more generic categories defined in top- and mid-level ontologies. Such a top-down approach enables this cyber ontology to reuse content of relevance across domains (e.g., computer, communication function, algorithm, telecommunication network, and vulnerability), thus promoting interoperability between data about cyber threats, mission status, environment, etc. This approach also enables our ontology to define categories and relationships for the cyber domain according to a generic semantic framework with proven success in data integration.t

3. RELATED WORK

A typical cyber ontology aims to represent some of the following categories and relationships:

- *Network devices* (together with their components, characteristics, and functionality): computers, hardware components and characteristics, data storage and directories, software systems and applications, software execution, asset performance, network requests.
- *Network infrastructure:* servers, clients, addresses, ports, nodes, endpoints, switches, routers.
- *Network traffic:* network connections and protocols, connection states, packet transfer, network performance, network permissions.

- *Threat actor behavior and methods:* threat indicators, cyber attack types, effects, and severity, adversary capabilities.

Some cyber ontologies, however, are decidedly narrower in scope, e.g., focusing exclusively on the areas of cloud security [19, 6] or network attack planning [20]. Many early cyber ontology efforts either lacked implementation in any semantic markup language [14, 8], or were implemented in now defunct formats [12], such as DAML+OIL.[1] More recent ontologies are typically encoded in OWL 2 (DL)[2] for use with associated semantic technologies (e.g., HermiT[3] and SPARQL[4]).

Some cyber ontologies have been designed primarily to integrate cyber data found in heterogeneous sources and formats. The Structured Threat Information eXpression (STIX) [3] represents the most successful and widely adopted schema for standardizing cyber threat data. Although initially published in XML, STIX has more recently been implemented in OWL.[5] Structurally, STIX consists of a nine core classes (e.g., observable, indicator, threat actor) accompanied by a set of ancillary models for enumerating cyber threat types and exchanging cyber threat information. Thus, STIX provides both a common schema for representing cyber attack scenarios and a database of threat types. In a similar vein, Iannacone, et al., [7] developed a cyber ontology for the STUCCO project, which can be used to map a variety of structured and unstructured cyber data sources. However, neither the STIX schema nor the STUCCO ontology provide a way to link data about cyber threats with data from other domains.

By contrast, the Unified Cyber Ontology (UCO) [18] is designed not only to integrate heterogeneous cyber datasets by means of a unified cyber data standard, but also to link cyber data and other domains by means of mapping UCO terms to Semantic Web resources (e.g., Linked Open Data,[6] DBPedia [2], Yago Knowledge Base [17]). However, this approach to multi-domain integration falls short insofar as it maps its cyber content to a grab bag of Semantic Web resources which vary widely in their accuracy and logical consistency. For example, Yago extracts terms from Wikipedia pages and automatically annotates them with ontology terms, but this process yields a database which is riddled with false assertions, for example:

> An elephants' graveyard is a mammal.[7]
>
> A Lockheed Martin A2100 is a person.[8]
>
> The Berneuse Mountain is a cable car.[9]

Therefore, although UCO correlates cyber data with data from other domains, the resultant fused data source fails to provide reliable information.

CRATELO [10] is designed to be a domain-level cyber extension of the top-level ontology DOLCE [5] and the mid-level security ontology SECCO [13]. However, CRATELO is also limited in a few significant ways. Because CRATELO takes SECCO as its mid-level ontology, it does not provide a way to link cyber data with general concepts relevant from outside the security domain, for example, enemy, computer, and country. Thus, it provides only limited interoperability between cyber data sources and other data.

4. BASIC FORMAL ONTOLOGY AND COMMON CORE ONTOLOGIES

By contrast, C3O is built as a domain-level extension of existent top- and mid-level ontologies, which allows cyber data to be seamlessly integrated with mission data, weather data, and so forth. First, C3O extends from the top-level Basic Formal Ontology (BFO) [1], which has been widely used in developing interoperable ontologies across domains,

[1] https://www.w3.org/TR/daml+oil-reference/
[2] https://www.w3.org/TR/owl2-overview/
[3] http://www.hermit-reasoner.com/
[4] https://www.w3.org/TR/sparql11-overview/
[5] https://github.com/daedafusion/cyber-ontology/
[6] http://linkeddata.org/
[7] https://gate.d5.mpi-inf.mpg.de/webyagospotlx/Browser?entity=%3CElephants%27_graveyard%3E
[8] https://gate.d5.mpi-inf.mpg.de/webyagospotlx/Browser?entity=%3CA2100%3E
[9] https://gate.d5.mpi-inf.mpg.de/webyagospotlx/Browser?entity=%3CBerneuse%3E

especially within biomedical information systems. BFO provides a generic, domain-neutral classificatory framework for integrating lower-level ontologies. Some important classes (with examples) are listed below:

- Object (e.g., a person, a vehicle, a building)
- Process (e.g., a course of action, a querying process, a cyber attack)
- Quality (e.g., the temperature or weight of a hardware component)
- Function (e.g., a sensor’s gunshot detection capability)
- Temporal region (e.g., the duration of a mission, the datetime of a message)
- Spatial region (e.g., the location of a target)

entity

- *continuant*
  - *independent continuant*
    - *material entity*
      - *object*
      - *object aggregate*
      - *fiat part*
    - *immaterial entity*
      - *spatial region*
      - *site*
      - *fiat boundary*
  - *specifically dependent continuant*
    - *quality*
    - *realizable entity*
      - *disposition*
      - *role*
  - *generically dependent continuant*
- *occurrent*
  - *process*
  - *process boundary*
  - *temporal region*
  - *spatiotemporal region*

Figure 2. A pared down overview of the BFO class hierarchy.

C3O also extends from the Common Core Ontologies (CCO), which are designed as a mid-level extension of BFO. Thus, CCO defines classes such as agent, criminal act, and year as subclasses of BFO’s object, process, and temporal region, respectively. Of particular relevance to the cyber domain are CCO classes for representing information artifacts and how they relate to actors, times, regions, artifact specifications, plans, and vulnerabilities. Therefore, before presenting C3O as a semantic representation of the cyber domain, we will describe some of the content already defined in BFO/CCO on which C3O is based.

First, CCO draws a distinction between the content of some information source (data), and the expressions of that content in a physical medium (a data source) [16]. For example, the content of a document should be distinguished from two physical or digital copies of that content. CCO draws a further distinction between information content and what that information is about in the world. For example, a record in a patient database is obviously not the same entity as the patient, but rather a record of information that is about that patient [15]. Thus, CCO defines the following two classes, together with two properties for capturing the relationships between them:

> *Information content entity* =def. An entity which generically depends upon an information bearer and is_about some entity.

Information bearing entity =def. An object which is bearer_of an information content entity.

Thus, a record in a database can be represented as an information content entity that is_about something (i.e., the patient), where there is some information bearing entity (i.e., the database) which is the bearer_of that information. This distinction between the information itself and the particular material bearers which store that information is important because it provides an explicit way of representing: (1) the provenance of a piece of information, (2) auditing history (e.g., who edited a computer file), and (3) permissions (e.g., permission to read that file).

CCO also defines several subtypes of information content entity of immediate relevance to the cyber domain, especially:

Algorithm =def. An information content entity that prescribes the inputs and outputs of mathematical functions as well as workflow of execution for achieving a predefined objective.

Thus, for the purposes of the cyber domain, an algorithm will be understood as the piece of information that inheres in a particular computer program or subroutine (method, function), and that prescribes the steps that need to be executed. It is the blueprint specifying what the program (virus, malware, etc.) should accomplish. Note, again, that an algorithm is distinct from any particular copy of a program (e.g., the program on one user's hard drive vs. another user's hard drive). Copies of a program can be created, modified, deleted, uploaded, and downloaded, but they share a common blueprint.

Next, CCO also provides a general way to represent artifacts, their design specifications, actual performance, maintenance, manufacturing lifecycle, vulnerabilities, and the roles they play in different contexts. Specifically, CCO defines a parent class artifact along with several cyber-relevant subclasses:

Artifact =def. An object that is designed by some agent to realize a specified function.

Digital storage device =def. An artifact is designed to bear some information content entity by means of recording that information in digital (binary) format.

Information bearing artifact =def. An artifact that is designed to bear an information content entity.

Information processing artifact =def. An artifact that is designed to bear an algorithm which prescribes how to transform an input information content entity into an output information content entity.

Computer =def. An information processing artifact is designed to execute an arbitrary set of arithmetic or logical operations automatically.

5. THE COMMON CORE-BASED CYBER ONTOLOGY (C3O)

C3O is designed as an extension of BFO and CCO with the aim of providing a representation of the cyber domain which automatically achieves cross-domain interoperability. Accordingly, categories of network devices, infrastructure, and traffic, along with categories of threat actor behavior and methods, will be represented as domain-specific extensions of CCO categories for computer artifacts, information processing, plans, procedures, and telecommunications. Below, we describe a sample of its content, focusing on the following central categories: digital file, computer program, computer network, information transfer process, and cyber attack.

The class digital file is designed to capture the digital encoding of a piece of information content as found on a digital storage device and which bears an information content entity. Thus, the class digital file represents not the information found on a file (e.g., the contents of a message), but rather the material bearer which encodes that information.[10] In other words, digital file is a type of information bearing artifact:

Digital file =def. An information bearing artifact that is located in a digital storage device.

[10] The relationship between a *digital file* and a *digital storage device* is that of **located in** as defined in the Relation Ontology (RO). Primarily, **located in** is defined as a relation holding between a bfo:continuant and the bfo:spatial_region which it exactly occupies. However, the scope of **located in** can be expanded such that for any continuant *c1*, which occupies a spatial region *s1*, where *s1* is wholly contained within another spatial region *s2*, which is occupied by some further continuant *c2*, *c1* is **located in** *c2*. For example, the space occupied by the skull wholly contains the space occupied by the brain. Therefore, we can say not only that the brain is located in some spatial region, but also that the brain is located in the skull.

In some cases, the data contained in a digital file is merely descriptive. For example, if the content of the digital file "report.doc" might just be a description of an event. In other cases, a digital file contains scripts, methods, or functions, which can be interpreted, compiled, or executed by devices. In such cases, C3O asserts that a digital file has_part some subroutine. However, when there is an aggregate of such digital files, designed in a coordinated fashion to execute a set of processes specified by an algorithm, there is a computer program. Thus:

> *Computer program* =def. An information processing artifact that consists of a collection of subroutines that is designed to be executed by a processing unit to perform a specific task.
>
> *Subroutine* =def. An information processing artifact that is designed to bear an algorithm and to be executed within the context of some computer program.

For a *computer program* or *subroutine* is executed when it **is input of** some process, the **agent of** which is some artifact which can realize the process prescribed by the corresponding *algorithm* (e.g., a CPU).

Moreover, C3O provides a representation of the formatting and language of files and programs. Thus, one can assert that a digital file uses_format some file format (e.g., .pdf, .xsl, .java), and that computer program uses_language some computer language (e.g., Python, Java, C#).

Digital files and computer programs reside on digital storage devices, which are parts of computers or other devices. Those devices, in turn, establish connections to form networks, which C3O captures by defining the class computer network, various computer network functions and roles (e.g., computer has_role some network server role), and networking devices (computer network has_part some network hub). Thus:

> *Computer network* =def. A telecommunication network that is designed to allow the exchange of data between two or more computers connected to the network.

These computer networks, in turn, facilitate information transfer processes, in which pieces of information are passed from one bearer to another (e.g., via packet-switching):

> *Information transfer process* =def. A process in which an information content entity inhering in one information bearing entity is sent to another information bearing entity, which then bears that information content entity.

Figure 3 summarizes these relationships. (Note that 'ICE' is an abbreviation of 'InformationContentEntity', and that the dashed lines represent the result of the InformationTransferProcess.)

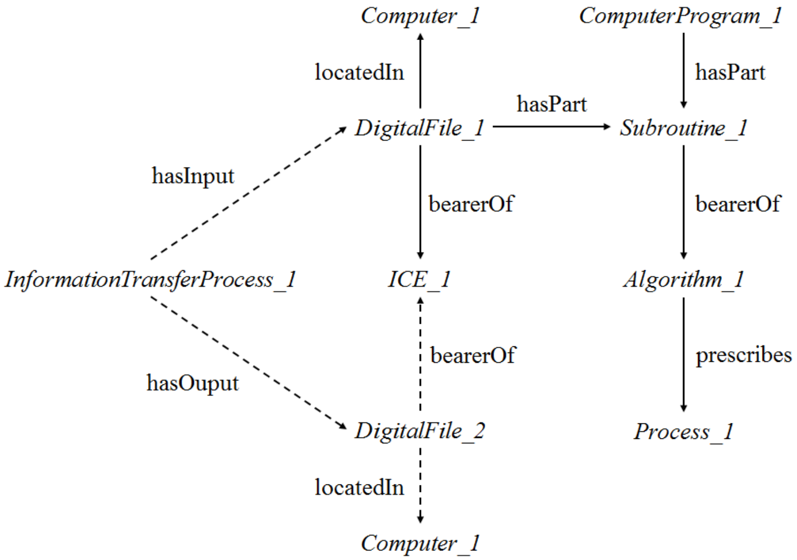

Figure 3. Summary of relationships between information, bearers, and subroutines.

6. USE CASES

6.1 SYN Flood Indicators

A denial of service (DoS) attack aims to render a network server inactive, and thus prevent client devices from accessing network services. A SYN flood is a form of DoS which attempts to cripple a server by exploiting a vulnerability in the process of requesting a server connection via Transmission Control Protocol (TCP). Normal successful TCP connections consist of three steps (the "three-way handshake"):

1. SYN: A client sends a request to synchronize with a server.
2. SYN-ACK: The server sends an acknowledgement in response to the client, combined with a like request to synchronize.
3. ACK: The client responds to the server's request with an acknowledgement.

In a SYN flood, the client submits a SYN request to the server, which responds with a SYN-ACK message, but then the client neglects to send the corresponding ACK message to establish the connection. Thus, the server will wait for an ACK message until the request times out. Alternatively, the client might falsify its IP address, so that the server will attempt to establish a connection with a client that does not exist. In either case, the server's resources are diverted to false connections. When a multiple such false SYN requests are submitted, the server's resources can become entirely exhausted, such that it can no longer provide services to legitimate clients.

In our first use case, we modeled a Wireshark packet-capture log obtained from IMPACT Cyber Trust,[11] which we then queried for indicators of a SYN flood attack on the network server. The goal of this use case is simply to demonstrate how C3O would be used to represent an ordinary cyber threat scenario. For the purposes of this paper, we present an example based on that data (actual IPs have been modified), in which a client with the IP 117.19.142.130 is successively requesting TCP connections with a network server with IP 162.20.30.221. The client is making several such requests, but not sending the ACK message to complete the TCP connection. Therefore, we have some indicators that this client is attempting or participating in a SYN flood attack on the server. Figure 4 captures (in a slightly compressed graph) these entities and relationships.[12]

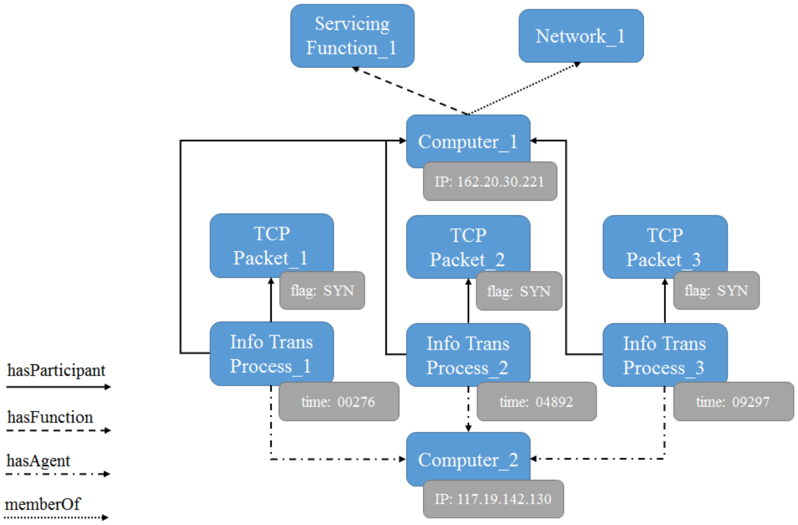

Figure 4. A SYN Flood targeting a network server.

[11] https://www.impactcybertrust.org/

[12] Note that in the following diagrams, blue nodes represent the relevant individuals. Associated literal values are represented here in the form of key-value pairs embedded in gray nodes. This is done purely in order to simplify the presentation of the graph. In the full CCO-based graph, IP addresses, times, and flags would be represented as individuals (i.e., blue nodes).

Thus, the following example SPARQL query could be issued against this graph as part of a test for whether there are indicators of a SYN flood attack in progress:

```
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>
PREFIX ex: <http://www.example.com/UseCases/>
SELECT ?packet WHERE {
    ?synreq cco:has_participant ?packet ;
      cco:has_participant ex:myComputer ;
      cco:has_agent ?client .
    ?packet_1 a cco:TCPPacket ;
      cco:bearer_of cco:SYNFlag .
    MINUS {
          ?ackreq cco:has_agent ?client ;
            cco:has_participant ?packet_2 ;
            cco:has_participant ex:myComputer .
          ?packet_2 a cco:TCPPacket ;
      cco:bearer_of cco:ACKFlag .
      }
}
```

Note that this query could easily be amended to return results within a designated range of times.

6.2 Compromised Mission Server in Planning and Operations

In our second use case, we represent a server affected by a SYN flood DoS attack, which is a critical asset in a military operation. The goal of this use case is demonstrate how a C3O-based representation of a cyber threat scenario is automatically interoperable with existent CCO-based mid-level mission ontologies, such as the Joint Doctrine Ontology [9] and the Space Object Ontology [4]. This use case links information about the cyber threat, which occurs during the mission, with information about which cyber assets are being employed by mission forces. Specifically, a network server which is being employed within an operation has been rendered inoperative by a SYN flood. See Figure 5.

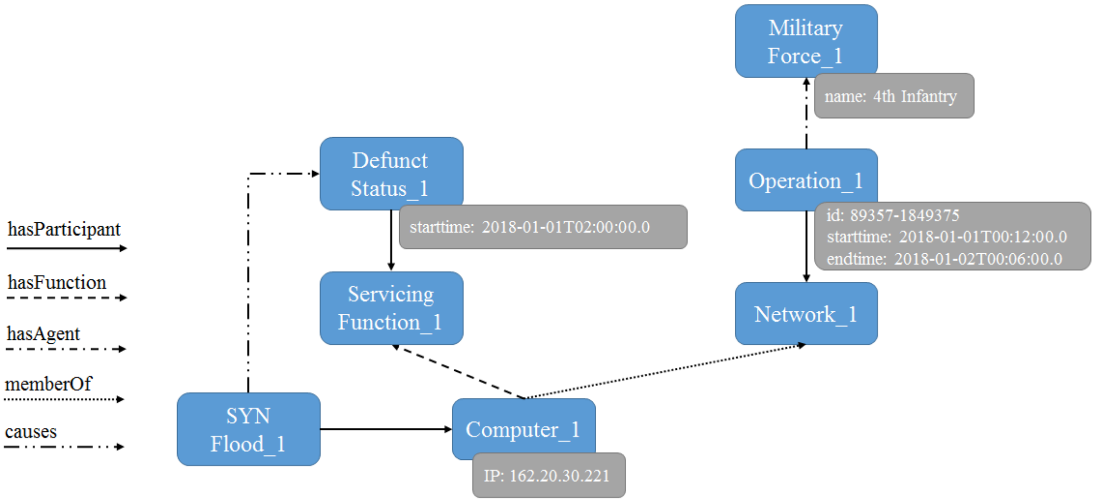

Figure 5. A SYN Flood affects a mission asset.

Thus, the following example SPARQL query could be issued against this graph as a way of discovering which missions are affected by the cyber attack:

```
PREFIX cco: <http://www.ontologyrepository.com/CommonCoreOntologies/>
SELECT ?op WHERE {
      ?op a cco:MilitaryOperation ;
```

```
                    cco:has_participant ?network .
            ?network a cco:ComputerNetwork ;
                    ?comp a cco:Computer ;
                    cco:member_of ?network ;
                    cco:has_function ?servicing .
            ?servicing a cco:ServicingFunction .
            ?defunct a cco:DefunctStatus ;
                    cco:has_participant ?servicing .
            ?synflood a cco:SYNFlood ;
                    cco:causes ?defunct ;
                    cco:has_participant ?comp .
    }
```

As in the first use case, the query could be restricted to return results within a designated timeframe.

7. FUTURE WORK

In the future, we hope to provide a full mapping of existing cybersecurity standards, especially the widely used MITRE cybersecurity models (STIX) and its associated threat and vulnerability vocabularies and enumerations (CVE, CWSS). Providing such a mapping would enable existing vendors and community users of such standards not only to exchange cyber threat information, but also to link cyber threat data with, in principle, any other domain of interest. In this vein, we also hope to demonstrate further interoperability with other existing CCO-based domain ontologies, especially those for land, sea, air, and space missions. Lastly, we plan to extend C3O to represent additional areas of the cyber domain, especially additional types of malware and asset vulnerabilities, as well as to provide a fuller treatment of cyber threat types, alert types, priority, and severity, threat stages, courses of action, and "kill chains".

In conclusion, we see C3O as an important first step in enhancing situational awareness in multi-domain scenarios. In approaching the development of a cyber ontology from the top-down perspective of global data interoperability, we aim to provide a robust, extensible framework for establishing semantic connections between cyber data sources and across domains of relevance to cyber security events.

ACKNOWLDGEMENTS

This research was partially supported by U.S. Air Force Research Laboratory (AFRL) under agreement AFRL FA9302-13-C-0236. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of AFRL or the U.S. Government.

REFERENCES

[1] Arp, R., Smith, B., Spear, A.D., [Building Ontologies with Basic Formal Ontology], MIT Press, Cambridge (2015).
[2] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z., "Dbpedia: A nucleus for a web of open data," The Semantic Web, 722-735 (2007).
[3] Barnum, S., "Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX)," MITRE, 2014, <http://stixproject.github.io/about/STIX_Whitepaper_v1.1.pdf> (8 Nov 2017).
[4] Cox, A.P., Nebelecky, C.K., Rudnicki, R., Tagliaferri, W.A., Crassidis, J.L., and Smith, B, "The Space Object Ontology," Proc. from the 19th International Conference on Information Fusion (FUSION), 146-153 (2016).
[5] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., & Schneider, L., "Sweetening ontologies with DOLCE," Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, 223-233 (2002).
[6] Hendre, A., & Joshi, K. P, "A semantic approach to cloud security and compliance," Proc. IEEE 8th International Conference on Cloud Computing (CLOUD), 1081-1084 (2015).
[7] Iannacone, M., Bohn, S., Nakamura, G., Gerth, J., Huffer, K., Bridges, R., & Goodall, J., "Developing an ontology for cyber security knowledge graphs," Proc. 10th Annual Cyber and Information Security Research Conference, 12 (2015).

[8] Massacci, F., Mylopoulos, J., Paci, F., Tun, T.T., & Yu, Y., "An extended ontology for security requirements," Proc. CAiSE Workshops (2011).

[9] Morosoff, P., Rudnicki, R., Bryant, J., Farrell, R., and Smith, B, "Joint Doctrine Ontology: A Benchmark for Military Information Systems Interoperability," Proc. STIDS 10th International Conference on Semantic Technology for Intelligence Defense and Security (STIDS) (2015).

[10] Oltramari, A., Cranor, L. F., Walls, R. J., & McDaniel, P. D., "Building an Ontology of Cyber Security," Proc. 9th International Conference on Semantic Technology for Intelligence Defense and Security (STIDS), 54-61 (2014).

[11] Pease, A., Niles, I., and Li, J., "The suggested upper merged ontology: A large ontology for the semantic web and its applications," Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web 28, 7-10 (2002).

[12] Pinkston, J., Undercoffer, J. Joshi, A., and Finin, T., "A target-centric ontology for intrusion detection," University of Maryland, Baltimore County Department of Computer Science and Electrical Engineering.

[13] Pirrò, G., Ruffolo, M., and Talia, D., "SECCO: on building semantic links in Peer-to-Peer networks," Journal on Data Semantics 12, 1-36 (2009).

[14] Simmonds, A., Sandilands, P., and Van Ekert, L., "An ontology for network security attacks," Asian Applied Computing Conference, 317-323 (2004).

[15] Smith, B. and Ceusters, W., "Aboutness," Proc. of the 6th International Conference on Biomedical Ontology (ICBO) (2015).

[16] Smith, B., Malyuta, T., Rudnicki, R., Mandrick, W., Salmen, D., Morosoff, P., Duff, D.K., Schoening, J., and Parent, K., "IAO-Intel: An Ontology of Information Artifacts in the Intelligence Domain," Proc. 8th International Conference on Semantic Technologies for Intelligence Defense and Security (STIDS), 33-40 (2013).

[17] Suchanek, F. M., Kasneci, G., & Weikum, G., "Yago: A large ontology from wikipedia and wordnet," Web Semantics: Science Services and Agents on the World Wide Web, 6(3), 203-217. (2008).

[18] Syed, Z., Padia, A., Finin, T., Mathews, M. L., & Joshi, A., "UCO: A Unified Cybersecurity Ontology, "Proc. AAAI Workshop: Artificial Intelligence for Cyber Security, February 2016 <http://www.aaai.org/ocs/index.php/WS/AAAIW16/paper/download/12574/12365> (8 Nov 2017).

[19] Takahashi, T., Kadobayashi, Y., and Fujiwara, H., "Ontological approach toward cybersecurity in cloud computing," Proc. of the 3rd International Conference on Security of Information and Networks, 100-109 (2010).

[20] van Heerden, R., Chan, P., Leenen, L., and Theron, J. "Using an ontology for network attack planning," International Journal of Cyber Warfare and Terrorism (IJCWT), 6(3), 65-78 (2016).