INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Development of an Integrated Ontology to Enhance Interoperability for Airports Security Solutions

KATIA ALEID
Outubro de 2020

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Development of an Integrated Ontology to Enhance Interoperability for Airports' Security Solutions

Katia Aleid

Dissertation to obtain the Master of Science degree in

Computer Science, Specialization in

Knowledge and Information Systems

Orientador: Isabel Praça

Co-orientador: Alda Canito

Júri:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, October 2020

Abstract

The importance of airports' cybersecurity is being given from the combination of two key components: airports and security. As airports provide transportation among other services to people which are the most valuable cargo, they are also of the biggest investments in any country. Hence, ensuring security of airports is a necessity in any possible aspect. In this context, we aim to overcome the difference in information representation formats used within airports. Thus, facilitating communication and knowledge exchange between different cybersecurity systems and solutions.

In this thesis, we are exploring the current state and activities of airports' cybersecurity as a base for the development process. Also, we discuss the design of an integrated ontology focused on airports' cybersecurity. After that, the resulting ontology will be checked against the chosen criteria to evaluate its usefulness in achieving the interoperability goal.

Keywords: Ontology, Cybersecurity, Airport, Interoperability, Integration

Resumo

A importância da cibersegurança nos aeroportos é conseguida através da junção de duas componentes chave, segurança e aeroporto. Sendo que os aeroportos oferecem transporte, entre outros serviços, para as pessoas, que são a carga mais valiosa, são também grandes investimentos em qualquer país. Portanto, assegurar a segurança no aeroporto é necessária em todos os aspetos possíveis.

Neste contexto, o nosso objetivo é superar a diferença na forma como a informação é usada em aeroportos. E deste modo, facilitar a comunicação e a troca de conhecimento entre os diferentes sistemas de cibersegurança e soluções. Nesta tese, vamos explorar o atual estado e atividades da cibersegurança nos aeroportos como base para o processo de desenvolvimento. Para além disso, vamos também discutir o design de uma ontologia integrada focada na cibersegurança nos aeroportos. Finalmente, a ontologia resultante será verificada contra os critérios escolhidos de forma a avaliar a sua utilidade em atingir o objetivo de interoperabilidade.

Palavras-chave: Ontologia, Cibersegurança, Aeroporto, Interoperabilidade, Integração

Acknowledgements

This work has received information from SATIE project, the European Union's Horizon 2020 research and innovation programme under grant agreement No 832969. This output reflects the views only of the authors, and the European Union cannot be held responsible for any use which may be made of the information contained therein.

The author was supported by the Global Platform for Syrian Students MSc scholarship 2018.

Parts of this work have been accepted as a paper in 2020 IEEE 6th International Conference on Computer and Communications ICCC), China. Paper is titled "An ontology to promote interoperability between cyber-physical security systems in critical infrastructures" with ID IC121.

Table of Contents

List of Figures

List of Tables

Acronyms and Nomenclature

List of Acronyms

ACM	Association for Computing Machinery
AIXM	Aeronautical Information Exchange Model
ANSSI	Agence Nationale de la Sécurité des Systèmes d'Information (national agency of information systems' security)
ASDL	Aviation Scenario Definition Language
ASIIO	Airport Security Interoperability Integrated Ontology
ATM	Air Traffic Management
ATMONTO	ATM Ontology
CAPEC	Common Attack Pattern Enumeration and Classification
CCM	Cloud Controls Matrix
CNCS	Centro Nacional de CiberSegurança (national center of cyber security)
CPE	Common Platform Enumeration
CRITIS	Critical Infrastructure Ontology
CSA	Cloud Security Alliance
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
CYBOS	Cyber Observable Expression
DDoS	Distributed Denial of Service
DoS	Denial of Service
EAV	Entity Attribute Value
ENISA	European Network and Information Security Agency
GML	Geography Markup Language
ICCC	International Conference on Computer and Communications
IDMEF	Intrusion Detection Message Exchange Format
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
INSPIRE	INcreasing Security and Protection through Infrastructure REsilience
IODEF	Incident Object Description Exchange Format
IRI	Internationalized Resource Identifier
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
MIOD	Methodology of Integration-oriented Ontology Development
NAS	National Airspace System
NASA	National Aeronautics and Space Administration
NCSS	National Cyber Security Strategy
NIST	National Institute of Standards and Technology

NVD	National Vulnerability Database
OD101	Ontology Development 101
ODE	Ontology Development Environment
OOPS!	OntOlogy Pitfall Scanner!
ORS	Ontology Requirements Specification
ORSD	Ontology Requirements Specification Document
OSCO	Ontologies of Secure Cyber Operations
OWL	Web Ontology Language
PAM	Privileged Access Management
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SAMOD	Simplified Agile Methodology of Ontology Development
SAO	Situation Awareness Ontology
SATIE	Security of Air Transport Infrastructure of Europe
SCAP	Security Content Automation Protocol
SECCO	Security Core Ontology
SIEM	Security Information and Events Management
SO	Security Ontology
SSLA	Security Service Level Agreement
STIX	Structured Threat Information eXpression
SWRL	Semantic Web Rule Language
TMI	Traffic Management Initiative
UCO	Unified Cybersecurity Ontology
URI	Uniform Resource Identifier
VLO	Vulnerability Lifecycle Ontology
VPC	Value Proposition Canvas
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

1 Introduction

Cybersecurity concerns are growing alongside the growth of online services. The more the community becomes dependent on the internet to communicate and conduct business, the more it becomes vulnerable to possible attacks and harmful usage. The efforts that are being made in the cybersecurity domain are still isolated and sparse. While there are many common features and concerns, different approaches use different ways to represent their knowledge. Which in turn makes it difficult to interact with other similar or complementary systems/processes and exchange information with them. This makes a big part of the work to be redundant instead of focusing on improvement towards achieving the higher goal.

Interoperability, by definition, is the capability to communicate and transfer information among several various functional sides. The problem is that each side uses its own data representation, organization, and semantics. This issue makes it harder to transfer data between applications. Therefore, the main goal is to form a clear comprehensive catalogue to facilitate data sharing and that is where ontologies became important.

This research aims to explore current work targeting the scope of cybersecurity related to airports, in order to obtain a solid base on what is available and what needs to be done. The goal is to develop an integrated ontology that serves as an integration and communication tool between airport cybersecurity systems and applications.

1.1 Problem Definition

In this modern age, where everything is connected to the internet, there are new threats associated to the new medium of communication. More and more services are provided online, which means more and more possible weak points to be exploited. According to (Risk Based Security 2015), just in the first half of 2015 more than 200 million records were exposed. A single hacking attack exposed about 78 million of those records. The number rose to more

than 15 billion records in 2019 making it the worst recorded year (Risk Based Security 2019). It is worth mentioning that this issue is a domain-crossing one. There is not a sector which uses technology that can be considered safe from such threats. Whether it is business, education, medical, or even governmental, they are all at risk if proper precautions are not taken.

As technology is continuously changing and developing, this makes the infrastructure unstable and vulnerable. However, this does not deny that humans play a role in this as well. Therefore, there is an interaction between human and machine elements which is very important when considering situation awareness in cybersecurity of systems.

Regardless of acting agents being humans or computers, any cybersecurity system needs to react as soon as possible to any change of state within its environment. To achieve that, it is necessary to collect and integrate information from different resources and systems. This information is needed to analyze events, make decisions, obtain feedback after applying those decisions, and gain knowledge to be used in future occurrences (Ulicny et al. 2014). The main challenge is that different systems use different representation of their knowledge. Therefore, an ontology that is focused on cybersecurity is needed to provide a standard way to exchange data between the corresponding systems.

1.2 Project Context

The Security of Air Transport Infrastructure of Europe[1] (SATIE) project aims to develop a cybersecurity toolkit to face cyber-physical threats in airports in a coordinated and effective way, supported by a shared situational awareness system.

Several systems of varying responsibilities contribute to this result by cooperating and exchanging security data. These systems' responsibilities include, but are not limited to:

- Cyber threat detection;
- Risk assessment;
- Simulation of impact propagation;
- Vulnerability management;
- Crisis alerting.

The cyber threat detection, air traffic monitoring services, passenger data, access control, traffic management, and baggage handling systems all work as data sources, providing real-time information about events occurring in different parts of the airport. To achieve the general goal of securing critical infrastructures, it is required from all involved systems to communicate with each other, such that a holistic view of the environment is possible. The main challenge facing the integration of these tools relies precisely on the communication problem.

[1] SATIE Website [Online]. Available: http://satie-h2020.eu. [Accessed: 20-Aug-2020]

An ontology focused on cybersecurity can provide a standard data-exchange between these systems, which would not only facilitate the existing communications but also the addition, removal, or compensation of systems from the overall architecture. For an ontology to be useful in this scenario, it would have to cover the knowledge representation needs of all the concerned parties, provide a unique frame of reference over the meaning of the exchanged messages which leaves no room for ambiguity, and guarantee that the same conclusions can be inferred in any part of the system. Such a semantic layer would facilitate a holistic, integrated view of the security status of the airport at any given moment, and the collaboration of multiple concerned parties would lead to an increase in the quality of the resulted work.

The main goal of the SATIE project is to construct a comprehensive, interoperable, and modular security toolkit that would be used by future Airport Operation Centre and Security Operation Centre to protect critical aviation infrastructures against possible threats. Ontologies will be the basis for the interoperability of these different tools. To assess the existence of ontologies useful for SATIE, related systems were analysed with regards to their inputs, outputs, and responsibilities. Afterwards, a high-level set of concepts was extracted, which was the starting point for this work.

1.3 Main Goals

The goal of this work is to enhance interoperability among different security systems. This interoperability requires the integration of knowledge from different sources represented in different formats and the definition and description of an application domain. Currently, it seems difficult for systems to exchange information due to this difference in representation. Any misinterpretation of data might lead to serious problems. Ontology is used to provide a semantic layer covering all the concerned parties, which would help with the translation process among the systems without any ambiguity.

1.4 Expected Results

The intended ontology is expected to make use of as much existing work as possible, while expanding and extending it even further to accommodate the domain's business requirements. The final ontology would represent a consensus definition of the major concepts related to airports' cybersecurity domain. By using ontological reasoning onto the monitored incidents, it would be expected to detect attack patterns and in turn help enhancing protection levels to avoid possible attacks.

1.5 Document Structure

This document will be structured as the following:

First chapter introduced the problem that is being studied and the goals that will be expected from this work. The context for this work will also be presented.

The second chapter will discuss ontologies and how they can be used to help achieve the interoperability goal. A general background to ontologies' principals, languages, and editing software will be presented. Ontology levels and development methodologies will also be discussed.

The third chapter will talk about cybersecurity and its related ontologies. Several cybersecurity ontologies will be presented along with their main concepts and features. Also in this chapter, some ontologies related to airports will be presented. A summary of studied ontologies will be available at the end of the chapter.

Fourth chapter will explore the value of this work for the prospective users. This will show the importance of this ontology from a business marketing point of view, and not just as a theoretical research.

The fifth chapter will focus on merging both domains to construct an integrated ontology that covers airports cybersecurity requirements. The design process will start with domain concepts survey and requirements assembly. Detailed evolution of the ontology throughout the design will be presented and discussed step by step. A summary of the developed ontology at the end of this chapter will be available to highlight the contribution towards the domain.

The sixth chapter will present the implementation of the ontology using the selected language and tool, as described in the second chapter.

The seventh chapter will present the research hypothesis that this work is trying to prove. In addition, the evaluation plan that includes the factors of interest and evaluation methodologies will be presented. The steps taken to validate the ontology will be discussed, along with the results and follow-up actions.

The final chapter will be the conclusion of this work and the discussion of future aspects.
This document will end with bibliographic references and appendices.

2 Introduction to Ontology

In this chapter, the description of ontologies is presented in addition to their part towards achieving interoperability between systems. Examples of ontology representation languages, editing software, levels, and development methodologies will also be provided.

2.1 Background

In philosophy, Ontology is the science of "structure of objects, properties, events, process and relations in every area of reality" (Smith 2003). The name comes from Greek *ont* meaning 'being' and the English study suffix 'logy', and the term Ontology was coined in 18th century[1]. Every field can create its own ontologies to organize data and prepare for information processing and knowledge extraction (Helle V. Dam, Jan Engberg et al. 2012). Figure 2.1 shows the increased usage of the term Ontology in books[2] between 1800 and 2018.

As for computer science, ontologies are designated to act as a communication intermediate between humans, human and computer, and several computers. By playing the expert's role, ontologies automate the translation process of unknown expressions that allows to transfer meaning between applications.

[1] Oxford Reference [Online]. Available: https://www.oxfordreference.com/view/10.1093/oi/authority.20110803100250688 [Accessed: 20-Aug-2020]

[2] Google Books Ngram Viewer [Online]. Available: https://books.google.com/ngrams/graph?content=ontology&case_insensitive=on&year_start=1800&year_end=2018&corpus=15&smoothing=5&share=&direct_url=t4%3B%2Contology%3B%2Cc0%3B%2Cs0%3B%3Bontology%3B%2Cc0%3B%3BOntology%3B%2Cc0%3B%3BONTOLOGY%3B%2Cc0#t4%3B%2Contology%3B%2Cc0%3B%2Cs0%3B%3Bontology%3B%2Cc0%3B%3BOntology%3B%2Cc0%3B%3BONTOLOGY%3B%2Cc0 [Accessed: 20-Aug-2020]

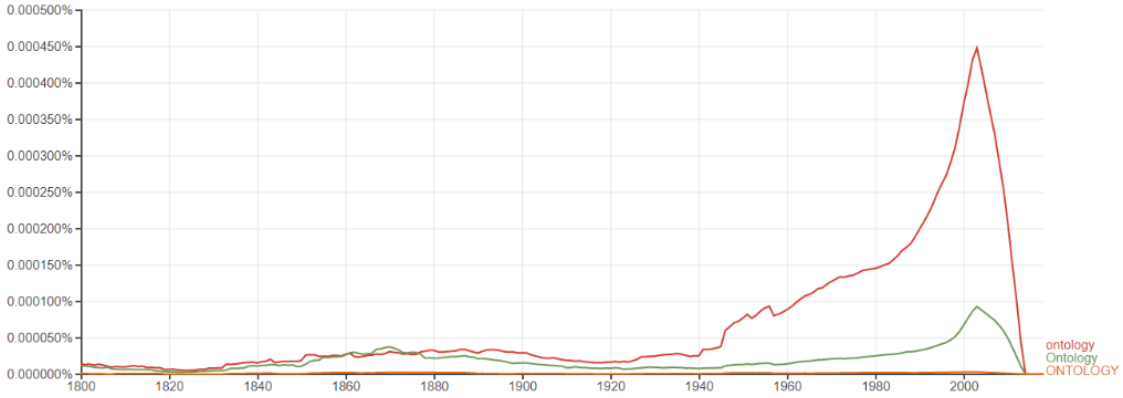

Figure 2.1 – Ontology term trendline[1]

The main part of building an ontology is creating a collective knowledge database that covers the hidden meanings and relations between domain related concepts. It is also possible, with the use of reasoners, to link familiar words' meaning to other unknown ones in order to enhance the original knowledge. Hence, the ontology must be (Michal et al. 2012):

- Highly dependable;
- Easy to implement and maintain;
- Able to reuse existing code;
- Adaptive to append additional knowledge.

2.2 Alternatives

In this section, methods that were used prior to ontologies will be introduced to give context to the progress towards ontologies and semantic languages.

The simplest format of representation used to store, and exchange information was raw text logs. Text logs are still being used till this day as logging and output format for most systems. Normally, the unconstrained structure of text files makes them easier to write enabling each system to compose its files the way it wants and understands. On the other hand, this makes text files to be not very useful nor easy to process by others. To overcome this issue, several standards and structured formats started to be used to give those text files some common structure. As an example of a highly used text semi-structure standard is SYSLOG protocol. The SYSLOG protocol aims to separate the content of textual messages into data elements that

[1] Google Books Ngram Viewer [Online]. Available: https://books.google.com/ngrams/graph?content=ontology&case_insensitive=on&year_start=1800&year_end=2018&corpus=15&smoothing=5&share=&direct_url=t4%3B%2Contology%3B%2Cc0%3B%2Cs0%3B%3Bontology%3B%2Cc0%3B%3BOntology%3B%2Cc0%3B%3BONTOLOGY%3B%2Cc0#t4%3B%2Contology%3B%2Cc0%3B%2Cs0%3B%3Bontology%3B%2Cc0%3B%3BOntology%3B%2Cc0%3B%3BONTOLOGY%3B%2Cc0 [Accessed: 20-Aug-2020]

can be transmitted easily. At the same time, this would enable easy extensibility of messages. The message structure as per documentation (Gerhards et al. 2009) is as follows:

```
PRI VERSION ISOTIMESTAMP HOSTNAME APPLICATION PID MESSAGEID [STRUCTURED-
DATA] MSG
```

Despite the structured approach to text files composition, they are still hard to use for knowledge extraction. Not to mention the difficulty in verifying the compliance to the standard or detection of mistakes and contradictions. The use of eXtensible Markup Language (XML) provided a clearer structure and hierarchy to textual content with useful validation and parsing capabilities. Several airport and cybersecurity domain-centered XML-based formats will be introduced in later chapters.

Later on, ontologies transformed the top-down hierarchy into a graph-like architecture. The superior benefit of ontology is the use of inference engines which can infer new information for the asserted one.

2.3 Languages

In order to be able to achieve the role they were designed for, ontologies are represented using formal languages in both syntax and semantics. Having formal defined syntax helps computers determine the correctness of statements, while having formal semantics enables computers to decide the consistency of statements. Using syntax alone provides constraints on how vocabulary of the language is combined. However, semantics are needed so that computers can understand, infer, and detect inconsistencies within the provided vocabulary (Ulicny et al. 2014). It is important for the used language to allow the incremental building, sharing, and using of knowledge.

2.3.1 RDF

Resource Description Framework (RDF) is a standard model for data interchange on the Web provided in 2004 by World Wide Web Consortium (W3C)[1]. This model represents knowledge in a form of semantic graph, in which the concepts are represented by nodes and relations are represented by links between those nodes. By using this simple model, it became possible to mix and share structured and semi-structured data among different applications. RDF consists of triples based on an Entity Attribute Value (EAV) model, in which the subject stands for the entity, the predicate stands for the attribute, and the object stands for the value. Each triple has a unique identifier known as the Uniform Resource Identifier (URI) where a triple

[1] "RDF - Semantic Web Standards." [Online]. Available: https://www.w3.org/RDF/. [Accessed: 20-Aug-2020]

represents a link in the graph. RDF triple's syntax for knowledge representation is demonstrated with the following example[1] and in Figure 2.2:

```
<Fred, hasSpouse, Wilma>
```

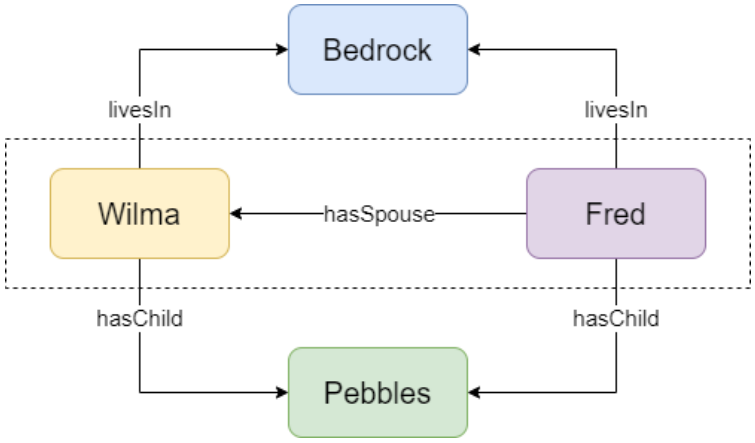

Figure 2.2 – RDF graph example

By adding schema to RDF, they provided us with RDFS which enables us to organize concepts and relations within an RDF into hierarchies. Terms that are provided by RDFS include Domain, Range, subClassOf, and SubPropertyOf. Although both RDF and RDFS provide the needed basic capabilities, its expressivity is not sufficient. There are several required features that are not available, such as cardinality constraints, transitive and inverse properties.

2.3.2 OWL

W3C also provides a language that supports interoperable vocabularies with semantics as ontologies. Web Ontology Language (OWL), which was first introduced in 2004, became the most used language to express ontologies. The latest version, OWL2, was introduced in 2009. OWL aims to[2]:

1. Define classes and properties within a certain domain;
2. Declare exemplary individuals/instances of these classes and properties;
3. Enable reasoning about these classes and instances.

W3C provides 3 sub-languages that are increasingly expressive that meet different usage needs by different communities.

[1] "Data modelling with RDF(S) — GraphDB Free 9.0 documentation." [Online]. Available: http://graphdb.ontotext.com/free/devhub/rdfs.html. [Accessed: 20-Aug-2020]
[2] "OWL Web Ontology Language Guide." [Online]. Available: https://www.w3.org/TR/owl-guide/. [Accessed: 20-Aug-2020]

OWL LITE provides basic constraints and classification hierarchy. Terms that are provided by OWL LITE include Class, Individual, Restriction, differentFrom, sameAs, and inverseOf. OWL DL extends LITE while providing maximum expressiveness and keeping the computational completeness and decidability. OWL FULL extends DL providing the maximum expressiveness and RDF's syntactic freedom without any computational guarantees. Terms that are provided by OWL DL and FULL include oneOf, disjointWith, unionOf, and hasValue. Figure 2.3 shows the nesting sub-languages of OWL.

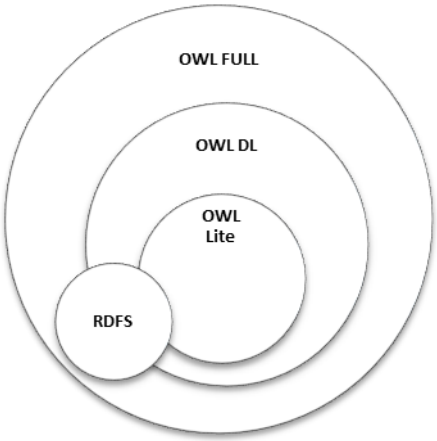

Figure 2.3 – OWL sub-languages (UNIVERSITY OF JYVÄSKYLÄ 2015)

It is worth mentioning that only OWL FULL is considered an extension of RDF. This means that every RDF document is an OWL FULL document. While any OWL (LITE, DL, or FULL) document is an RDF document.

OWL, as a description logic language, consists of two components: TBox and ABox (De Giacomo et al. 1996). TBox represents a set of universally quantified assertions that describe concepts and roles. Whereas Abox covers assertions on individual objects.

2.3.3 Selected Language

In this work, OWL will be used in ontology development for its semantics and interoperability support. OWL can express complex relationships and apply precise constraints on concepts. Also, there are already several ontologies developed in OWL that can be used in the extension process.

2.4 Editing Software

Ontologies can be written using any text editor but that would be more prone to human errors. That also makes it nearly impossible to avoid contradictions within the ontology, because it is hard to track all concepts' relation manually as the ontology grows bigger. Therefore, the use of an editing tool/software is necessary to help keep the ontology under control.

This section will introduce the following tools: Protégé, Fluent, Ontolis, and NeOn. Table 2.1 summarize the features of these editors.

2.4.1 Protégé[1]

Protégé is a free open-source tool that was developed in 2001 by Stanford University, USA. It is currently the most used ontology development tool using desktop and web applications and it has a large active users' community. Protégé has multiple extensions to provide different services like ontology visualization, reasoning engines, and queries execution. However, as it is a local software, it cannot be used by groups of users to edit the same ontology. Last updated version was released in 2019.

2.4.2 Fluent[2]

Fluent Editor is free for individual developers, open source projects, academic research, education, and small professional teams. This editor is developed by Cognitum company, Warsaw, Poland. It supports ontology visualization, reasoning, and Semantic Web Rule Language (SWRL) debugging while being interoperable with Protégé. Its predictive editor prevents having any incorrect sentences both grammatically and morphologically. It also supports collaborative editing for large ontologies using Ontorion server (Seganti et al. 2016).

2.4.3 Ontolis[3]

Ontolis is a commercial web application for managing ontologies and knowledge engineering. This editor is developed in Nuremberg, Germany. It provides collaborative features and web-browser-based graphical rules editor.

[1] "Protégé." [Online]. Available: https://protege.stanford.edu/. [Accessed: 20-Aug-2020]
[2] "Cognitum | Fluent Editor." [Online]. Available: https://www.cognitum.eu/Semantics/FluentEditor/. [Accessed: 20-Aug-2020]
[3] "Home | ONTOLIS." [Online]. Available: https://www.ontolis.com/en/index.html. [Accessed: 20-Aug-2020]

2.4.4 NeOn[1]

NeOn toolkit is a free, open-source, multiplatform ontology engineering environment. It was developed as part of NeOn Project involving 14 European partners. Last updated version was released in 2011.

2.4.5 Selected Editor

Table 2.1 provides a summary about the mentioned tools and a comparison between their features. Within this research, Protégé will be used to develop the ontology as it is free, open source, popular among researchers, covers the needed features and it is the most recently updated tool to the date of this research.

Table 2.1 – Ontology editors summary

Tool	Protégé	Fluent	Ontolis	NeOn
Free	Yes	Yes	No	Yes
Open source	Yes	Yes	No	Yes
Visualization	Yes	Yes	Yes	Yes
Reasoning	Yes	Yes	Yes	Yes
Collaborative	No	Yes	Yes	No
SWRL debugging	No	Yes	No	No
Last update	2019	2016	2018	2011

2.5 Levels

Ontologies are categorized into four levels of abstraction: application, task, domain, and top levels. In this section, these levels will be discussed.

2.5.1 Application

Application ontologies are made with limitations related to the application they are designed for. They are expressed using tractable sub-logic rather than full first order logic (Menzel et al. 2003). Such ontologies provide definitions that are related to the specified application's focus and present concepts' labels tailored for specific users (Malone et al. 2010).

[1] "NeOn Wiki." [Online]. Available: http://neon-toolkit.org/wiki/Main_Page.html. [Accessed: 20-Aug-2020]

2.5.2 Task

Task ontologies are used to express a generic task and describe its vocabulary, and they involve two aspects: task decomposition which is about managing sub-tasks, and knowledge roles which is concerned with task-related concepts and relationships' specification (Martins et al. 2008). Task ontologies aim to reuse task knowledge without the intensity of domain level ontologies.

2.5.3 Domain

A business domain is the term used to describe the categorization of systems into autonomous unit of business, where these units get defined during business analysis. Domain ontologies are used to represent a generic domain and describe its vocabulary (Martins et al. 2008). They consist of domain related concepts that represent its characteristics (Lim et al. 2004).

2.5.4 Top

Top level ontologies, also known as Upper Level ontologies, help making the integration of domain ontologies easier (Hoehndorf 2010). They also provide guidance in the development of new ontologies. Such ontologies provide a general overview over several domains and common categories among all these domains. Top level ontologies are helpful when dealing with multiple domains to achieve semantic interoperability. In addition to supplying axioms as restrictions on the categories. Such restrictions will be inherited by the underlying domain ontologies.

2.5.5 Selected Level

This work intends to develop a domain level ontology to achieve the interoperability goal between all cybersecurity systems available now or in the future. However, as the implementation process might face some limitations, it is planned to start with an achievable core for the said ontology to be expanded in future works.

2.6 Development Methodologies

Several processes and methodologies for developing ontologies were proposed by research groups. In this section, some of these methodologies will be presented.

2.6.1 Agile

Simplified Agile Methodology for Ontology Development (SAMOD) adapted the agile practices from software engineering into ontology development process. This methodology uses small iterations, known as sprints, in order to create well-documented well-developed models (Peroni 2016).

SAMOD states that the development process goes through three phases:

- Pre-Game: identify the goal and scope of the developed ontology, tools, and techniques to be used, competency questions to specify requirements, and available sources;
- Development: several iterates over a cycle consisting of planning, acquiring knowledge, conceptualizing, formalizing, integrating, and reviewing the ontology;
- Post-Game: final preparation of the ontology including evaluation and maintenance.

SAMOD also defines parallel supportive activities like documentation and configuration management. The process requires the collaboration ontology owner, ontology engineer, and ontology user(Abdelghany et al. 2019).

2.6.2 Integration

Methodology of Integration-oriented Ontology Development (MIOD) helps with reusing existing ontologies in the development process (Leung et al. 2011). MIOD consist of six sequential phases:

1. Preparation: problem definition, goal, and scope;
2. Analysis: motivation scenarios and competency questions, conceptualization, existing ontologies assessment;
3. Design: add the missing knowledge to the integrated ontologies;
4. Implementation: using ontology language of choice;
5. Evaluation: requirements verification, scenarios validation, and application test;
6. Maintenance: regular updates according to the usage.

2.6.3 Methontology

Methontology framework helps construct ontologies at knowledge level (Fernandez et al. 1997). It is similar to production lines in its life cycle, where the process starts with the initial need for the ontology towards the final product. This methodology focuses on the activities necessary to build the ontology rather than the order of their execution. These activities can be classified into three main categories (López 1999):

- Management activities: including planning, control, and quality assurance;

- Development activities: including requirements specification, conceptualization, formalization, and ontology implementation;
- Supportive activities: including knowledge acquisition, evaluation, integration, documentation, and configuration management.

2.6.4 Ontology Development 101

Ontology Development 101 (OD 101) describes an iterative process to develop ontologies, starting with general first run to be reviewed and improved along the way (Noy et al. 2001). One rule of thumb in this methodology is that concepts represent physical or logical objects would the nouns and relationships would be verbs in the business domain.

The main steps in the life cycle:

1. Determine domain/scope;
2. Study the possibility of reusing existing ontologies;
3. List important terms/concepts;
4. Define classes and their hierarchy;
5. Define properties;
6. Define facets;
7. Create instances.

2.6.5 Selected Methodology

In this work, OD101 will be used in the development process. The iterative progress help get feedback from business partners on every increment. Which in turn helps overcome any issue on regular basis rather than accumulating them till the end.

2.7 Summary

This chapter introduced the general field of ontologies. As the aim of this work is to develop an ontology, a deeper look was taken into some related aspects, including the levels, tools, and development methodologies. The methodologies were noted to be very similar to each other, with some minor details that may make a different depending on the situation. After considering our needs for this process, Protégé was chosen to develop a domain-level ontology following OD101 methodology. With this chapter, a technical base for development was set, and now the theoretical part will be explored.

3 Cybersecurity and Airports Ontologies

In this chapter, the methodology for conducting a state of the art review will be presented. The aim of this review is to explore the current work and find what is important for the domain. The review will also discover any already existing resources that can be reused to avoid re-inventing the wheel. As a result of the research, available ontologies in the field of cybersecurity and airports, and their characteristics will be introduced. At the end of the chapter, there will be a summary of the main concepts available.

3.1 State of the art Review Methodology

In order to explore existing ontologies in the targeted domain, a state-of-the-art review was conducted. Regarding research methodology that was applied during this work, it was aimed at scientific journal articles and conferences proceedings papers published in English between 2009 and 2019. These papers were obtained by searching the following search engines:

- Web of Science[1];
- Association for Computing Machinery (ACM) Digital Library[2];
- Institute of Electrical and Electronics Engineers (IEEE) Xplore[3];
- Science Direct[4];
- Semantic Scholar[5];
- Google Scholar[6].

[1] "Web of Science" [Online]. Available: https://webofknowledge.com [Accessed: 20-Aug-2020]
[2] "ACM Digital Library" [Online]. Available: https://dl.acm.org [Accessed: 20-Aug-2020].
[3] "IEEE Xplore" [Online]. Available: https://ieeexplore.ieee.org/Xplore/home.jsp [Accessed: 20-Aug-2020].
[4] "Science Direct" [Online]. Available: https://www.sciencedirect.com [Accessed: 20-Aug-2020].
[5] "Semantic Scholar" [Online]. Available: https://www.semanticscholar.org [Accessed: 20-Aug-2020].
[6] "Google Scholar" [Online]. Available: https://scholar.google.com [Accessed: 20-Aug-2020].

The used search query consists of a collection of desired terms that would help retrieve the targeted resources, and they are: "ontology of cyber security" OR "ontology of cybersecurity" OR "cybersecurity ontology" OR "cyber security ontology" OR "ontology for vulnerability" OR "vulnerability ontology" OR "situation awareness in airport" OR "airport ontology". The initial set of search result contained 586 publications. After eliminating 490 duplicated papers and filtering the remaining search results based on content's quality and relevance to our work, resources set was defined consisting of 25. While studying the papers some relevant ones were introduced from their indices and some from experts' recommendations. Figure 3.1 shows the progress of the review.

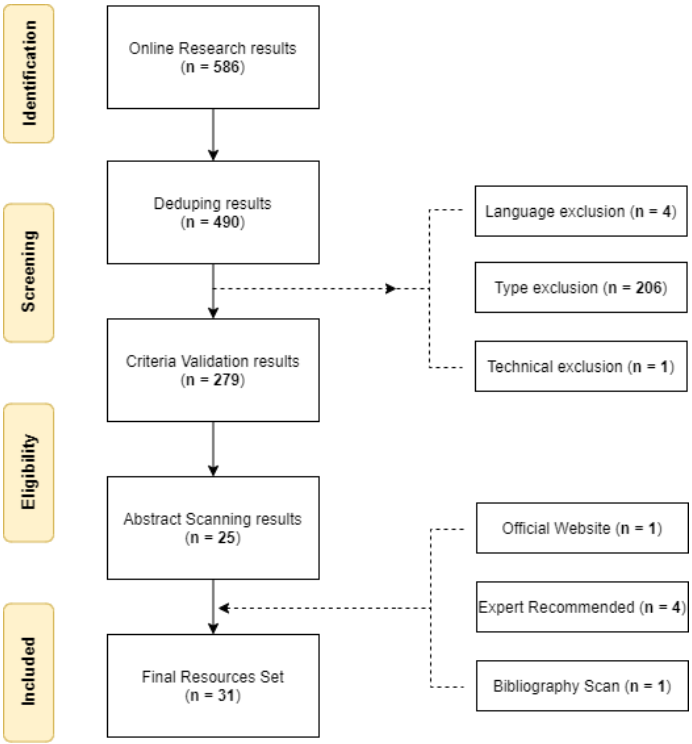

Figure 3.1 – State of the art review flow chart

The final set of 31 resources was reviewed and summarized into the following two sections: cybersecurity related work and airports related work. In addition to journals and conferences' publications, few websites providing official documentation were also used. Figure 3.2 shows the timeline distribution of the research results.

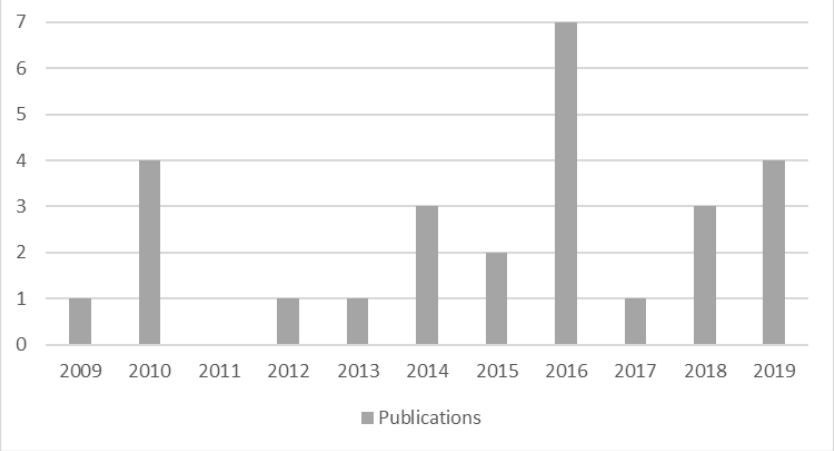

Figure 3.2 – Reviewed research distribution chart

3.2 Cybersecurity Related Ontologies

After studying the current state of available ontologies related to cybersecurity, there are few ontologies that were found most relevant to this research. This section will present these ontologies and their characteristics.

(Wang et al. 2009) have developed an Ontology for Vulnerability Management (OVM) that focuses on software vulnerability by capturing relationships between IT products, vulnerabilities, and other relevant concepts. It is based on multiple vulnerability standards like Common Vulnerabilities and Exposures (CVE) and Common Attack Pattern Enumeration and Classification (CAPEC). Later on, (Wang et al. 2010) elaborated on the uses of OVM as a knowledge base for vulnerability management and vulnerability similarity measurement. This ontology is rich in instances and relationships. OVM was designed for vulnerability analysis and management and it can accurately describe patterns for external threats and internal vulnerabilities. Some of the key concepts defined in OVM are Vulnerability, IT_Product, Attacker, Attack, Consequence, and Countermeasure. Figure 3.3 shows the conceptual model for OVM.

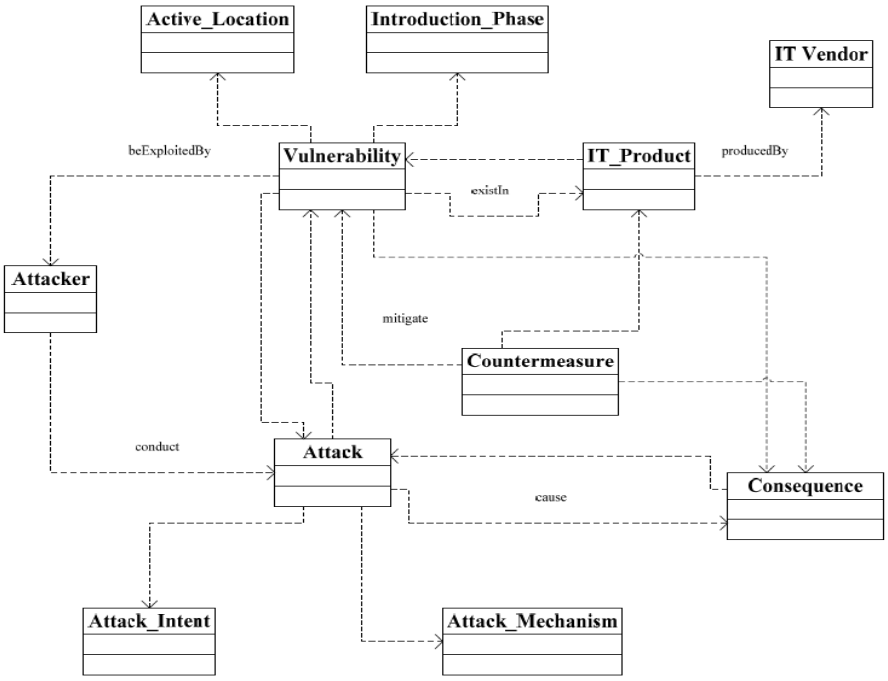

Figure 3.3 – OVM conceptual model (Wang et al. 2009)

In OVM, an Attacker can conduct an Attack to exploit a Vulnerability in an IT_Product. To protect the IT_Product against any Consequences caused by the Attack, Countermeasures can be used to mitigate the Vulnerability.

(Aime et al. 2010) analyzed the limits in several existing vulnerability models and came up with an enhanced vulnerability ontology. They focused on the distinction between vulnerability and threat as they found in their analysis that many frameworks confuse these concepts. The proposed ontology provides an improvement to vulnerability management, and risk assessment. Some of the main concepts included in this ontology are Vulnerability, Threat, Impact, Asset, and Control. Figure 3.4 shows the core model of the proposed ontology.

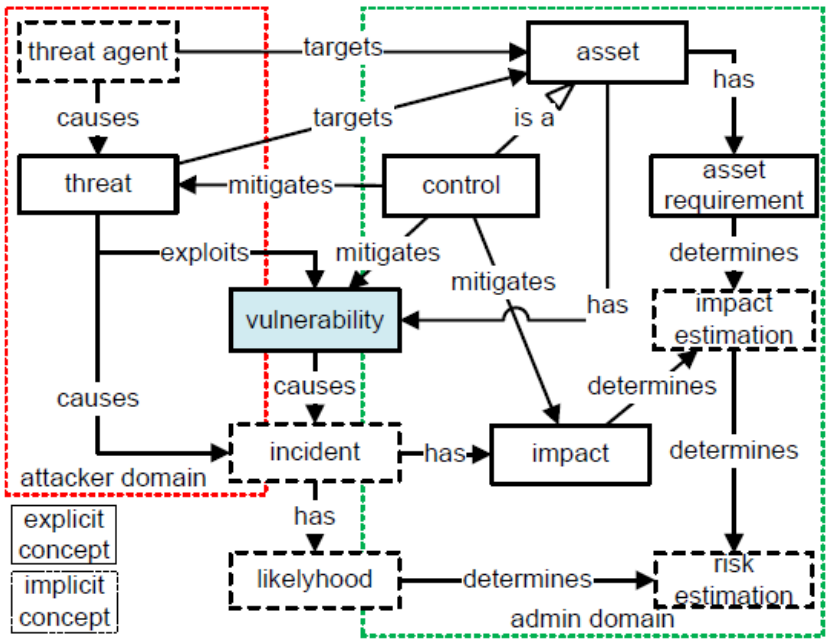

Figure 3.4 – Ontology core model (Aime et al. 2010)

In this ontology, they used Threat to represent a fault that activates a dormant error that is represented by Vulnerability. This activation leads to an Impact of an Incident which is "a concrete error in the intended behavior of the system" (Aime et al. 2010).

(Wita et al. 2010) aimed to describe the relationship between information related to vulnerabilities and their life cycle' phases. Such information would help in setting a vulnerability's priority in light of vulnerabilities increase and administrative resources limitations. For that purpose, they used ontology as a knowledge base and proposed Vulnerability Lifecycle Ontology (*VLO*). This ontology is based on studied vulnerability ontologies, taxonomies, standards, and databases. In addition to Vulnerability concept, the lifecycle phases are important concepts within this ontology which are: Discovery, Disclosure, Exploit, Publicity, and Remediation. The authors claimed that this ontology is useful, when used with information retrieval, for vulnerabilities classification and relevance estimation.

(Gonzalez Granadillo et al. 2012) noticed that most Security Information and Events Management (SIEM) systems are using information syntax more than its semantic, which makes interoperability between different systems difficult. Therefore, they proposed a solution that keeps the existing formats usable and allows for more flexibility. This model distinguishes between information and operation, which assumed to make SIEM more consistent and efficient. Figure 3.5 presents the information class model for the proposed SIEM solution.

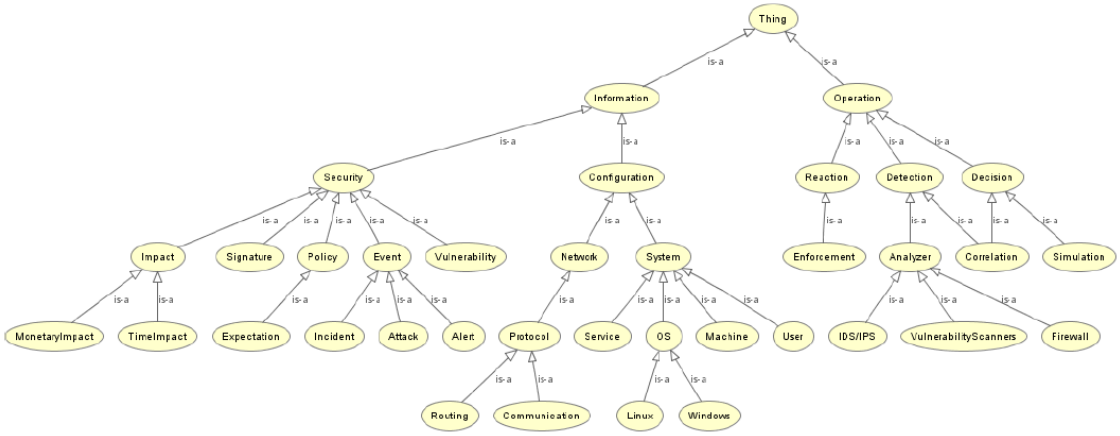

Figure 3.5 – SIEM information class model (Gonzalez Granadillo et al. 2012)

The information part includes several important concepts like Event, Alert, Attack, Incident, and Vulnerability. While the operation one represents the processes with SIEM operations like: Detection, Decision, and Reaction. The Event class is used to represent abnormal activities that require a security device to analyze if it is an Attack or not. The Impact class distinguishes between Time and Monetary cost that a response against intrusion would have.

(Gao et al. 2013) thought that Attacks are getting more complicated and need to be clearly categorized. This would help with conducting security assessment systematically for any system by attack effects' evaluation. They propose an Attack taxonomy consisting of five dimensions: Attach impact, Attack vector, Attack target, Vulnerability, and Defense. Based on that taxonomy and other ontologies, they introduced an Attack ontology with concepts that match the taxonomy's dimensions.

In order to reach shared understanding about cybersecurity, (Mundie et al. 2014) believed that it is best to represent the science of cybersecurity in formal models. This representation would help to obtain a common language that is needed along with a collection of key concepts. They proposed an ontology for incident management to overcome the drawback of previous Incident Management Meta-Models such as partial representation or difficult machine processing of knowledge represented in a natural language. Some key concepts of this ontology include Activities, Defense-hardening-service, Training-service, and Incident-response-services.

(Obrst et al. 2014) wanted to develop cybersecurity domain ontology was with the goal of enabling data integration and providing a formal semantic definition that would make complex queries execution possible. The proposed ontology was made up from modular sub-ontologies that are grouped according to their specificity: Upper, Mid-level, and Domain ontologies. The authors scanned several relevant ontologies and taxonomies to be incorporated into their work. However, they were only focusing on Malware. Some of the main concepts within this ontology are: Malware, TrojanHorse, Exploit, RemoteExploit, LocalExploit, and Spyware. Figure 3.6 shows the class hierarchy of malwares.

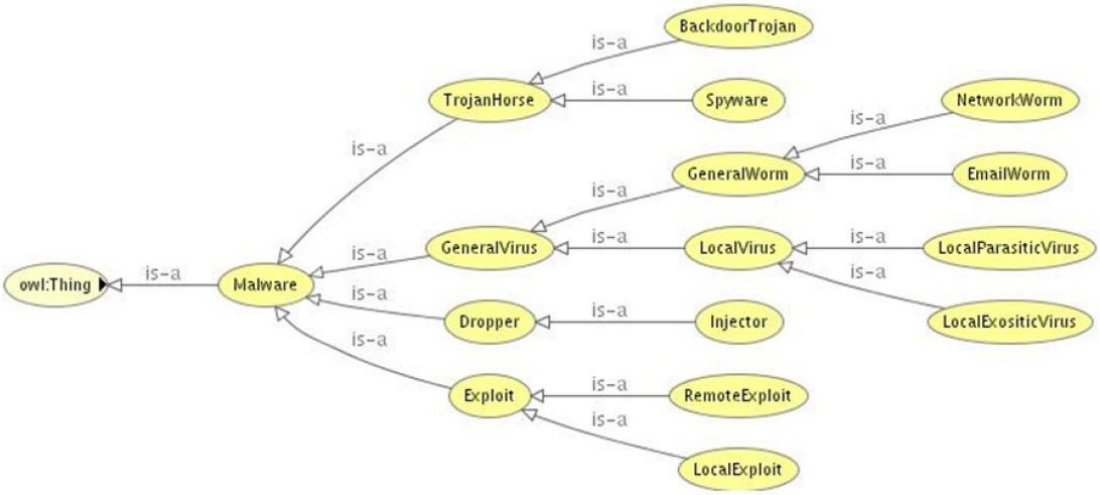

Figure 3.6 – Malware class hierarchy (Obrst et al. 2014)

In this hierarchy diagram, there are clear and well-defined categories of Malware like Trojan horse, Exploit, and Virus. However, this categorization might not be helpful for instances that are combination of multiple classes.

(Oltramari et al. 2014) also adopted the three-layer structure for their proposed ontology CRATELO. The goal of the proposed ontology is to improve cyber defenders' situation awareness. Thus, enhancing the operational decision-making progress. This ontology was based on three other main ontologies: DOLCE-SPRAY, Security Core Ontology (SECCO), and Ontologies of Secure Cyber Operations (OSCO). DOLCE-SPRAY was used as an upper ontology due to its capability to capture conceptual primitives that lay under natural language and common-sense reasoning. The mid-level ontology, SECCO, is more of a security-related ontology that provides key definitions of domain-independent security concepts like Asset, Stakeholder, Threat, and Risk. Finally OSCO ontology, being the domain ontology of cybersecurity, includes some key concepts such as Cyber_operation, Cyber_attack, Cyber_exploitation. In their next paper, (Oltramari et al. 2015) elaborated more on how CRATELO can be used in analysis of multi-level attacks and measurement of reliability and trust. Figure 3.7 depicts a partial representative view of CRATELO.

Figure 3.7 – Representative view of CRATELO (Oltramari et al. 2014)

(Lee et al. 2015) were concerned about cloud security and the security service level agreement. Therefore, they proposed an ontology for Security Service Level Agreement (SSLA) representation that would help understand the security agreements and negotiate levels of security. Main concepts modelled by this ontology include: Vulnerability, AccessControl, Audit, and Transparency. This ontology is thought to be helpful for both Cloud infrastructure and services providers.

(Adesemowo, von Solms, and Botha 2016) focused on presenting IT assets in a structured way. They claimed that their assets' ontology will help with achieving a more integrated security ontology. This ontology has ITAsset concept as a sub-class of Asset in order to support interoperability and better integration with other ontologies. ITAsset is also divided into tangible and intangible. Some of the key concepts is this ontology are Asset, IT Asset, and Risk.

In the field of Supervisory Control and Data Acquisition (SCADA), (Al Balushi, McLaughlin, and Sezer 2016) introduced a new approach for SCADA intrusion detection systems that is based on ontologies. They took advantage of the semantic data definitions to represent knowledge about intrusions in a formal language that can be both human and machine readable. Some of the main concepts provided by this ontology are Attack, Attacker, Impact, and Vulnerability.

A more specific vulnerability analysis has been conducted by (Chen et al. 2016) for metro operation systems. They noticed that vulnerability knowledge was defined by various disciplines and contexts. Therefore, exist different models describing the available vulnerability knowledge which in turn makes it difficult to reuse it. They applied ontology into the vulnerability analysis to establish a basis for a common knowledge base that enables information sharing. Some of the key concepts of this ontology are Vulnerability, Indicator, Control, Impact, and Event. Figure 3.8 shows the conceptual model of the proposed ontology of metro operating system and the internal and external types of vulnerabilities. The internal vulnerabilities include defects and flaws in the metro network's topology. While the external vulnerabilities that are forced by nature and humans.

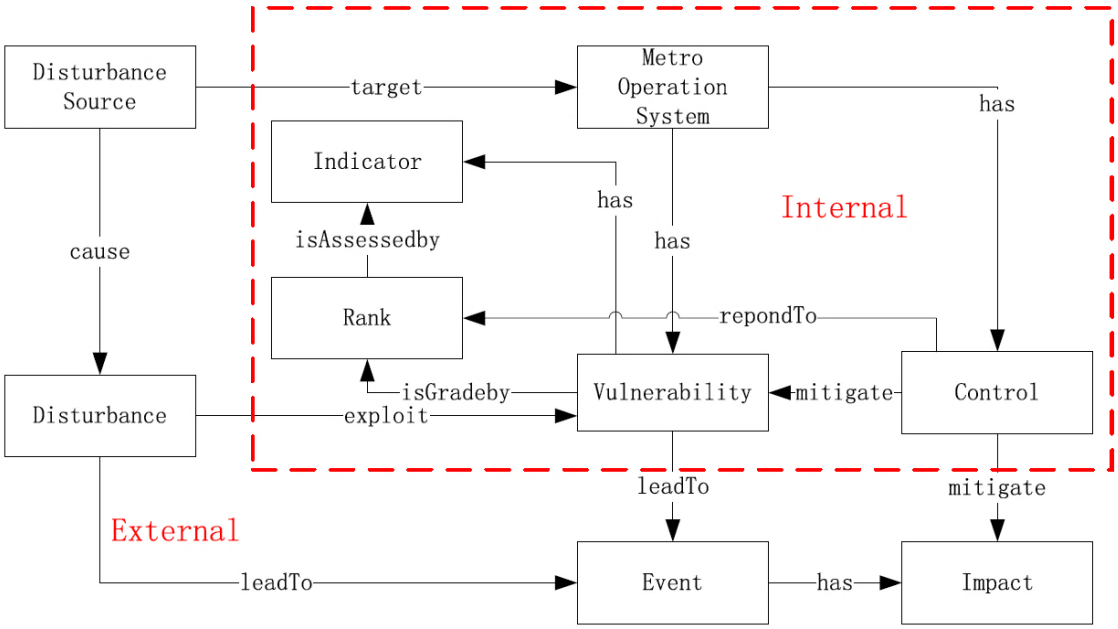

Figure 3.8 – Conceptual model of metro operation system's vulnerability ontology (Chen et al. 2016)

(Kenaza et al. 2016) noticed that, in general, the exchanged information within cooperative intrusion detection systems through SIEM was based on different taxonomies and structured as XML which is lacking in semantic value. Therefore, they proposed an ontology to represent the shared vocabulary used to describe the exchanged information. (Kenaza et al. 2018) further worked on the ontology and introduced it as ONTO-SIEM. This ontology merges several representation formats and information sources and divided the intrusion detection knowledge into conceptual groups. Some of the main concepts in this ontology are Vulnerability, Attack, Attacker, Impact, and Alert. Figure 3.9 shows the main conceptual groups of this ontology.

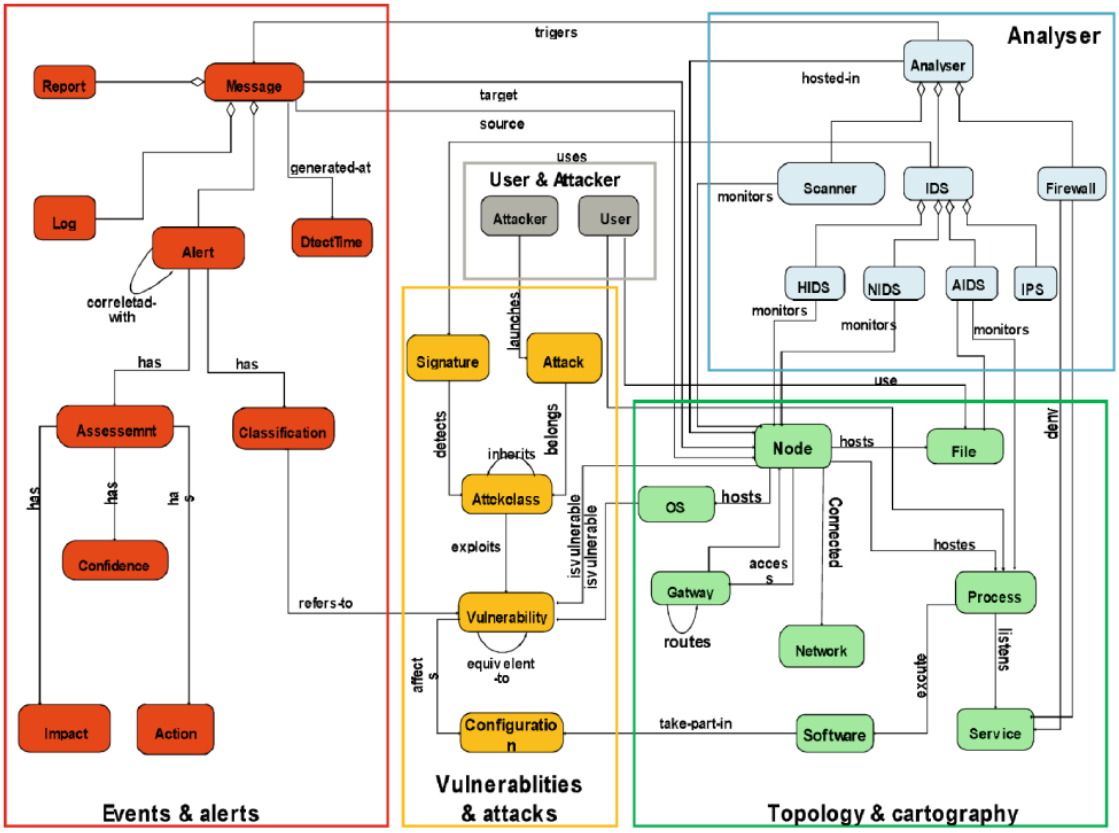

Figure 3.9 – ONTO-SIEM ontology for intrusion detection (Kenaza et al. 2016)

(Krauß et al. 2016) recognized the importance of quick detection and efficient reaction to attack. They proposed an ontology to model the security events, attacks, and vulnerabilities. Alert ontology represents alerts parsed from logs and reports in Intrusion Detection Message Exchange Format (IDMEF) format inspired by (Cuppens-Boulahia et al. 2009), while the Attack ontology represents the attacks inferred by the reasoning component using information like attacker and target. The Vulnerability ontology represents vulnerabilities and security gaps' information in compliance with taxonomies and vulnerability databases. Figure 3.10 shows the Alert part of the ontology. Alert is defined by having Target, Source, Assessment, Classification, Analyzer, Creation Time, and Additional Data.

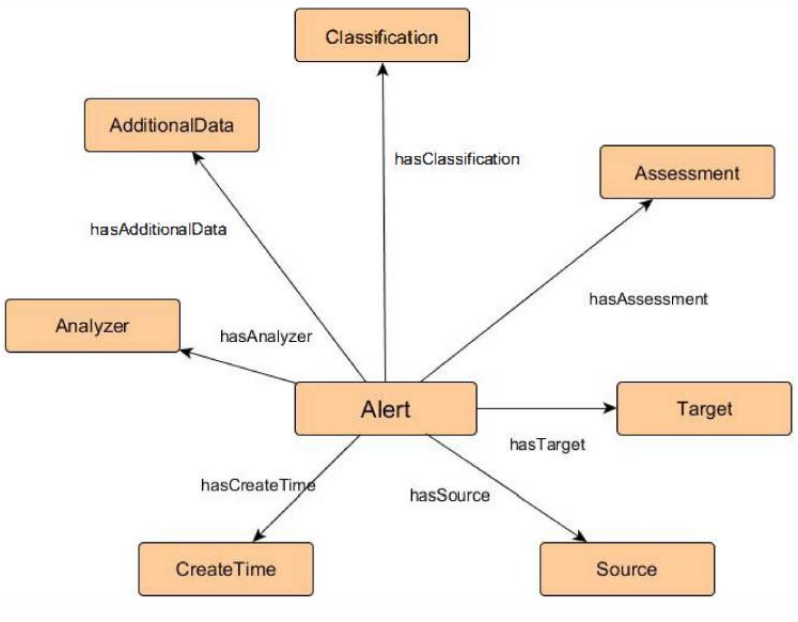

Figure 3.10 – Alert ontology (Krauß et al. 2016)

Unified Cybersecurity Ontology (UCO) is an extension to Intrusion Detection System (IDS) ontology developed earlier in 2004, which integrates different schemas from different systems to obtain data and knowledge related to cybersecurity. This integration helps with the transition from reactive approach to a more proactive and eventually a predictive approach. UCO provides better understanding of cybersecurity by mapping some of the existing ontologies related to this field. UCO uses rules to infer new information which cannot be captured by OWL reasoner (Syed et al. 2016). This ontology can be considered as a semantic version of Structured Threat Information eXpression (STIX), which is an XML representation for cybersecurity vocabulary. In addition to STIX, UCO has been extended with more cybersecurity and general world knowledge resources. The main classes available in UCO include Means, Consequences, Attack, Attacker, Attack Pattern, Exploit, Exploit Target, and Indicator. This ontology's latest version is publicly available on GitHub[1]. Figure 3.11 shows the concepts' diagram of UCO.

[1] "GitHub - Ebiquity/Unified-Cybersecurity-Ontology: Unified Cybersecurity Ontology." [Online]. Available: https://github.com/Ebiquity/Unified-Cybersecurity-Ontology. [Accessed: 20-Aug-2020]

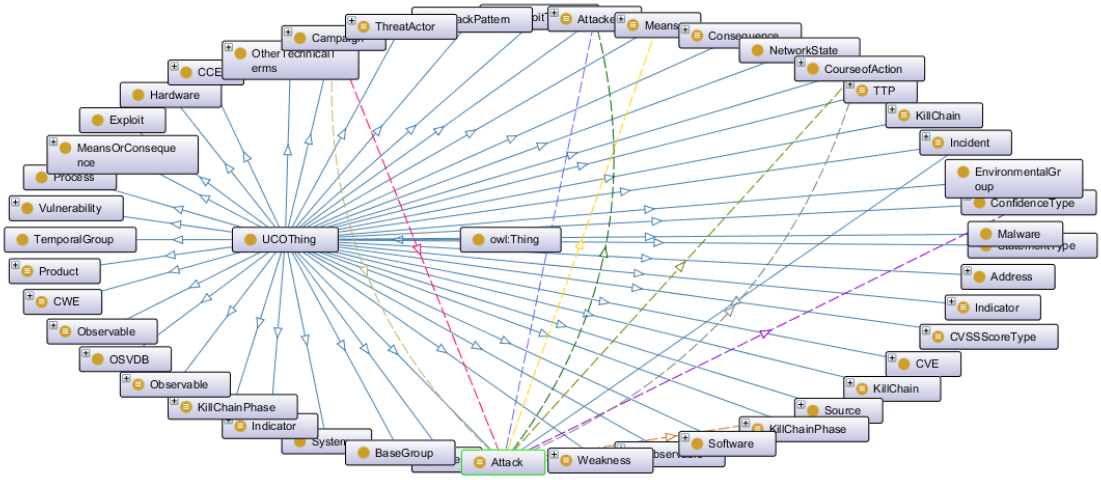

Figure 3.11 – UCO ontology graph (Syed et al. 2016)

(Bergner et al. 2017) were concerned with the frequent hackers' attacks on critical infrastructures i.e. waterworks, governmental institutions, and airports. They worked on an ontology for IT-security concerning these critical infrastructures which merges related requirements and measurement, and vulnerabilities. The proposed ontology consists of sub-ontologies that focus on parts of their work while sharing a common knowledge base: Project, CRITIS, Measures, and IT-Security. Figure 3.12 shows these sub-ontologies and concepts.

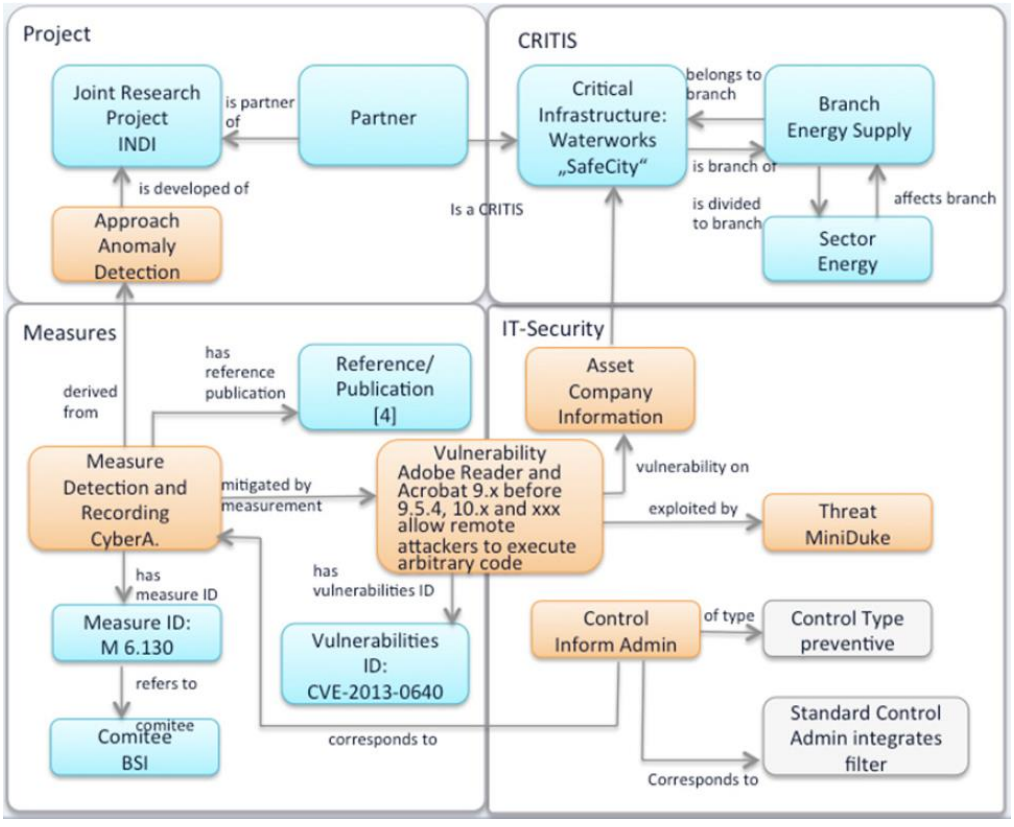

Figure 3.12 – Critical infrastructure sub-ontologies (Bergner et al. 2017)

In alignment with National Institute of Standards and Technology (NIST) framework, (Onwubiko 2018) proposed Cybersecurity Operations Center Ontology Analysis (CoCoa). CoCoa is supposed to provide operation situational awareness to the cybersecurity analysts by moving from log collection to threat intelligence and information sources. Based on the application of CoCoa, they represented a knowledge-based ontology. This ontology would help with understanding cyber incident detection. Figure 3.13 shows the knowledge graph with the ontology concepts. Some of the key concepts in this ontology are Cyber Incident, Alert, Event, Vulnerability, Threat, and Malware.

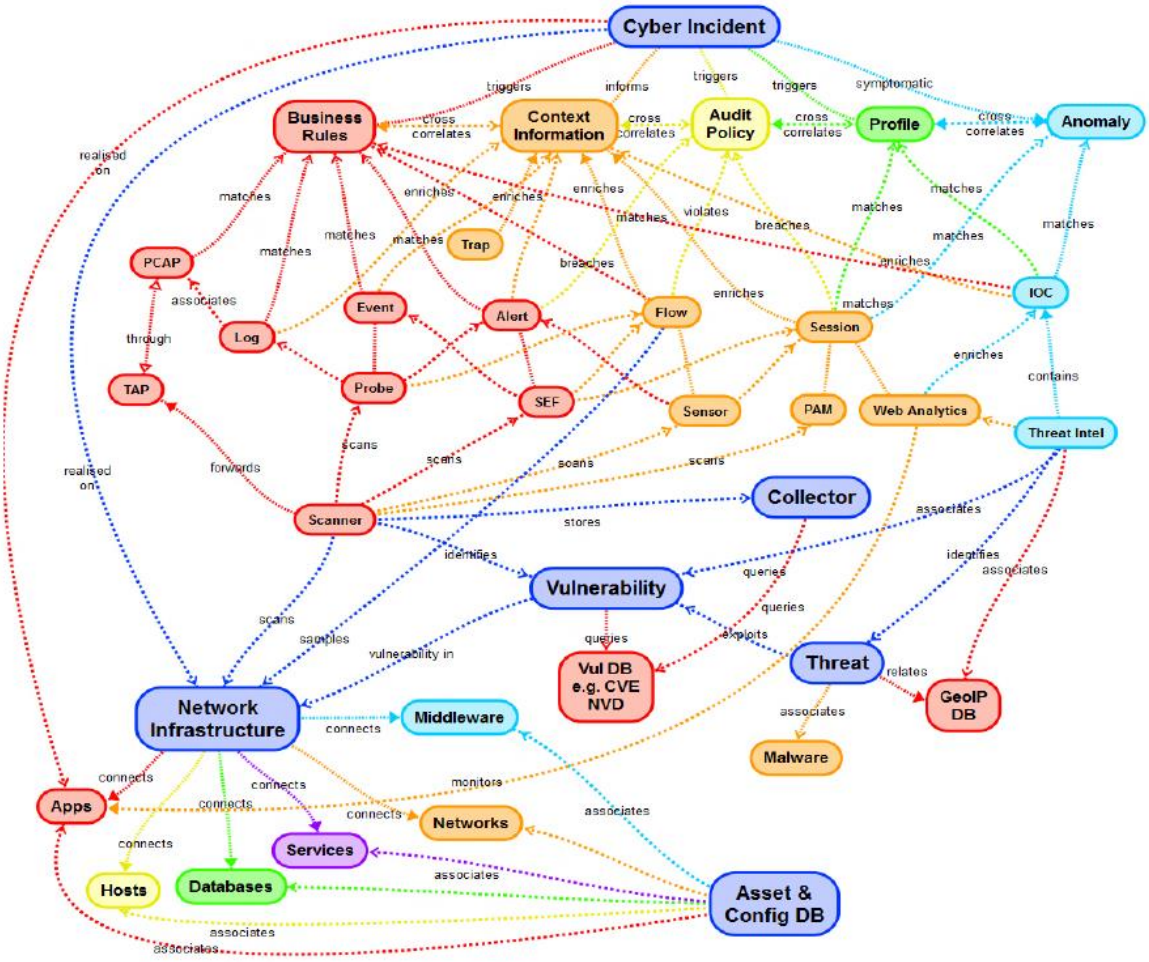

Figure 3.13 – CoCoa Cyber incident ontology-based knowledge graph (Onwubiko 2018)

(Zhao et al. 2018) proposed a cybersecurity ontology and presented a unified model based on the ontology to describe threat intelligence coming from multiple sources and different formats. The model would make threat intelligence sharing and analysis more efficient. The ontology was built after studying the behavior, traffic, and communication characteristics of cyber-attacks and extracting knowledge from them; in addition to analyzing several threat intelligence standards. Some of the concepts available in this ontology are Threat Information, Attack Type, Attack behavior, and Vulnerability. Figure 3.14 shows the main architecture for the proposed ontology.

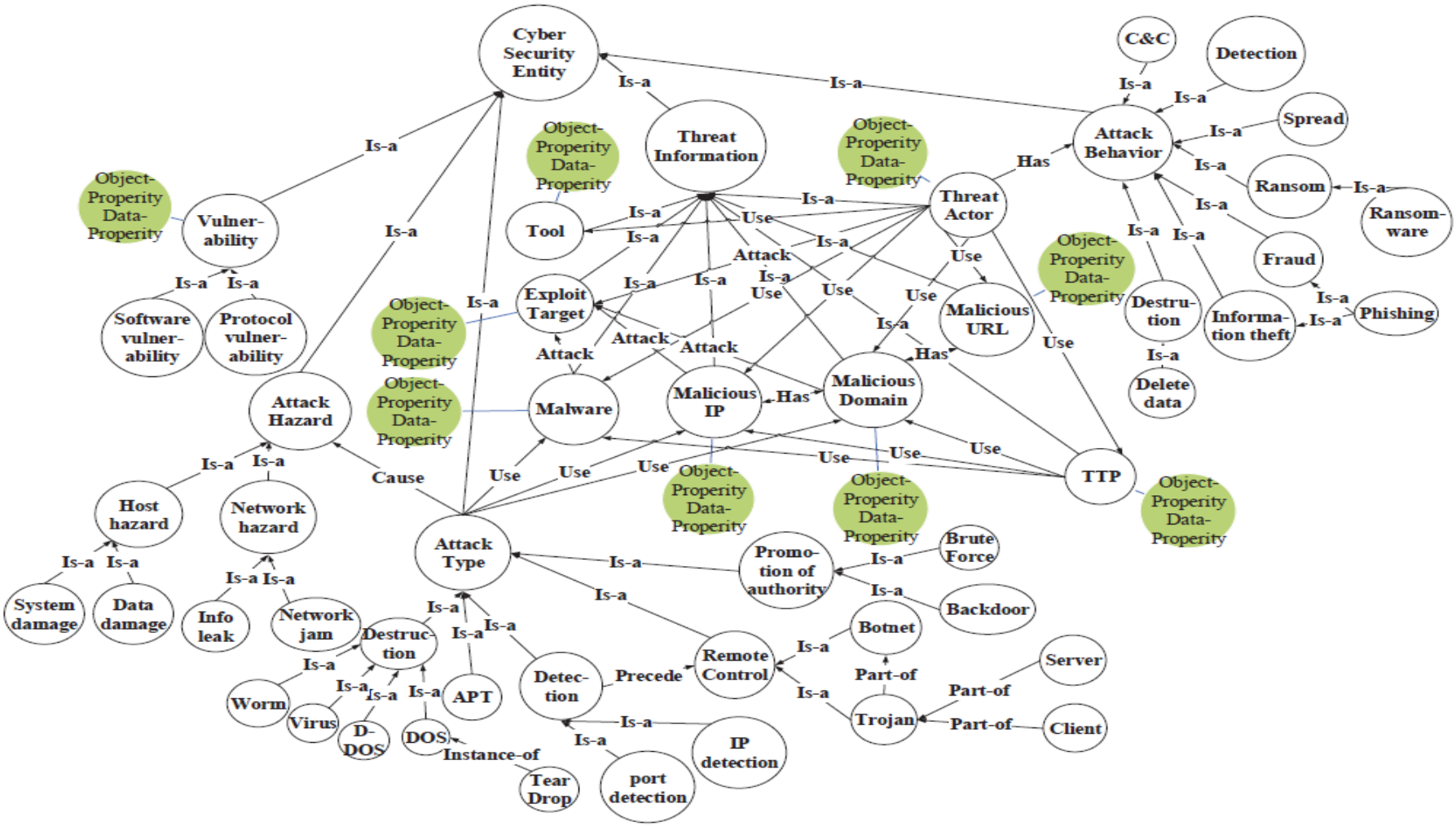

Figure 3.14 – Cybersecurity ontology main architecture (Zhao et al. 2018)

(Doynikova et al. 2019) thought it is important to develop a set of metrics that could be useful for security assessment and decision making. For that purpose, they proposed an ontology focused on cybersecurity assessment metrics. The ontology is divided into four main parts: data sources, security information, infrastructure, and security metrics. In the metrics part, each metric is represented by a separate concept and its value is an instance. Some key concepts in this ontology are Vulnerability, Attacker, and Product. Figure 3.15 depicts a simplified inheritance hierarchy of this ontology.

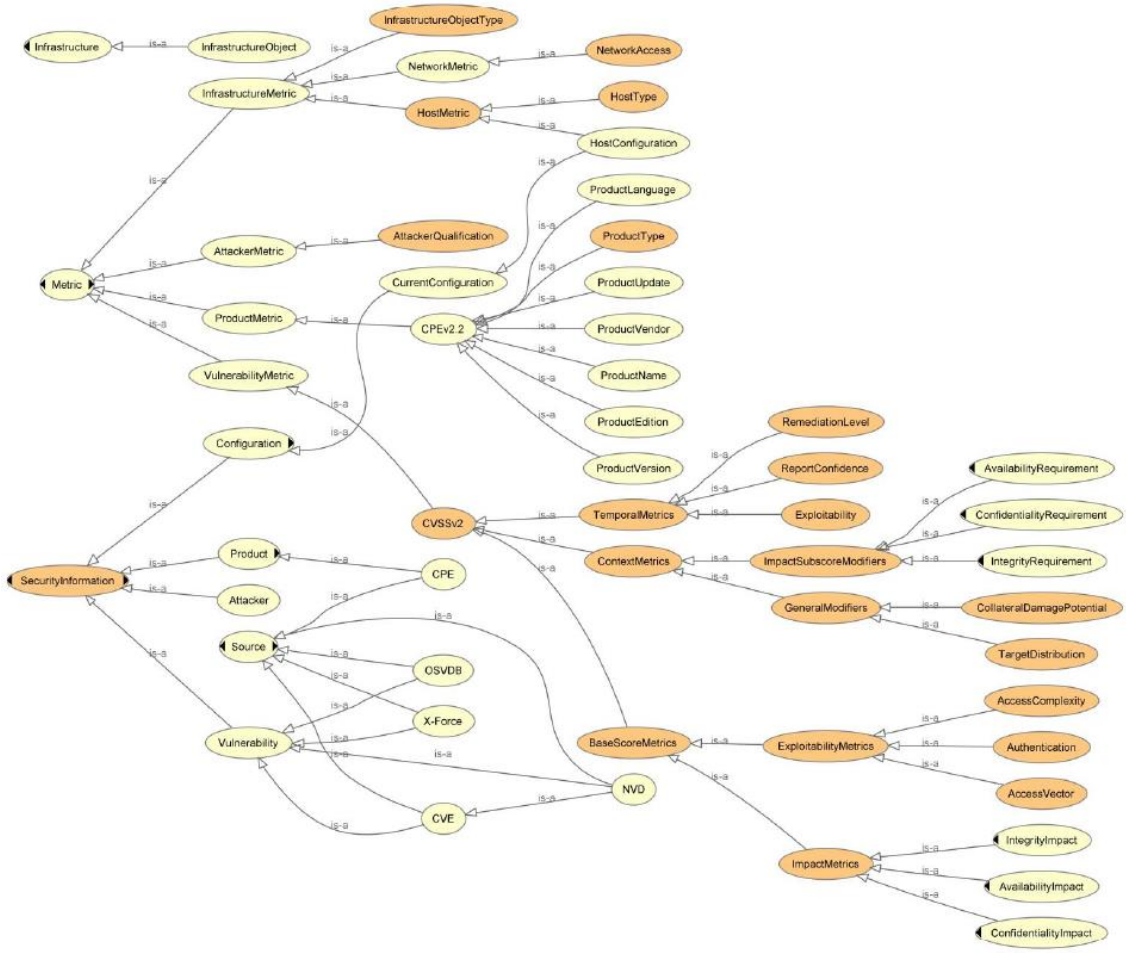

Figure 3.15 – Ontology inheritance hierarchy (Doynikova et al. 2019)

(Greitzer et al. 2019) presented an insider threat ontology called Sociotechnical and Organizational Factors for Insider Threat (SOFIT). The goals intended from this ontology are to share expert knowledge, provide assistance to insider threat assessment profile evaluation, and help with insider threat risk assessment. Some of the main concepts available in this ontology are Actor, Factor, Factor Role, Intention, and Threat Type

(Islam et al. 2019) worked on enhancing the interoperability of security tools that are integrated with Security Orchestration Platforms (SecOrP). They proposed a semantic approach for automatic selection and integration of security tools to attempt automatic execution of incident's response process. This ontology contains three key concepts: Activity, SecurityTool, and Capability. Figure 3.16 depicts the concepts and relation within this proposed ontology.

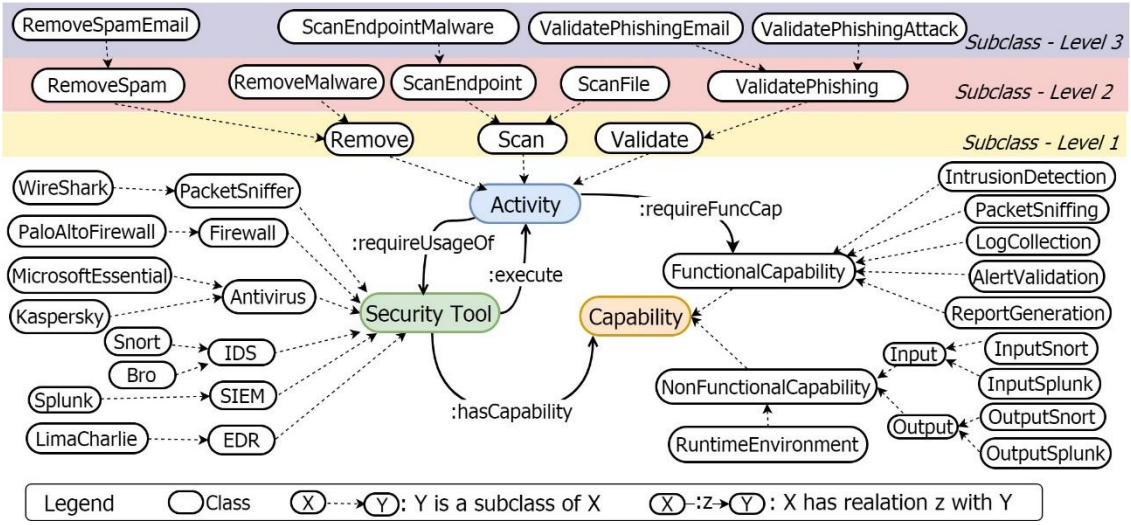

Figure 3.16 – Security tools ontology (Islam et al. 2019)

(Singh et al. 2019) were concerned with cloud computing security as they observed several issues that require more security entities' collaboration. Therefore, they proposed Cloud Security Ontology (CSO) which covers concepts like Cloud Security Threat, Cloud Risk, Cloud Asset, and Cloud Vulnerability. Figure 3.17 shows the CSO architecture.

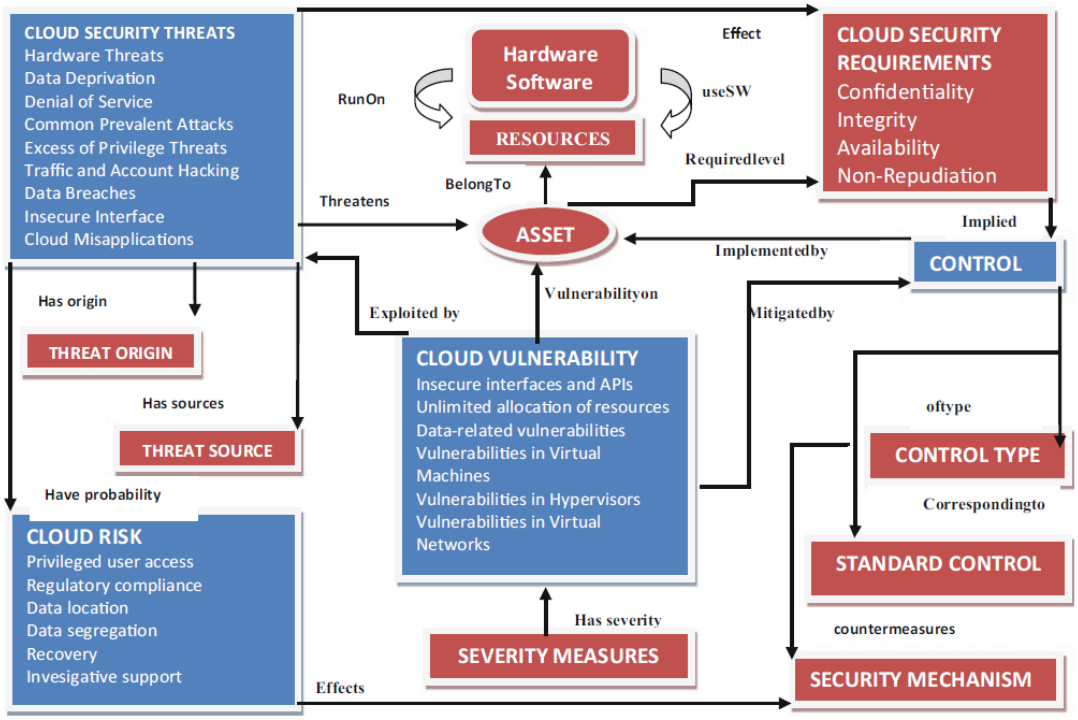

Figure 3.17 – CSO architecture (Singh et al. 2019)

Aside from ontologies, there is another important format that is still useful to work on cybersecurity applications. Incident Object Description Exchange Format (IODEF) is a good example of an XML-based format that is concerned with cybersecurity domain. This format was made to represent reports and indicators of security incidents. It is the common format that most teams of operational security use to communicate. IODEF has specific element to represent critical cybersecurity concepts like Incident and Assessment. The following code example shows the minimal document structure as per IODEF documentation (Danyliw 2007).

```
<?xml version="1.0" encoding="UTF-8"?>
   <!-- Minimum IODEF document -->
   <IODEF-Document version="2.00" xml:lang="en"
      xmlns="urn:ietf:params:xml:ns:iodef-2.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation=
   "http://www.iana.org/assignments/xml-registry/schema/
   iodef-2.0.xsd">
     <Incident purpose="reporting" restriction="private">
       <IncidentID name="csirt.example.com">492382</IncidentID>
       <GenerationTime>2015-07-18T09:00:00-05:00</GenerationTime>
       <Contact type="organization" role="creator">
         <Email>
           <EmailTo>contact@csirt.example.com</EmailTo>
         </Email>
       </Contact>
       <!-- Add more fields to make the document useful -->
     </Incident>
   </IODEF-Document>
```

3.3 Airports Related Ontologies

As part of the project INcreasing Security and Protection through Infrastructure REsilience (INSPIRE), (Choraś et al. 2010) worked on an ontology-based decision support engine to be used in protection of critical infrastructure. The goal of the ontology proposed for this project is to provide interdependencies description between vulnerabilities, SCADA assets, safeguards, source of attacks, and risk-categorized threats. Figure 3.18 depicts the ontology's main concepts and relationships. The diagram shows that Threats can exploit available Vulnerabilities to expose important Assets. Safeguards work on reducing those Vulnerabilities in order to protect the Assets.

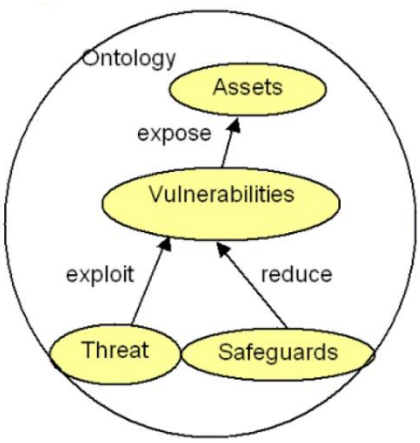

Figure 3.18 – INSPIRE security ontology (Choraś et al. 2010)

Situation Awareness Ontology (SAO) is a specialized ontology designed using OWL to be the core of a framework to manage and reason about events, situations and actions that simplify situation awareness in airports (Tamea et al. 2014). The main class in this ontology is Event which has two sub-classes: Low-Level Event and High-Level Event. Low-level events refer to the events triggered by sensors and can be used by other systems to generate other complicated high-level events. Some of the main relations provided within this ontology are relatedEvents which link Events together, and relatesWith which links Events with other objects like luggage. Another important class is Situation, which represents airport situations during a pre-defined time interval and can be linked to Events.

(Jafer et al. 2016) created an ontology as a first step towards developing Aviation Scenario Definition Language (ASDL). This ontology has two different parts, one that describes the physical model and flights' operation, while the second describes important control tower – pilots communications. The main base high-level concepts of this ontology are: Air_Traffic_Control, Aircraft, Airport, and Weather. Figure 3.19 shows the Aircraft concept's hierarchy for the proposed ontology.

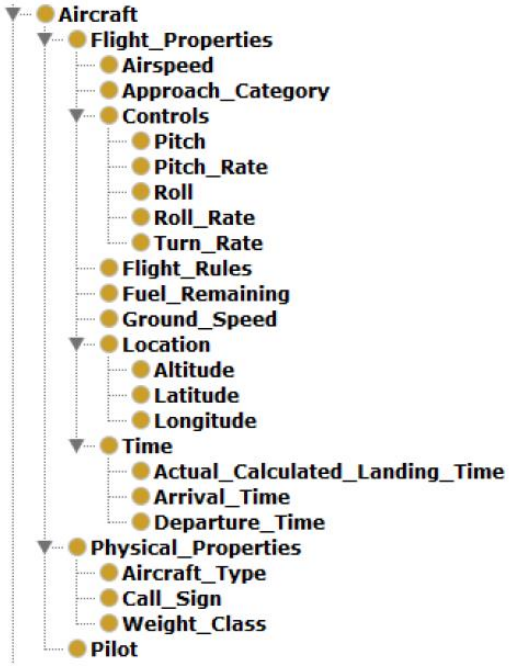

Figure 3.19 – Aircraft concept in ASDL ontology (Jafer et al. 2016)

Air Traffic Management (ATM) Ontology (ATMONTO) is provided by National Aeronautics and Space Administration (NASA)[1]. ATMONTO was released in 2018 and it describes classes, properties, and relationships related to air traffic management general domain. The main entities represented by this ontology include flights, aircraft and manufactures, airport and infrastructure, airlines, US National Airspace System (NAS) facilities, Air Traffic Management Initiatives (TMIs), surface weather conditions and forecasts, airspace components, and departure/arrival routes. NASA provides three interrelated ontologies depending on the level of details that might be required. The ontology is publicly available on the corresponding website. Table 3.1 maps the available features in each layer of ATMONTO.

ATMONTO has been organized into several RDF files that can be imported to any Ontology Development Environment (ODE). Figure 3.20 depicts the ontology graph for ATMONO equipment RDF file.

[1] "The NASA Air Traffic Management Ontology." [Online]. Available: https://data.nasa.gov/ontologies/atmonto/ATM [Accessed: 20-Aug-2020].

Table 3.1 – Layers of ATMONTO and their features

Ontology layer	ATMONTO Core	ATMONTO	ATMONTO Plus
Classes definition	Yes	Yes	Yes
Classes instances	No	Yes	Yes
Property definitions	Yes	Yes	Yes
Property values	No	Yes	Yes
Additional instances[1]	No	No	Yes

Aside from ontologies, there is another useful format that is being used in aviation and airport related applications. Aeronautical Information Exchange Model (AIXM)[2] is a Geography Markup Language (GML) based model that stands as a digital format for aeronautical information. Such information is increasingly getting more complex and made up from several interconnected systems. Therefore, AIXM became a widely used data exchange standard in this domain. This model cover the following conceptual areas (Brunk et al. 2004): Aerodromes, Airspace, Fixes, Routes, Procedures, and Services.

[1] Additional instances provided by ATMONTO Plus cover a sample set of temporally dependent flight, advisory, and weather data from 2014.

[2] AIXM latest version 5.1.1 [Online]. Available: http://www.aixm.aero/page/aixm-51-511 [Accessed: 20-Aug-2020]

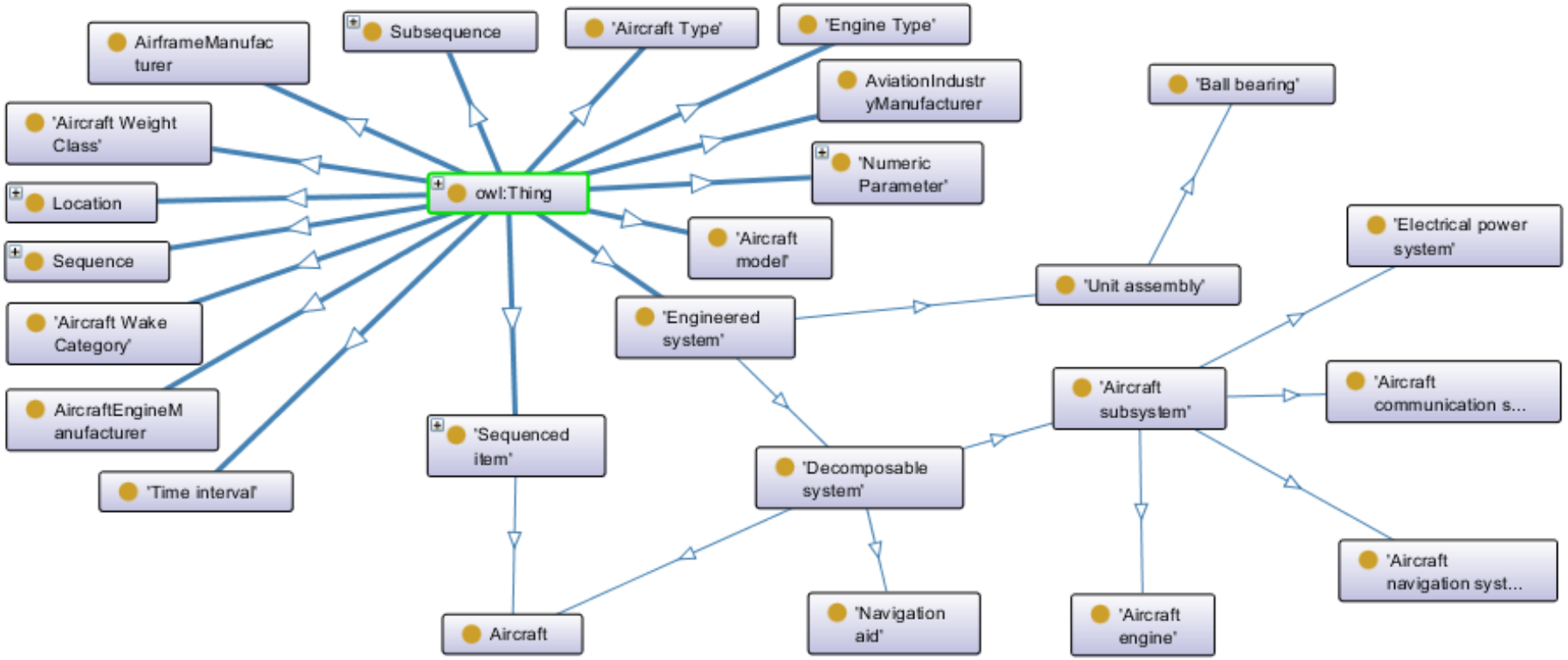

Figure 3.20 – Ontograph for a part of NASA's ATNONTO

3.4 Summary

As it can be seen from the conducted research described in section 3.1, the work so far has been focusing on separate sections of the airports' cybersecurity domain. Different levels of details are provided by different sources, as well as using technologies ranging from XML to ontologies represented in RDF or OWL. To summarize the result of this research, Figure 3.21 plots the usage frequency for concepts in the domain of cybersecurity and airports. The chart shows that the main concepts in overall approaches are: Vulnerability, Asset, Attack, Threat, Attacker, and Impact. The total set of scanned concepts consists of 413 concepts. Concepts with frequency of 1 were not included for the clarity of the chart. This would help learn about the focus points of the domain that reflect the actual needs for the industry.

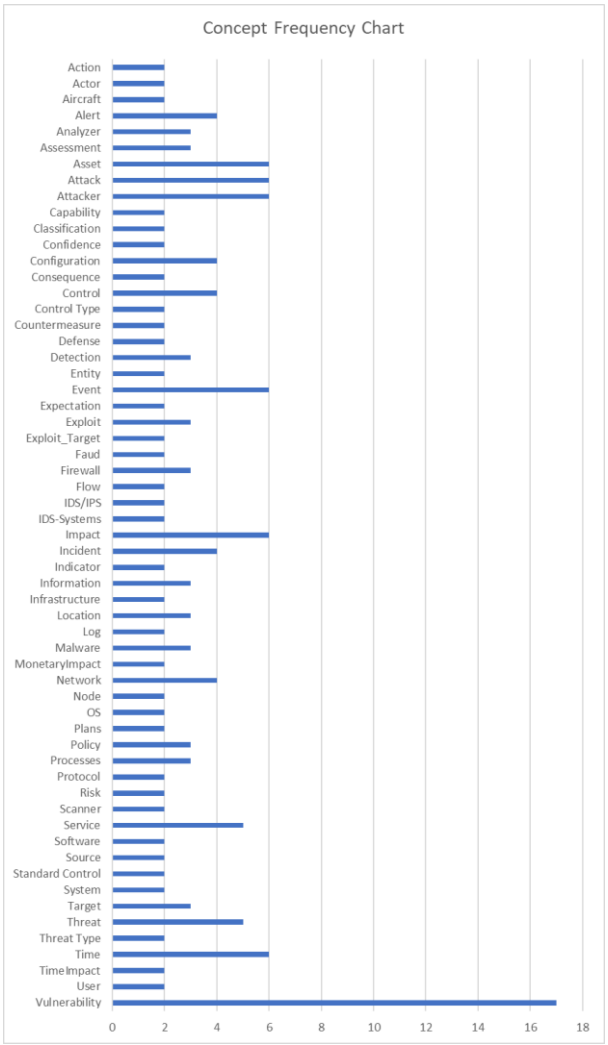

Figure 3.21 – Concepts frequency chart

The studied approaches were analyzed and summarized in terms of their focus points, domain of usage, and other aspects. Table 3.2 shows the previously mentioned ontologies and formats along with their characteristics.

Table 3.2 – Ontologies summary

#	Author(s)	Year	Name	Focus	Related Standards	Usage Domain	Avilablity	Key Concepts	Coverage %
1	Wang & Guo Wang et al.	2009 2010	OVM	Software vulnerability	CVE, CAPEC	Knwoledge base for vulnerability management and similarity measurement	-	Vulnerability, IT_Product, Attacker, Attack, Consequence, Countermeasure	2.66
2	Aime & Guasconi	2010	-	Distinction between vulnerability and threat	-	Vulnerability and risk management	-	Vulnerability, Threat, Impact, Asset, control	2.66
3	Wita er. Al	2010	VLO	Vulnerabilities and their lifecycle information description	CVE, CAPEC, CWE[1], CCE[2], CPE[3], NVD[4]	Knowledge base for vulnerability and life cycle	-	Discovery, Disclosure, Publicity, Remediation	2.91
4	Gonzalez Granadillo et al.	2012	-	Distinction between SIEM information and	-	Interoperability between SIEM systems	-	Event, Alert, Attack, Incident, Vulnerability,	8.47

[1] Common Weakness Enumeration (CWE)
[2] Common Configuration Enumeration (CCE)
[3] Common Platform Enumeration (CPE)
[4] National Vulnerability Database (NVD)

#	Author(s)	Year	Name	Focus	Related Standards	Usage Domain	Avilablity	Key Concepts	Coverage %
				operation				Detection, Deciosion, Reaction	
5	Gao et al.	2013	-	Attacks categorization	CVE, CAPEC, NVD, CWE, CVSS[1]	Systematic security assessment by attack effects' evaluation	-	Attack impact, Attack vector, Attack target, Vulnerability, Defense	1.21
6	Mundie et al.	2014	IMO	Formal models representation of cybersecuirty science	OVM	Incident response organization categorization	-	Activities, Defense-hardening-service, Training-service, Incodent-response-services	30.02
7	Obrst et al.	2014	-	Malware and diamond model aspects	CVE, STIX, CPE, CCE, CVSS	Data integration and formal semantic definition within cybersecuirty domain	-	Maleware, TrojanHorse, Exploit, RemoteExploit, LocalExploit, Spyware,	8.96
8	Oltramari et	201	CRATELLO	Situation	DOLCE-SPRAY,	Operational	-	Asset, Threat,	9.2

[1] Common Vulnerability Scoring System (CVSS)

#	Author(s)	Year	Name	Focus	Related Standards	Usage Domain	Avilablity	Key Concepts	Coverage %
	al.	4 201 5		awareness	SECCO, OSCO, CVE CWE, CAPEC, STIX	decision making progress enhancement		Stakeholder, Risk, Cyber_operation Cyber_attack Cyber_exploitation	
9	Lee et al.	201 5	-	Cloud security and SSLA	CSA[1], CCM[2]	Securing cloud infrastructure and services	-	Vulnerability, AccessControl, Audit, Transparancy,	3.15
10	Adesemowo , von Solms, and Botha	201 6	-	IT assets categorization	ISO 22274, SO[3]	Information risk in knowledge economy	-	Asset, IT Asset, Risk	1.69
11	Al Balushi, McLaughlin, and Sezer	201 6	-	SCADA intrusion detection systems	-	Malicious industrial communication s and intrusion detection	-	Attack, Attacker, Impact, Vulnerability	1.94
12	Chen et al.	201 6	-	Metro system vulnerability analysis	-	Vulnerability knowledge base	-	Vulnerability, Indicator, Control, Impact, Event	1.94
13	Kenaza &	201	ONTO-	Cooperative	IDMEF,	Information	-	Vulnerability,	6.54

[1] Cloud Security Alliance (CSA)
[2] Cloud Controls Matrix (CCM)
[3] Security Ontology (SO)

#	Author(s)	Year	Name	Focus	Related Standards	Usage Domain	Avilablity	Key Concepts	Coverage %
	Aiash Kenaza et al.	6 2018	SIEM	intrusion detection systems through SIEM	STIX, NVD, OVAL	exchange vocabulary		Attack, Attacker, Impact, Alert	
14	Krauß & Thomalla	2016	-	Security events, attacks, and vulnerabilities	IDMEF, CVE, NVD	Attack detection and reaction effciency enhancement	-	Alert, Attack, Vulnerability, Target (Asset), Source (Attacker)	1.69
15	Syed et al.	2016	UCO	Intrusion detection systems	STIX, CVE, CCE, CAPEC, CYBOX[1]	Cybersecuirty systems integration and knowledge extraction	Public	Means, Consequences, Attack, Attacker, Attack Pattern, Exploit, Expolit Target, Indicator	1.94
16	Bergner & Lechner	2017	-	IT security concerning critical infrastructure (like waterworks and airports)	CVE, CRITIS[2]	Securing critical infrastructure against frequenct hacker attack	-	Asset, Threat, Control, Vulnerability	3.87
17	Onwubiko	2018	COCOA	Operational situation awareness	CVE, NVD, PAM[3]	Threat intelligence knowledge base	-	Cyber Incident, Event, Alert, Vulnerability,	6.78

[1] Cyber Observable Expression (CYBOX)
[2] Critical Infrastructure Ontology (CRITIS)
[3] Privileged Access Management (PAM)

#	Author(s)	Year	Name	Focus	Related Standards	Usage Domain	Avilablity	Key Concepts	Coverage %
								Threat, Malware	
18	Zhao et al.	2018	-	Threat intelligence from multiple sources	JSON[1]	Threat intelligenct sharing and analysis enhamncement	-	Threat Information, Attack Type, Attack Behavior, Vulnerability, Virus, Malware	11.14
19	Doynikova et al.	2019	-	Cybersecurity assessment metrics	CVE, CWE, CAPEC, CCE	Security assessment and decision making	-	Vulnerability, Attacker, Product	4.84
20	Greitzer et al.	2019	SOFIT	Insider threats factors	-	Insider threat risk management and expert knowledge sharing	-	Acotr, Factor, Factor Role, Intention, Threat Type	3.87
21	Islam et al.	2019	-	Security orchestration platfroms integration	STIX	Automatic incident response process	-	Activity, SecurityTool, Capability	2.18
22	Signh & Pndey	2019	CSO	Cloud computing security	-	Securing cloud computing services	-	Cloud Security Threat, Cloud Risk, Cloud	2.42

[1] JavaScript Object Notation (JSON)

#	Author(s)	Year	Name	Focus	Related Standards	Usage Domain	Avilablity	Key Concepts	Coverage %
								Asset, Cloud Vulnerability	
23	Choras et al.	2010	INSPIRE	Interdependencies description between vulnerabilities, SCADA assets, and others	CVE, CCE, SCAP1, CPE, CVSS	Decision support engine for critical infrastructure protection	-	Assets, Vulnerabilities, Threat, Safeguards	0.97
24	Tamea et al.	2014	SAO	Airports situatoin awareness and events management	-	Reasoning about events, situations, and actions related to airport security	-	Event	0.73
25	Jafer et al.	2016	ASDL	Flights' operation and control tower-pilot communication	-	Aviation management	-	Air_Traffic_Control, Aircraft, Airport, Weather	1.45
26	NASA	2018	ATMONTO	Aircrafts and flights management	-	Air traffic and airspace components managemt	Public	Aircraft	2.66
27	Danyliw	200	IODEF	Incidient object	-	Security	Public	Incidnet,	3.63

[1] Security Content Automation Protocol (SCAP)

#	Author(s)	Year	Name	Focus	Related Standards	Usage Domain	Avilablity	Key Concepts	Coverage %
		7		description		incidents reporting		Assessment	
28	Brunk & Porosnicu	2004	AIXM	Avaiation and airport applications	-	Information exchange between interconnected systems	Public	Aerodromes, Airspace, Routes, Services	0.97

From this summary it can be seen that there are many approaches to cybersecurity already. However, each approach has its own focus and interest in this domain. Even though some of the listed approaches do overlap, as the maximum concept frequency is 17 out of 28 for Vulnerability, many are isolated. In addition to the low coverage rate in respect to the whole defined set. Which means their information may not be easily used nor understood by others, hence the need for an integrated ontology that covers as much as possible of the domain needs in order to facilitate and clarify communication.

In this chapter, the process of the state of the art review was introduced, and its findings were described and summarized. With this, the theoretical and practical background is covered and ready to advance into constructing the new ontology.

4 Value Analysis

This chapter will present the value analysis of the proposed ontology. Fuzzy Front-End (Koen et al. 2014) approach will be considered when discussing the opportunity available for the intended ontology.

4.1 Innovation Front-End

Innovation Front-End or Fuzzy Front-End is a part of the product engineering development process that is concerned in innovation and idea management. This provides a base for success in later phases and makes sure not to miss any opportunities.

4.1.1 Opportunity Identification

The importance of this airports' cybersecurity ontology comes from the importance of the key components: airports and security. Airports are one of the largest investments in any country, in addition to dealing with the most important cargo, which is people. Therefore, securing airport is a necessary task in every possible aspect. Cybersecurity, like physical security, is increasingly valued along with the technological advancement. However, for this field of application, the work so far has been separated and isolated. Which means there are many efforts being duplicated instead of being shared and integrated. Having all interested parties sharing the experience and work will help save both time and effort in the future, while expanding the work perimeter even further than each of them would on their own.

As for the used technologies, the basic format of representation was raw text logs. Text files are easier to write due to the absence of structure constraints; this means text files are not very useful or easy to process. Even with enforcing several standards and formats to give them some structure, text logs are still hard to use for knowledge extraction. XML came along to provide clear structure and hierarchy to textual content with useful validation and parsing capabilities. Later, ontologies represented with RDFs or OWL helped to transform the

interrelation form top-down hierarchy to more graph-like connections. The superior benefit of ontology is the use of inference engines which can infer new information from the asserted one. Table 4.1 shows a comparison between Text, XML, and Ontology languages.

Table 4.1 – Technology comparison

	Text	XML	Ontology languages
Structure	No	Yes	Yes
Validation	No	Yes	Yes
Inference	No	No	Yes

According to the research conducted in the previous chapter, the interest in cybersecurity ontologies has been increasing in the past decade. Figure 4.1 depicts the trend of research in this domain with a linear forecast of increased interest.

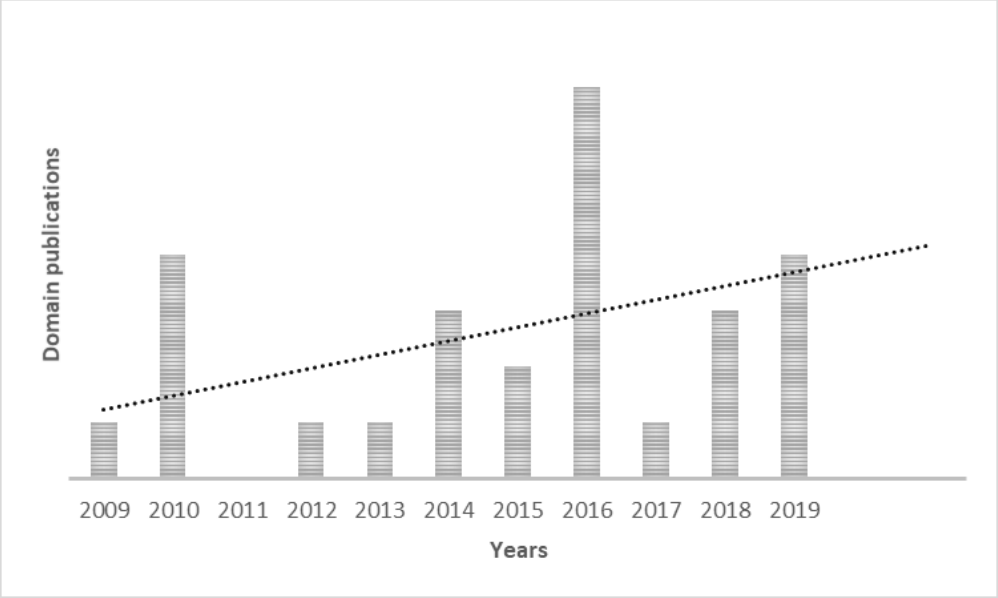

Figure 4.1 – Research trend analysis

4.1.2 Opportunity Analysis

As the preliminary research indicated, the work on ontology integration is still limited. This presents a huge opportunity to be exploited, as the integrated ontology would be able to make a substantial contribution towards knowledge exchange's improvement. This integrated ontology would serve as a common ground for all airports and security related organizations to cooperate, share information, and exchange knowledge in a way that all stakeholders can understand and work with. After that, it would become a central base for any future enhancement that can benefit all involved entities.

4.2 Solution Value

This section will further explore the value of the expected outcome of this work. Discussion about the ontology's value, benefits and sacrifices will be presented in addition to the analysis of the users' pains and gains. Finally, a SWOT analysis will be available to summarize the advantages and disadvantages of the ontology.

4.2.1 Value

In the process of value analysis, it is distinguished between two important terms: User Value and Perceived Value (Moliner 2009). The first term refers to the actual value of the product or the service regardless of any other expectations. While the later term refers to the value from consumers point of view.

4.2.2 User Value

In order to achieve the goal of securing airports, it is necessary for all involved systems to be able to communicate among each other. Each system specialized in a segment of the whole process and manages the information obtained from it. Information from different systems can be represented in different formats which makes it harder for others to make use of. Which in turn, would affect the overall performance. Therefore, all involved parties would highly value a way to integrate all the available information sources into a more unified representation.

4.2.3 Perceived Value

The collaboration of multiple concerned parties would lead to an increase in the quality of the resulted work. That is because they would present different points of view and help detect any shortage that one side would miss on their own. Also, as the resulted ontology would have to serve as common base for different systems, that would motivate the concerned parties to focus on making it more reliable for all of them. As the ontology would start out by incorporating existing knowledge, it would be built in a flexible manner that would enable coverage over different sources and formats. This flexibility would be useful towards future expansion as the business requirements evolve with time. Technical competence and overall trust would also increase due to this collaboration and integration of sources.

On the other hand, the initial phase might take a considerable amount of time, effort, and energy. The establishment of the integrated base would require a lot of teamwork and discussion in order to obtain a consensus on the concepts and relations and resolve any conflicts that might occur. However, once this phase is complete, the work will get easier and the benefits to come will compensate for the rough start. Table 4.2 summarizes the benefits and sacrifices of this work based on the guidelines made by (Woodall 2003).

Table 4.2 – Benefits and sacrifices summary

	Product	Service	Relationship
Benefit	Quality	Reliability Flexibility Technical Competence	Trust
Sacrifice			Time/Effort/Energy Conflict

4.2.4 Value Proposition

(Radziwill 2015) addressed the relationship between products and consumers in a form called Value Proposition Canvas (VPC) that help to easily correlate the intended work with customers' actual needs. Using the VPC template provided by Strategyzer[1] will help give an overall idea of this correlation. Figure 4.2 depicts the correlation between the needs and the features of the proposed ontology. It can be seen that the integrated ontology and the semantic layer that it provides can solve most of the customers' pains in order to improve the interoperability among all concerned parties.

[1] Strategyzer [Online]. Available: https://platform.strategyzer.com/resources [Accessed: 20-Aug-2020]

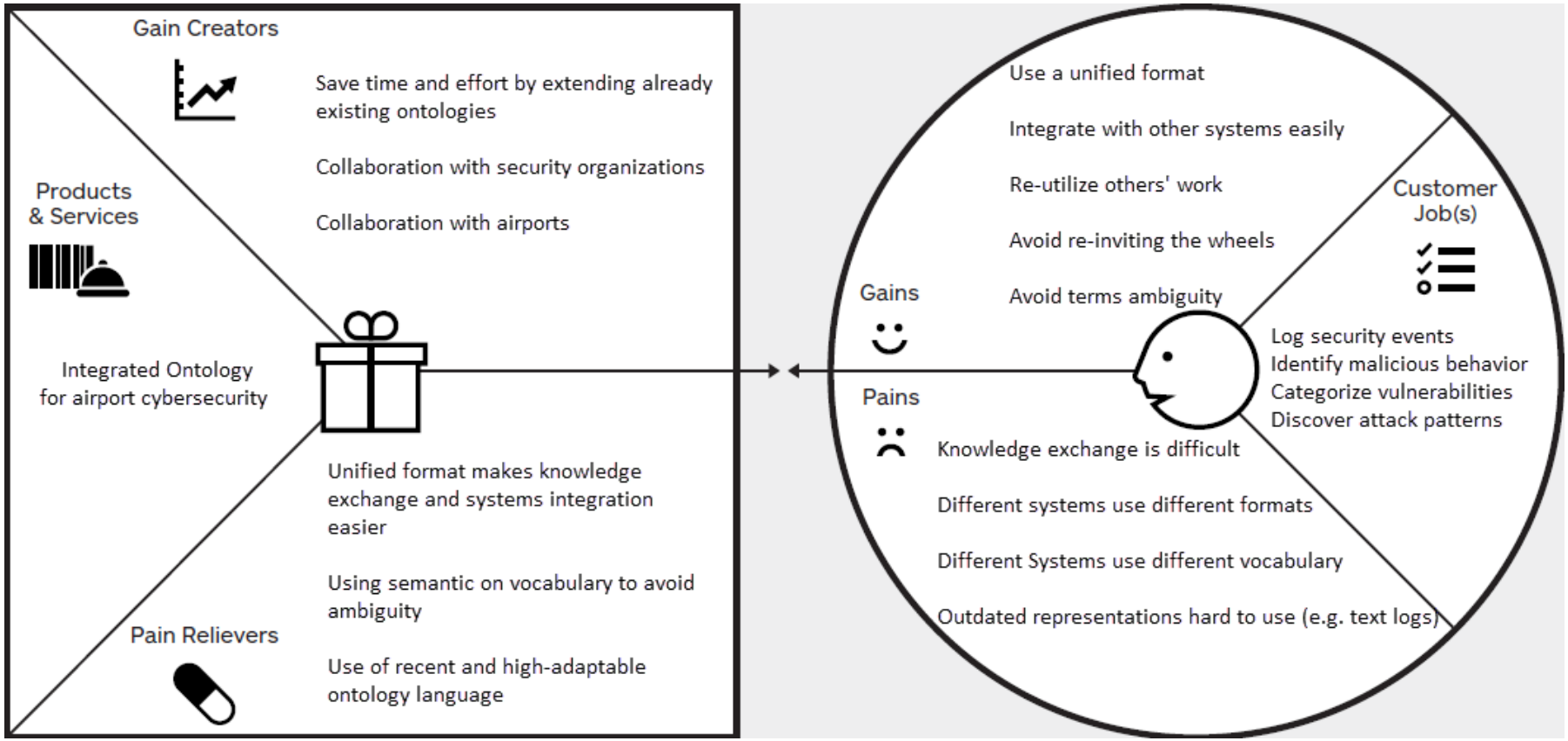

Figure 4.2 – Value Proposition Canvas

4.2.5 SWOT Analysis

In SWOT analysis, a summary of the overall aspects of this work is presented in categories: internal or external origin, and helpful or harmful affect. This categorization will lead to four factors contributing to the SWOT analysis which are: Strengths, Weaknesses, Opportunities, and Threats. The main goal of SWOT analysis is to highlight the strengths and opportunities to take advantage of them. In the meantime, it also helps to warn about the weaknesses and threats in order to avoid falling in disadvantage because of them (Pershing 2006). Table 4.3 shows the details of SWOT analysis.

Table 4.3 – SWOT analysis

	Helpful	Harmful
Internal	Strengths: Use of ontologies unifies the available systems' interfaces Semantic layer provides protection against terms' confusion	Weaknesses: Initial phase might be time consuming in order to gather consensus concepts' definitions
External	Opportunities: Limited work on this specific topic Several attempts at separates fields Several taxonomies available	Threats: Not all available ontologies are available for public use Not all parties might be willing to cooperate in timely fashion

4.2.6 Summary

In this chapter, the importance of this work towards the market was introduced. The value is based on the importance of the airports and cybersecurity domain. The opportunity to provide the market with improved interoperability was discussed. In addition to the chance to contribute to the scientific community.

5 Design

In this chapter, the ontology's design phase will be presented. First, domain concepts will be discussed to introduce the consensus concepts set. Then, ontology's requirements will be documented. The progress of the design process will be explained to show the reasoning and evolution of the ontology along, as well as description logic representation with a final summary.

5.1 Domain Concepts

As seen through the ontologies described in section 3, there is no single ontology that meets all the requirements of this research. However, some work has been done in various fields that can be built upon instead of starting from scratch. Due to accessibility issues, only UCO and ATMONTO ontologies are publicly available from the set of reviewed papers. Therefore, these ontologies might be the only possible base that can be used for further expansion.

As a start for the extension process, we will begin with some main concepts that are essential to this domain, such as Alert, Asset, Event, and Incident. These concepts were chosen based on the study of IDS and the conducted state if the art review, as well as SATIE partners' concerns expressed in several meetings. In order to develop a general understanding of the targeted concepts, a survey was conducted covering some major cybersecurity organizations, such as::

- European Union Agency for Cybersecurity (ENISA)
- National Institute of Standards and Technology (NIST);
- Cybersecurity Library, Italy (CYBRARY.IT);
- The National Cyber Security Strategy (NCSS), Republic of Croatia (HR);
- Agence Nationale de la Sécurité des Systèmes d'Information, France (ANSSI);
- Centro Nacional de CiberSegurança, Portugal (CNCS);
- Associazione Italiana per la Sicurezza Informatica, Italy (CLUSIT);
- The National Cyber Security Strategy (NCSS), Italy (IT).

The result of the survey is detailed in tables Table 5.1, Table 5.2Table 5.3Table 5.4Table 5.5Table 5.6 to Table 5.7. Definitions available in native languages have been translated to English.

Table 5.1 – Survey result for Attack concept

Source	Attack Definition
NIST[1]	- An attempt by an unauthorized individual to fool a Verifier or a Relying Party into believing that the unauthorized individual in question is the Subscriber. - The realization of some specific threat that impacts the confidentiality, integrity, accountability, or availability of a computational resource. - Cyber-attack[2]: An attack, via cyberspace, targeting an enterprise's use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information.
CYBRARY.IT[3]	An attack is an action with malicious intention to interrupt the operations of a network or steal the data, etc.
HR NCSS[4]	Cyber (computer) crime: criminal offences against computer systems, software support and data; committed in cyberspace using information and communication technologies.

Table 5.2 – Survey result for Alert concept

Source	Alert Definition
NIST[5]	Notification that a specific attack has been directed at an organization's information systems.
CYBRARY.IT[6]	Alert Situation: An alert situation is when the interruption in an enterprise is not resolved even after the competition of the threshold stage, an alert situation requires the enterprise to start escalation procedure.

[1] NIST Glossary [Online]. Available: https://csrc.nist.gov/glossary/term/attack [Accessed: 20-Aug-2020]
[2] NIST Glossary [Online]. Available: https://csrc.nist.gov/glossary/term/Cyber_Attack [Accessed: 20-Aug-2020]
[3] CYBRARY Glossary [Online]. Available: https://www.cybrary.it/glossary/a-the-glossary/attack/ [Accessed: 20-Aug-2020]
[4] HR NCSS Report [Online]. Available: https://euagenda.eu/publications/the-national-cyber-security-strategy-of-the-republic-of-croatia [Accessed: 20-Aug-2020]
[5] NIST Glossary [Online]. Available: https://csrc.nist.gov/glossary/term/alert [Accessed: 20-Aug-2020]
[6] CYBRARY.IT Glossary [Online]. Available: https://www.cybrary.it/glossary/a-the-glossary/alert-situation/ [Accessed: 20-Aug-2020]

Table 5.3 – Survey result for Event concept

Source	Event Definition
NIST[1]	- Any observable occurrence in an information system. - Any observable occurrence in a network or system. - Something that occurs within a system or network. - Any observable occurrence on a manufacturing system. Events can include cybersecurity changes that may have an impact on manufacturing operations (including mission, capabilities, or reputation). - Security Relevant Event[2]: Any event that attempts to change the security state of the system (e.g., change access controls, change the security level of a user, change a user password). Also, any event that attempts to violate the security policy of the system (e.g., too many logon attempts).
CYBRARY.IT[3]	An Event is an action or an occurrence that a program can detect. Examples of some events are clicking of a mouse button or pressing the key, etc.

Table 5.4 – Survey result for Asset concept

Source	Asset Definition
NIST[4]	Resources of value that an organization possesses or employs.
CYBRARY.IT[5]	An Asset is the resources of an organization, business either having tangible value – finance, Infrastructure, physical properties, human resource – or of intangible value such as goodwill that helps business and can be converted to cash for future use

[1] NIST Glossary [Online]. Available: https://csrc.nist.gov/glossary/term/event [Accessed: 20-Aug-2020]
[2] NIST Glossary [Online]. Available: https://csrc.nist.gov/glossary/term/Security_Relevant_Event [Accessed: 20-Aug-2020]
[3] CYBRARY.IT Glossary [Online]. Available: https://www.cybrary.it/glossary/e-the-glossary/event/ [Accessed: 20-Aug-2020]
[4] NIST Glossary [Online]. Available: https://csrc.nist.gov/glossary/term/asset [Accessed: 20-Aug-2020]
[5] CYBRARY.IT Glossary [Online]. Available: https://www.cybrary.it/glossary/a-the-glossary/asset/ [Accessed: 20-Aug-2020]

Table 5.5 – Survey result for Incident concept

Source	Incident Definition
NIST[1]	- An occurrence that actually or potentially jeopardizes the confidentiality, integrity, or availability of an information system or the information system processes, stores, or transmits or that constitutes a violation or imminent threat of violation of security policies, security procedures, or acceptable use policies. - Anomalous or unexpected event, set of events, condition, or situation at any time during the life cycle of a project, product, service, or system. - A violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices. - Cyber incident[2]: Actions taken through the use of an information system or network that result in an actual or potentially adverse effect on an information system, network, and/or the information residing therein.
CYBRARY.IT[3]	An incident is an unplanned disruption or degradation of a network or system service and needs to be resolved immediately. An example of an incident is a server crash that causes a disruption in the business process. However, if the disruption is planned, say, a scheduled maintenance, it is not an incident.
ANSSI[4]	A security incident is an event that affects the availability, confidentiality, or integrity of a property. Examples: Illegal use of a password, theft of computer equipment, intrusion into a file or application, etc.
HR NCSS[5]	Computer security incident: one or more computer security events that have disturbed or are disturbing the security of the information system.
CNCS[6]	An event with a real adverse effect on the security of networks and information systems.

[1] NIST Glossary [Online]. Available: https://csrc.nist.gov/glossary/term/incident [Accessed: 20-Aug-2020]
[2] NIST Glossary [Online]. Available: https://csrc.nist.gov/glossary/term/cyber_incident [Accessed: 20-Aug-2020]
[3] CYBRARY.IT Glossary [Online]. Available: https://www.cybrary.it/glossary/i-the-glossary/incident/ [Accessed: 20-Aug-2020]
[4] ANSSI Glossary [Online]. Available: https://www.ssi.gouv.fr/entreprise/glossaire/i/ [Accessed: 20-Aug-2020]
[5] HR NCSS Report [Online]. Available: https://euagenda.eu/publications/the-national-cyber-security-strategy-of-the-republic-of-croatia [Accessed: 20-Aug-2020]
[6] CNCS Report [Online]. Available: https://www.cncs.gov.pt/content/files/cncs_qnrcs_2019.pdf [Accessed: 20-Aug-2020]

Table 5.6 – Survey result for Vulnerability concept

Source	Vulnerability Definition
NIST[1]	- Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source. - A flaw or weakness in system security procedures, design, implementation, or internal controls that could be exercised (accidentally triggered or intentionally exploited) and result in a security breach or a violation of the system's security policy. - A security exposure in an operating system or other system software or application software component. A variety of organizations maintain publicly accessible databases of vulnerabilities based on the version numbers of software. Each vulnerability can potentially compromise the system or network if exploited. - Software Vulnerability[2]: A security flaw, glitch, or weakness found in software that can be exploited by an attacker.
ANSSI[3]	Faulty, malicious, or clumsy, in the specifications, design, realization, installation or configuration of a system, or in the way of using it. Notes: A vulnerability can be used by an exploit code and lead to an intrusion into the system.
CLUSIT[4]	It represents an intrinsic weakness or due to conditions of exercise or lack of controls, which can be exploited by a threat to cause damage.
CNCS[5]	Weakness of an asset or control that may be exploited by a threat.

[1] NIST Glossary [Online]. Available: https://csrc.nist.gov/glossary/term/vulnerability [Accessed: 20-Aug-2020]
[2] NIST Glossary [Online]. Available: https://csrc.nist.gov/glossary/term/Software_Vulnerability [Accessed: 20-Aug-2020]
[3] ANSSI Glossary [Online]. Available: https://www.ssi.gouv.fr/entreprise/glossaire/v/ [Accessed: 20-Aug-2020]
[4] CLUSIT Report [Online]. Available: https://consapevolmentecloud.clusit.it/_files/Consapevolmente_Cloud.pdf [Accessed: 20-Aug-2020]
[5] CNCS Report [Online]. Available: https://www.cncs.gov.pt/content/files/cncs_qnrcs_2019.pdf [Accessed: 20-Aug-2020]

Table 5.7 – Survey result for Threat concept

Source	Threat Definition
NIST[1]	- An event or condition that has the potential for causing asset loss and the undesirable consequences or impact from such loss. Note: The specific causes of asset loss, and for which the consequences of asset loss are assessed, can arise from a variety of conditions and events related to adversity, typically referred to as disruptions, hazards, or threats. Regardless of the specific term used, the basis of asset loss constitutes all forms of intentional, unintentional, accidental, incidental, misuse, abuse, error, weakness, defect, fault, and/or failure events and associated conditions. - A possible danger to a computer system, which may result in the interception, alteration, obstruction, or destruction of computational resources, or other disruption to the system. - Cyber Threat[2]: An event or condition that has the potential for causing asset loss and the undesirable consequences or impact from such loss. Note: The specific causes of asset loss, and for which the consequences of asset loss are assessed, can arise from a variety of conditions and events related to adversity, typically referred to as disruptions, hazards, or threats. Regardless of the specific term used, the basis of asset loss constitutes all forms of intentional, unintentional, accidental, incidental, misuse, abuse, error, weakness, defect, fault, and/or failure events and associated conditions.
CYBRARY.IT[3]	A threat is a possible danger that might exploit a vulnerability to violate security protocols and thus, cause possible harm. A threat can be either deliberate (example, an individual cracker or a criminal organization) or accidental (example, the possibility of a computer malfunctioning, or the possibility of a natural disaster such as an earthquake, a fire, or a tornado) or otherwise a circumstance, capability, action, or event.
IT NCSS[4]	We define the cyber threat as the complex malicious conducts that can be exercised in and throughout cyberspace, or against cyberspace and its fundamental elements.
CLUSIT[1]	The threat is defined as an event of malicious or accidental nature which, by

[1] NIST Glossary [Online]. Available: https://csrc.nist.gov/glossary/term/threat [Accessed: 20-Aug-2020]
[2] NIST Glossary [Online]. Available: https://csrc.nist.gov/glossary/term/Cyber_Threat [Accessed: 20-Aug-2020]
[3] CYBRARY.IT Glossary [Online]. Available: https://www.cybrary.it/glossary/t-the-glossary/threat/ [Accessed: 20-Aug-2020]
[4] Italian Cybersecurity Action Plan [Online]. Available: https://www.enisa.europa.eu/topics/national-cyber-security-strategies/ncss-map/national-cyber-security-strategies-interactive-map/strategies/national-strategic-framework-for-cyberspace-security [Accessed: 20-Aug-2020]

Source	Threat Definition
	exploiting a vulnerability of the system, could cause damage.
CNCS[2]	Potential cause of an unwanted incident that could cause damage to a system, individual or organization.

After completing the survey and discussing the results with business partners, a set of focus concepts were made which includes: Alert, Asset, Event, and Incident. Then, a consensus definition for these concepts was debated and agreed upon. The consensus definition is provided in Table 5.8.

Table 5.8 – Initial concepts definition

Concept	Definition
Alert	A notification that a specific attack has been directed at an organization's information systems.
Asset	Information or resource which have value to an organization or person.
Event	A discrete change of state or status of an Asset or group of Assets. Specific Events may trigger Alerts.
Incident	An Event (or group of Events) that compromises an Asset. An Incident may be retroactively classified as an attack. Additionally, it has some sort of impact within the organization, which is described by its severity and completion level.

[1] CLUSIT Report [Online]. Available: https://consapevolmentecloud.clusit.it/_files/Consapevolmente_Cloud.pdf [Accessed: 20-Aug-2020]
[2] CNCS Report [Online]. Available: https://www.cncs.gov.pt/content/files/cncs_qnrcs_2019.pdf [Accessed: 20-Aug-2020]

5.2 Ontology Requirements

Ontology Requirements Specification (ORS) (Suárez-figueroa et al. 2009) is the process of defining the requirements that should be covered by the intended ontology, which is inspired from the known Software Requirement Specification. This is important to clarify the reason the ontology is being developed, how it is going to be used, who will use it, and which requirements are supposed to be fulfilled. The process starts with a set of ontological needs according to the targeted business domain, and results in an Ontology Requirement Specification Document (ORSD). ORSD will be useful in obtaining agreement between involved parties and verifying the ontology along the development process. This process needs the cooperation of software developers, ontology practitioners, users, and domain experts. Moreover, it should be performed at the start of the ontology project and continues along with knowledge acquisition. The ontology specifications for this work is presented in Table 5.9.

Table 5.9 – Ontology requirements specification

Integrated Airport Cybersecurity Ontology Requirements Specification	
1	Purpose
	The purpose of building this integrated ontology is to enhance the interoperability among several security systems and solutions that can be used by airports.
2	Scope
	This ontology must focus on the cybersecurity domain applications such as SIEM, SCADA, IDS etc. Physical security aspects can be included such as protection of personnel, hardware, software, networks and data from fire, flood, natural disasters, burglary, theft, vandalism, and terrorism. Physical security standard systems like access control and surveillance can be covered.
3	Implementation Language
	The ontology must be implemented in OWL language.
4	Intended End-Users
	User1. Airport Organizational Personnel that are responsible for airport internal systems and infrastructure; User2. Airport Service Personnel that responsible for the airport supply chain and support services operated and maintained by third party suppliers; User3. External Security Organizations that are responsible for general security vulnerability and threat information; User4. Governmental Organizations and EU partner countries that want to analyze the statistics and evaluate the security state of their airports.
5	Intended Uses
	Use1. Define and update assets to be monitored and secured; Use2. Define and update possible vulnerabilities, their associated assets, consequences, and countermeasures; Use3. Define and update the known threat taxonomy; Use4. Detect weak points in the defined infrastructure; Use5. Detect attack pattern from the monitored events.
6	Ontology Requirements
	a. Non-Functional Requirements
	NFR1. The ontology must be implemented in English; NFR2. The ontology must use a unique prefix to distinguish the added entities from the

	imported ones; NFR3. The ontology must provide easy-to-read labels along the entity's URI and description
	b. Functional Requirements: Groups of Competency Questions
	CQ1: What is the asset? Badge; Passenger; CQ2: What is the category of the asset? Physical Asset; Equipment; Personnel; CQ3: What is the identifier of the asset? 987654321 CQ4: What is the detected location of the asset? Terminal 2; Gate 5 CQ5: What is the incident that occurred to the asset? Attack; Event; CQ6: What is the type of the attack? Unknown Person; Unidentified Baggage Matching; CQ7: What is the attack's additional information? Badge does not match the biometrics; Unidentified baggage matching; CQ8: What is the date and time of the incident? Oct 11 2019 10:00:39; Feb 11 2020 19:50:39 CQ9: What is the criticality of the incident? Normal; Escalation; Emergency; CQ10: What is the triggered alert's severity level? Low; Medium; High; Extreme; CQ11: What is the triggered alert's status? Pending; Ignored; Solved; CQ12: What is the application that reported the incident? User Access Control; Baggage Handling System; CQ13: What is the name of the personnel linked to the badge? John Smith CQ14: Where did the last attack happen? Portugal; Spain; Germany; Greece; CQ15: What is the vulnerability that attack exploited? Expired antivirus software; USB port; CQ16: What countermeasures were taken to cover this vulnerability? Regular updates; Disable USB in unprotected platforms;
7	Pre-Glossary of Terms
	a. Terms of Competency Questions + Frequency

Asset	5	Severity Level	1
Category (type)	2	Status	1
Identifier	1	Application	1
Location (department, country)	2	Personnel	1
Incident (event, attack)	6	Badge	1
Date and Time	1	Vulnerability	2
Criticality	1	Countermeasures	1
Alert	2		

	b. Terms from Answers + Frequency
Badge; Passenger;	1
Normal; Escalation; Emergency;	1
Pending; Ignored; Solved;	1
Physical Asset; Equipment;	1
User Access Control; Baggage Handling System;	1
	c. Objects
	Belgium, Finland, France, Germany, Netherlands, and Portugal.

5.3 Design Progress

The design phase started with the set of focus concepts that were discussed with the business partner as mentioned before. An Incident can refer to one or more Events that occurred during a defined period of time. Each Event can affect an Asset with a certain level of Criticality. An Event can be further categorized as per its Criticality level into Normal, Escalation, or Emergency event. Where a Normal Event does not impact any critical components and does not require intervention. An Escalation Event requires the application of corrective measures. Also, an Emergency Event is one that has impacts to the safety of critical systems.

An Alert gets triggered by the Event with a Severity Level corresponding to the Event's Criticality; how that relationship is defined can be specified by each individual tool issuing the Alert, or it can be inferred through the affected Assets' Criticality. Then, the Alert gets sent to the responsible Audience for processing. An alert can also be categorized as per its Severity level into Low, Medium, High, or Extreme. Here we may introduce some sub-classes that comply with specific practical conditions. Alarm concept can be used to represent a special case of Alert, where the severity level is High or Extreme.

As for time attributes, a time-related ontology named OWL-Time[1] is used to provide representation for beginning and end time instants. This ontology was provided by W3C in 2017 to describe temporal properties and to provide vocabulary related to durations in different format like Gregorian calendar and clock, Unix time, or geologic time among other calendars. The classes and properties in OWL-Time are enough to describe the temporal information needed for this work. Figure 5.1 shows the first version of the ontology's concepts and relations.

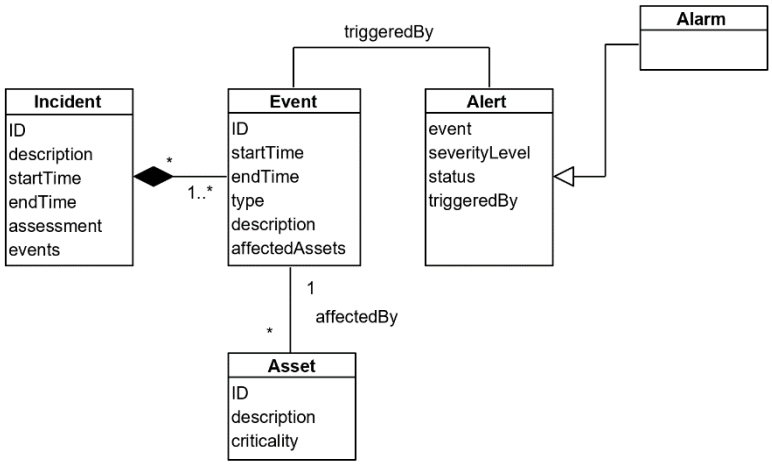

Figure 5.1 – Initial concepts' representation v 1.0 (basic)

[1] "OWL-Time" [Online]. Available: https://www.w3.org/TR/owl-time/ [Accessed: 20-Aug-2020].

Next step is to extend the ontology with other available ontologies to link them together and make use of the existing efforts. First ontology to be extended is UCO ontology where Incident concept is equal to “ucoIncident” concept. Therefore, this concept is used to extend the UCO ontology.

On the other hand, IODEF also includes structure to represent an Incident and it is mappable with UCO. In addition to the Incident class, IODEF provides further description related the Indecent. Like the Assessment of the Incident in terms of its Impact to the Assets. The Impact concepts is used to describe the degree of Incident’s effect on the Asset with the completion attribute. Figure 5.2 shows the updated version of the ontology after extending UCO and IODEF.

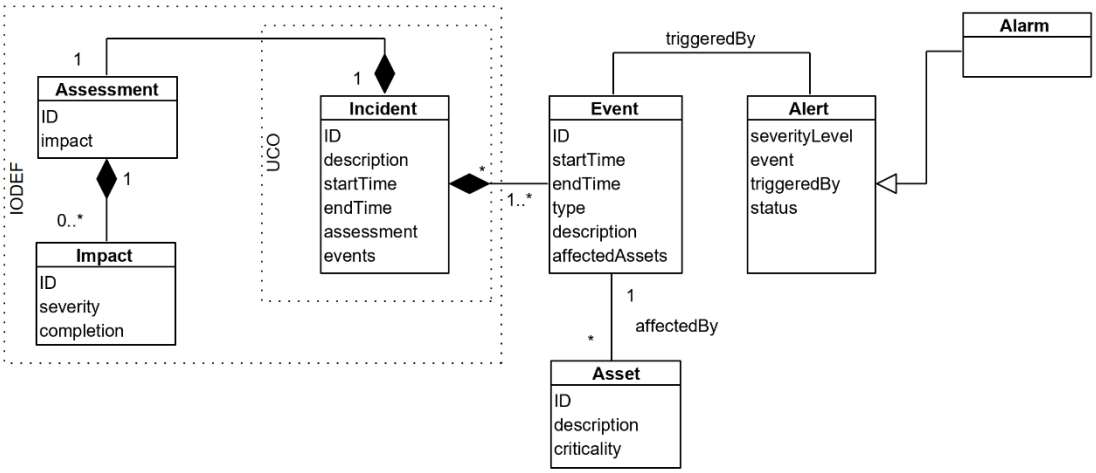

Figure 5.2 – Initial concepts’ representation v 1.1 (UCO and IODEF extension)

Regarding the Alert concept, it was thought that it would be useful to have several sub-classes that can be defined to further fit the domain requirements like Info, Warning, and Advisory. Specific criteria to distinguish these sub-class are to be determined later.

Further extension towards airports’ domain as achieved by linking the ontology to ATMONTO. ATMONTO provides the Engineered System concept that represents several sub-systems like Navigation and Electrical Power systems. These can be used, to an extent, to describe Assets, but are not sufficient to describe existing inventory. Figure 5.3 shows the updated design after adding the addition Alert sub-classes and extending ATMONTO.

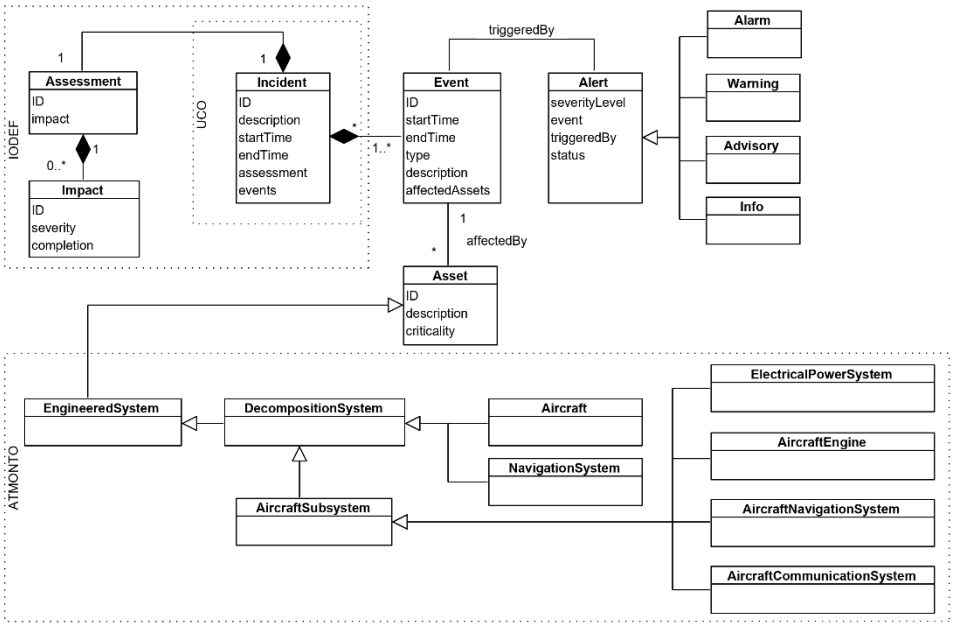

Figure 5.3 – Initial concepts' representation v 1.2 (ATMONTO extension)

After several discussions on the frequently used concepts presented in section 3.4, and specially focusing on SIEM systems' ontologies, we thought it is important to represent Vulnerability, Attack, and Attacker in this ontology. Any Asset may have Vulnerability in different categories that causes Consequences and can be prevented or handled by applying certain Countermeasures. When an Attacker launches an Attack that exploits a Vulnerability, it will be recognized as an Event that is affecting the vulnerable Asset as described above in Figure 5.2. Figure 5.4 shows the representation of the initial concepts along with the extension with other ontologies.

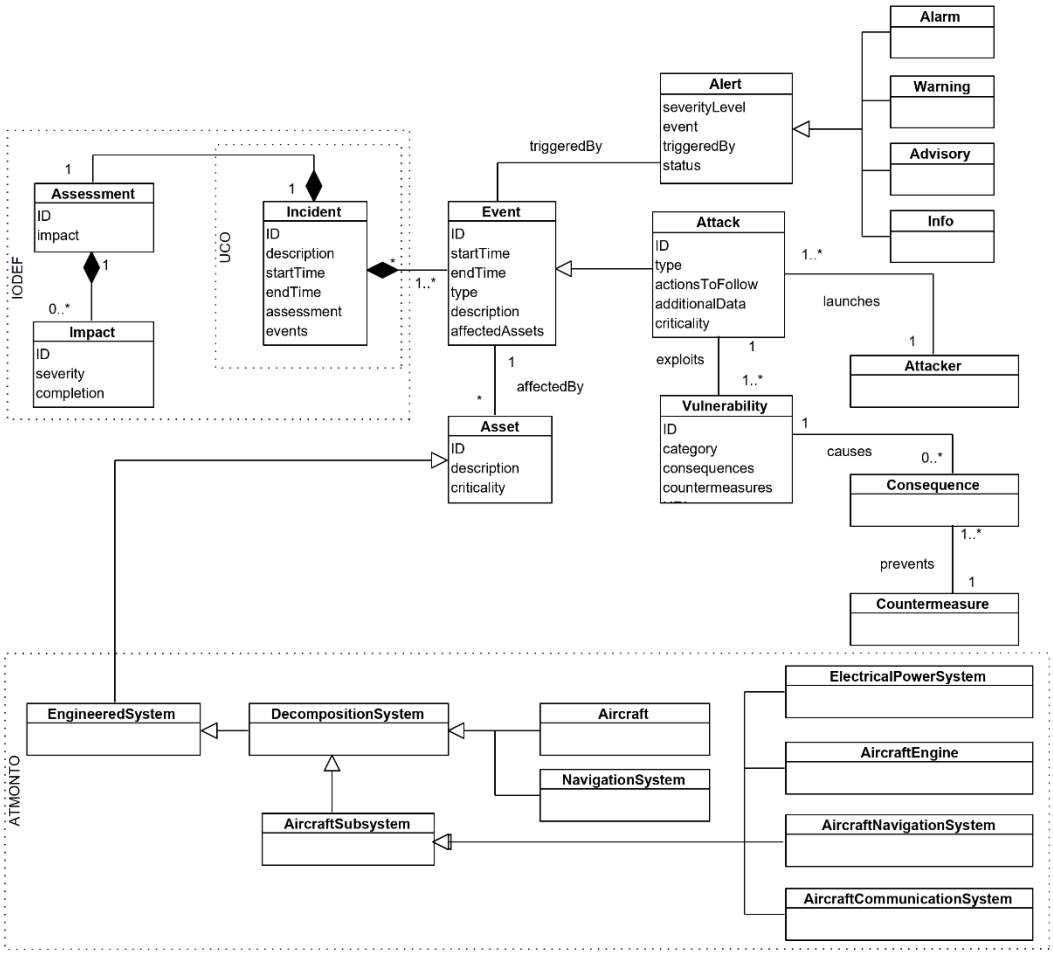

Figure 5.4 – Initial concepts' representation v 1.3 (extended)

A Vulnerability is known to affect a particular SoftwareVersion or Configuration, which are installed in specific Assets. A Vulnerability Exposure is the Event through which a new Vulnerability is discovered and added to the system. However, a Vulnerability may be known but not necessarily be an issue. That is as long it affects Configurations that are not installed on any specific Assets or, at least, not on those with high criticalities. An Event that exploits a known Vulnerability may be retroactively reclassified as an Attack. It is worth noting that the property CVE_ID in the Vulnerability refers to the specific CVE's ID in the cases the Vulnerability has been identified by existing tools; URL points to the online description of this Vulnerability; and Score, as indicated by its name, represents its possible threat/priority level in a scale of 1 to 10. Figure 5.5 focuses on the relationships between Vulnerability, Asset, Event, and Attack.

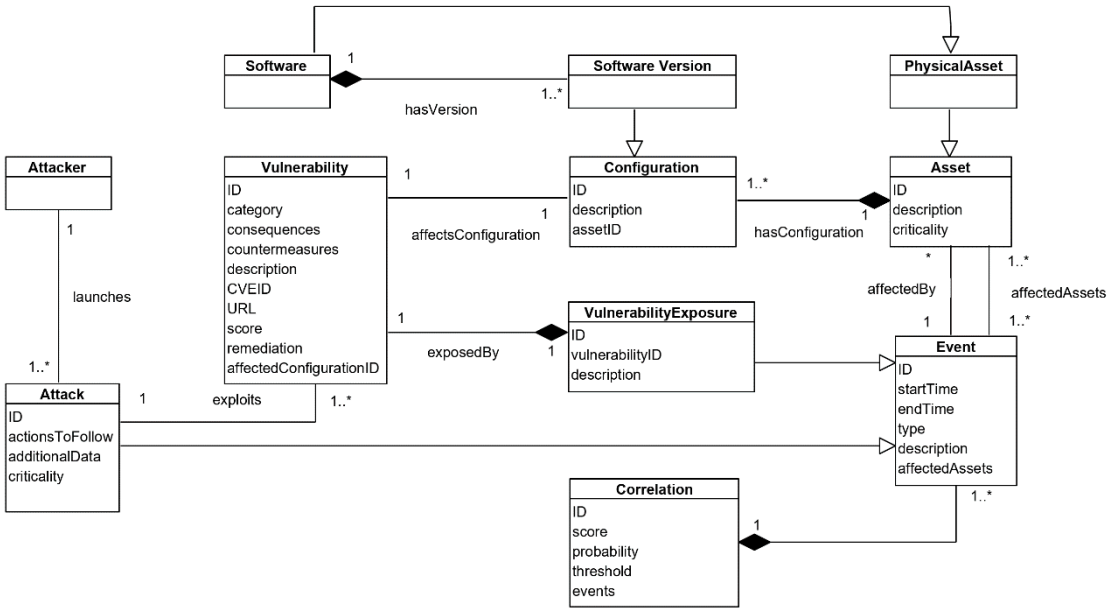

Figure 5.5 – Relationships between Vulnerability, Asset, Event and Attack v 1.4

This version also includes various examples of assets that are considered important for the general security domain. These assets can be organized into four main categories: Logical Assets, Physical Assets, Personnel, and Data or Documents. More airport-specific assets can be added if necessary. The hierarchy of assets in the ontology is shown in Appendix A – Asset Hierarchy. It is worth mentioning that there are more airport-related assets in ATMONTO[1] that were not added directly here. Such assets include Airport, Gate, Terminal, Physical Runway, Operational Runway, ATCT, and Crew Member.

Within the SATIE scenarios, Incidents are generated exclusively through manual means by an operator. The operator goes through a list of Events, evaluates their Correlation, and determines whether these are related and should be considered an Incident. After this assessment, it is possibly to query existing tools about what the Incident's estimated Impact, describing how the Performance of Assets may be affected, how Assets affect each other and suggesting possible Mitigation Strategies, each with their own expected performances, as shown in Figure 5.6. How different Events and Assets may affect each other is described by the ThreatPropagationPath and ThreatPropagationEvent concepts. Through reasoning, it is possible to automatically assess which Assets are affected by a given Incident, although this list may not be exhaustive. The operator, through the analysis of impact assessment tools, may add more Assets to this list. Assessment and Impact were merged into the Impact concepts to simplify the relationships within the ontology without affecting the functionality.

[1] ATMONTO classes [Online]. Available: https://data.nasa.gov/ontologies/atmontoCore/doc/ [Accessed: 08-Sept-2020].

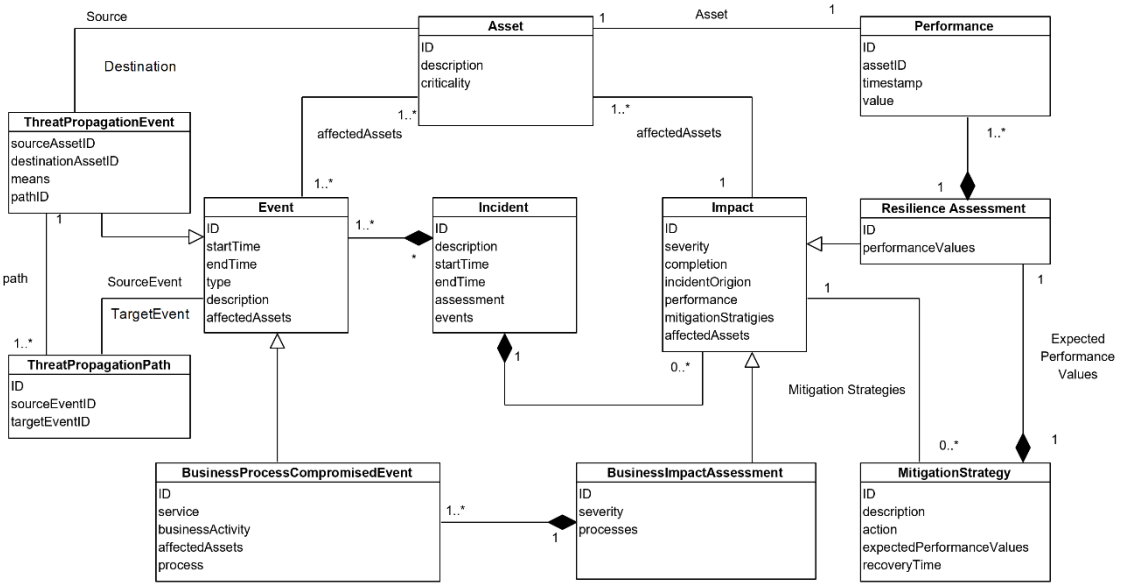

Figure 5.6 – Relationships between Incidents, Impact and Assessment v 1.5

Figure 5.7 shows an overall view of the ontology at v 1.5, where assets' hierarchy and attributes of the concepts were hidden for the simplicity of the diagram.

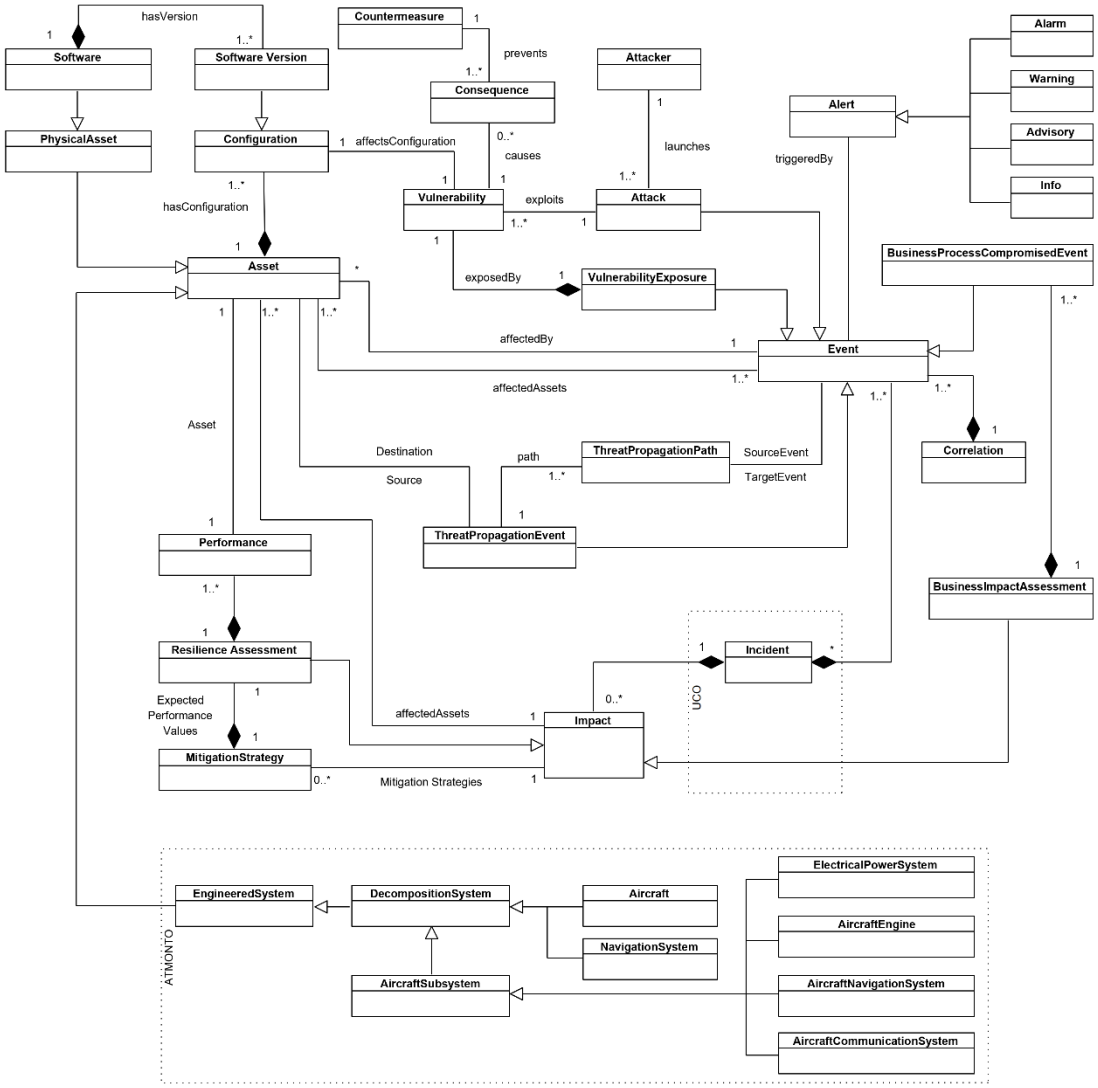

Figure 5.7 – Overall view of the ontology v 1.5

In order to enrich the ontology with attack-related categorization, AttackType was added with sub-classes inspired from (Obrst et al. 2014)(Zhao et al. 2018). Appendix B – Attack Type Hierarchy shows the hierarchy of the AttackType concept and its sub-classes. This would help more accurate identification of the attacks and, therefore, the corresponding actions.

One of the main debated issues with SATIE business partners is the physical focus. How is the inventory going to be better represented along other logical aspects in the ontology? It was later decided to expand even more on the physical assets in order to cover more of the interest areas. This expansion included physical security related systems like Fire Alarm, Closed-circuit television (CCTV), and Access Control. This version of the ontology also includes the expansion of hardware sensors. The detailed hierarchy of sensors would help with more accurate inventory of the equipment. Extended sensors include Temperature Sensors, Smoke Sensors, and IR Sensors. Appendix A – Asset Hierarchy shows the added assets related to sensors and physical security.

Further physical coverage was achieved by including two more ontologies with different focus points. The first ontology is Ticket Ontology which is "a Web vocabulary for describing tickets for concerts, locations like museums, or any kind of transportation for e-commerce"[1]. It provides example usage scenarios for public transportation and airfare. Some of the interesting classes include Ticket, POI, Scope of Access, Seating Layout, and Transportation Service. The other ontology is being developed by ICARUS project as part of European Union's Horizon 2020 research and innovation programme[2]. ICARUS ontology introduces classes like Baggage, Baggage Belt, and Passenger and it is available online[3]. For the extension, the Asset class got a new sub-class 'ICARUS Entities' to be defined equal to ICARUS's 'ICARUS Extra Entities'. Figure 5.8 shows simplified overview of the ontology along with the extended ontologies.

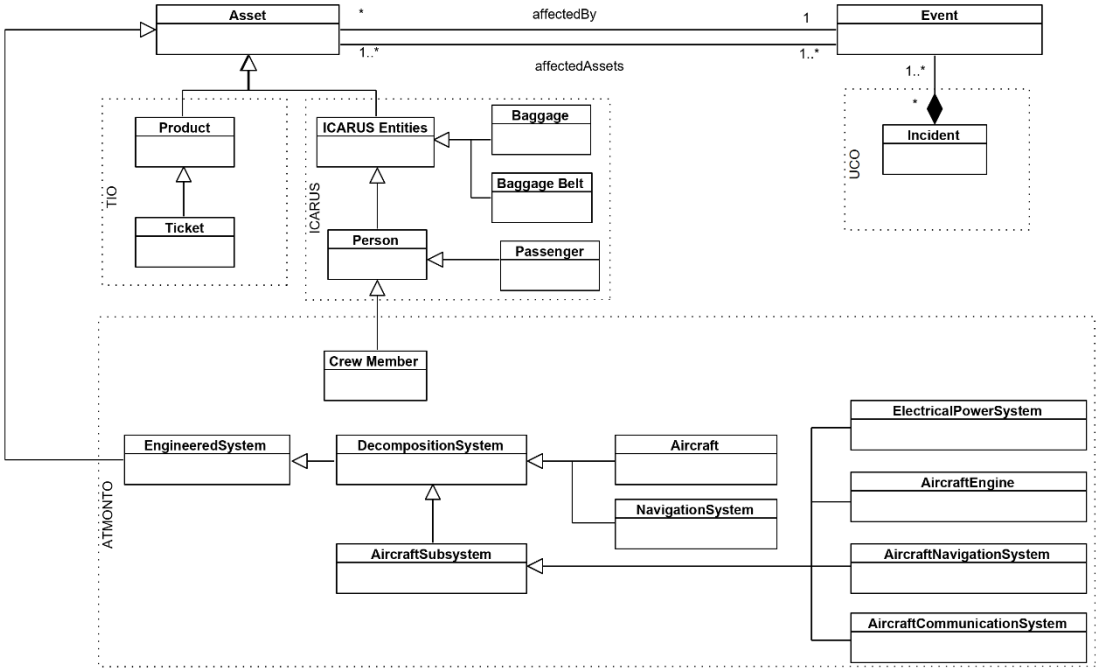

Figure 5.8 – High level view with the extended ontology

[1] The Ticket Ontology [Online]. Available: http://www.heppnetz.de/ontologies/tio/ns [Accessed: 08-Sept-2020].
[2] ICARUS project [Online]. Available: https://www.icarus2020.aero [accessed: 08-Sept-2020]
[3] ICARUS ontology on GitHub [Online]. Available: https://github.com/UCY-LINC-LAB/icarus-ontology [Accessed: 08-Sept-2020]

5.4 Description Logic

In this section, the formal representation of some major points in the ontology will be presented using Description Logic syntax. Concepts and relations from the extended ontologies are noted by the format: “ontology#item”.

$$ASIIOThing \sqsubseteq OWL{:}Thing \tag{1}$$

$$\left\{\begin{array}{c} Alert, Asset, AttackType, Attacker, Configuration, \\ Consequence, Correlation, Countermeasure, Event, \\ Impact,\ Incident,\ MitigationStrategey, Performance, \\ ThreatPropagationPath,\ ValuePartition,\ Vulneratbility \end{array}\right\} \sqsubseteq ASIIOThing \tag{2}$$

$$\begin{array}{c} Alert \sqcap Asset \sqcap AttackType \sqcap Attacker \sqcap Configuration \sqcap \\ Consequence \sqcap Correlation \sqcap Countermeasure \sqcap Event \sqcap \\ Impact \sqcap Incident \sqcap MitigationStrategey \sqcap Performance \sqcap \\ ThreatPropagationPath \sqcap ValuePartition \sqcap Vulneratbility \end{array} = \bot \tag{3}$$

$$\{Severity, Status, Criticality\} \sqsubseteq ValuePartition \tag{4}$$

$$Severity \equiv \{Low, Medium, High, Extreme\} \tag{5}$$

$$Low \sqcap Medium \sqcap High \sqcap Extreme = \bot \tag{6}$$

$$Stauts \equiv \{Enabled, Disabled\} \tag{7}$$

$$Enabled \sqcap Disabled = \bot \tag{8}$$

$$Criticality \equiv \{Normal, Emergency, Escalation\} \tag{9}$$

$$Normal \sqcap Emergency \sqcap Escalation = \bot \tag{10}$$

$$\{Advisory, Alarm, Info, Warning\} \sqsubseteq Alert \tag{11}$$

$$Advisory \sqcap Alarm \sqcap Info \sqcap Warning = \bot \tag{12}$$

$$\begin{array}{l} Alert \sqsubseteq \geq 1 hasSeverity.Severity \sqcap \leq 1 hasSeverity.Severity \sqcap \\ \qquad \geq 1 hasStatus.Status \sqcap \leq 1 hasStatus.Status \sqcap \\ \qquad \geq 1 triggeredby.Event \sqcap \leq 1 triggeredby.Event \end{array} \tag{13}$$

$$\top \sqsubseteq \leq 1 triggeredBy \sqcap triggers \equiv triggeredBy^{-} \tag{14}$$

$$Alarm \equiv \left(Alert \sqcap (\forall hasSeverity.Extreme \sqcup \forall hasSeverity.High)\right) \tag{15}$$

$$\left\{\begin{array}{c} Data, EngineeredSystem, ICARUSEntities, LogicalAsset, \\ Personnel, PhysicalAsset, Product \end{array}\right\} \sqsubseteq Asset \tag{16}$$

$$\begin{array}{c} Data \sqcap EngineeredSystem \sqcap ICARUSEntities \sqcap LogicalAsset, \\ Personnel \sqcap PhysicalAsset \sqcap Product \end{array} = \bot \tag{17}$$

$$\begin{array}{l} Asset \sqsubseteq \forall affectedBy.Event \sqcap \forall assetAffectedBy.Impact \\ \qquad \sqcap \forall has.Performance \sqcap \forall hasConfiguration.Configuration \sqcap \\ \qquad \geq 1 hasCriticality.Criticality \sqcap \\ \qquad \leq 1 hasCriticality.Criticality \\ \qquad \sqcap \forall sourceAssetOf.ThreatPropagationEvent \\ \qquad \sqcap \forall destinationAssetOf.ThreatPropagationEvent \end{array} \tag{18}$$

$$affects \equiv affectedBy^{-} \tag{19}$$

$$hasAffectedAsset \equiv assetAffectedBy^{-} \tag{20}$$

$$configurationOf \equiv hasConfiguration^{-} \tag{21}$$

$$EngineeredSystem \equiv ATMONTO\#EngineeredSyetem \quad (22)$$

$$ICARUSEntities \equiv ICARUS\#ICARUSExtraEntities \quad (23)$$

$$Product \equiv TIO\#ProductOrService \quad (24)$$

$$\begin{Bmatrix} APT, Backdoor, BruteForce, DDoS, DoS, IPDetection, \\ Malware, Phishing, PortDetection \end{Bmatrix} \sqsubseteq AttackType \quad (25)$$

$$APT \sqcap Backdoor \sqcap BruteForce \sqcap DDoS \sqcap DoS \sqcap IPDetection \sqcap Phishing \sqcap PortDetection = \bot \quad (26)$$

$$APT \sqcap BruteForce \sqcap DDoS \sqcap DoS \sqcap IPDetection \sqcap Malware \sqcap Phishing \sqcap PortDetection = \bot \quad (27)$$

$$BackdoorTrojan \equiv Backdoor \sqcap (TojanHorse \sqsubseteq Malware) \quad (28)$$

$$\begin{Bmatrix} Attack, BusinessProcessCompromisedEvent, \\ ThreatPropagationEvent, VulnerabilityExposure \end{Bmatrix} \sqsubseteq Event \quad (29)$$

$$\begin{aligned} Event \sqsubseteq \forall triggers.Alert \sqcap > 1affects.Asset \sqcap \geq 1hasCriticality.Criticality \sqcap \\ \leq 1hasCriticality.Criticality \sqcap \\ \geq 1OWLTime\#'has\ beginning'.OWLTime\#'Time\ instant' \sqcap \\ \leq 1OWLTime\#'has\ beginning'.^{OWLTime\#'}Time\ instant' \sqcap \\ \geq 1OWLTime\#'has\ end'.OWLTime\#'Time\ instant' \sqcap \\ \leq 1OWLTime\#'has\ end'.OWLTime\#'Time\ instant' \\ \sqcap \forall sourceEventOf.ThreatPropagationPath \\ \sqcap \forall destinationEventOf.ThreatPropagationPath \end{aligned} \quad (30)$$

$$Correlation \equiv > 1consistsOf.Event \quad (31)$$

$$\begin{aligned} Attack \equiv Event \sqcap \forall launchedBy.Attacker \sqcap \forall has.AttackType \\ \sqcap exploits.Vulnerability \end{aligned} \quad (32)$$

$$launches \equiv launchedBy^{-} \quad (33)$$

$$Attacker \sqsubseteq > 1launches.Attack \quad (34)$$

$$\begin{aligned} Impact \equiv \forall has.MitigationStrategy \sqcap > 1hasAffectedAsset.Asset \sqcap \\ \geq 1hasSeverity.Severity \sqcap \leq 1hasSeverity.Severity \end{aligned} \quad (35)$$

$$\{BusinessImpactAssessment, ResilienceAssessment\} \sqsubseteq Impact \quad (36)$$

$$\begin{aligned} Incident \equiv > 1consistsOf.Event \sqcap \forall consistsOf.Impact \sqcap \\ \geq 1OWLTime\#'has\ beginning'.OWLTime\#'Time\ instant' \sqcap \\ \leq 1OWLTime\#'has\ beginning'.OWLTime\#'Time\ instant' \sqcap \\ \geq 1OWLTime\#'has\ end'.OWLTime\#'Time\ instant' \sqcap \\ \leq 1OWLTime\#'has\ end'.OWLTime\#'Time\ instant' \end{aligned}$$

$$Incident \equiv UCO\#ucoIncident \quad (37)$$

$$\begin{aligned} Vulnerability \sqsubseteq \forall \text{causes.Consequence} \sqcap \forall \text{exploitedBy.Attack} \sqcap \\ \geq 1\text{exposedBy.VulnerabilityExposure} \sqcap \\ \leq 1\text{exposedBy.VulnerabilityExposure} \sqcap \\ \geq 1\text{affectsConfiguration.Configuration} \sqcap \\ \leq 1\text{affectsConfiguration.Configuration} \end{aligned} \quad (38)$$

$$Consequence \sqsubseteq \forall causedBy.Vulnerability \sqcap \forall preventedBy.Countermeasure \quad (39)$$

$$causes \equiv causedBy^{-} \quad (40)$$

5.5 Summary

As a result of the design process, the domain concept set previously studied in chapter 3 was extended. We added concepts we thought would be useful towards the cybersecurity interoperability goal in general, and of the SATIE project in particular. The updated domain concepts set includes 513 concepts, and Figure 5.9 shows the updated coverage percentage in descending order. The developed ontology as designed so far holds the highest coverage rate of 29.82%. While keeping in consideration that this is a core ontology to be expand later, it is expected to cover even more in future work.

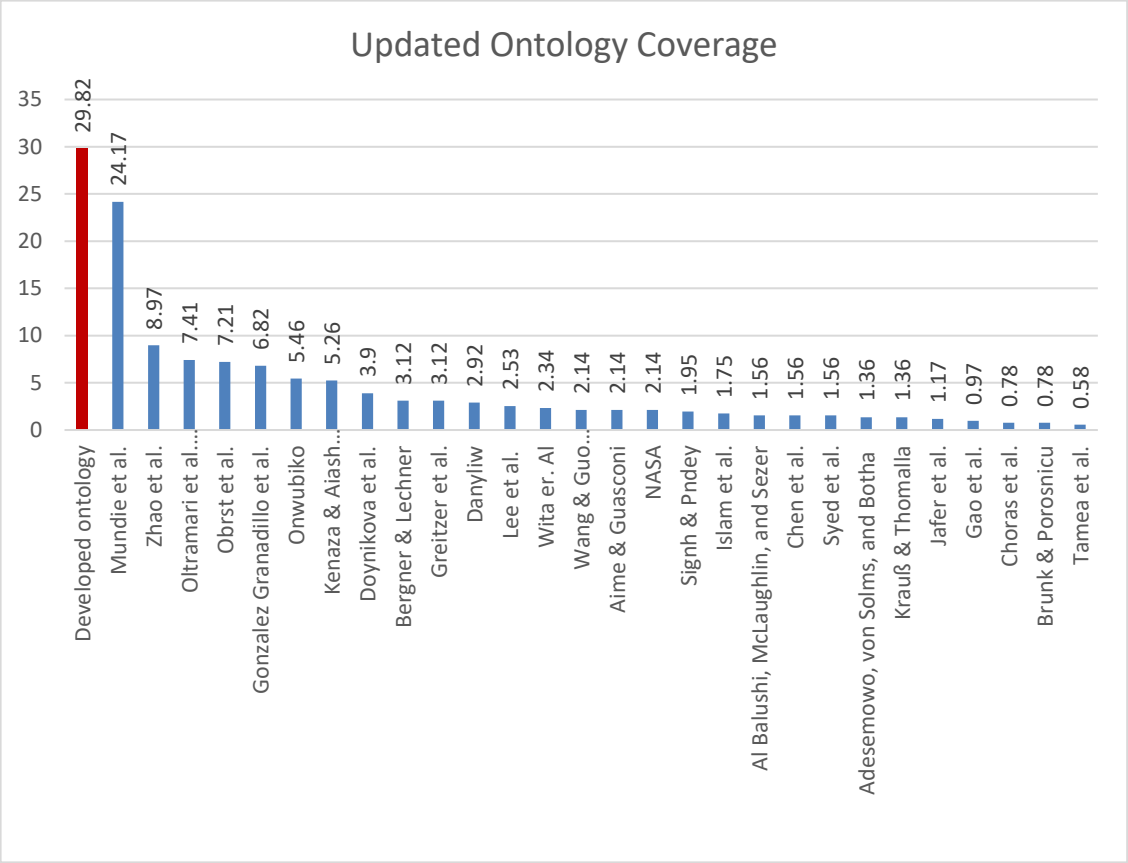

Figure 5.9 – Updated ontologies coverage rate

6 Implementation

The developed ontology was named as Airport Security Interoperability Integrated Ontology (ASIIO). This chapter will show the detailed implementation of the ontology using Protégé which was selected in section 2.4. It will include the classes representing concepts, object properties representing relations, and data properties representing concepts' attributes. General metrics related to ASIIO will be presented in the summary.

6.1 Classes

In this section, the main concepts' classes will be presented after implementing them along with their descriptive logic specification. A full list of the classes with their definition is provided in Appendix C – List of Concepts.

- ***ASIIOThing:*** The implementation of ASIIO starts with a local ASIIOThing class that is a direct subclass of 'OWL:Thing' class. This local class would be the parent of all classes in this ontology in order to group and organize the related concepts and sperate them from other ontologies introduced later. The description of ASIIOThing class is presented below, and Figure 6.1 shows the first level of the classes' hierarchy.
- ***Alert:*** Alert concept was defined, as per the consensus in Table 5.8, as "a notification that a specific attack has been directed at an organization's information systems". Sub-classes were defined for Alert which are Advisory, Alarm, Info, and Warning. Figure 6.2 shows the Alert class and its sub-classes in the class hierarchy.

  As for ***Alarm*** class, it is a specific type of Alert that comply with the defined severity constraints mentioned in section 5.3, having severity level High or Extreme. Figure 6.3 shows the constraint applied to the class definition.

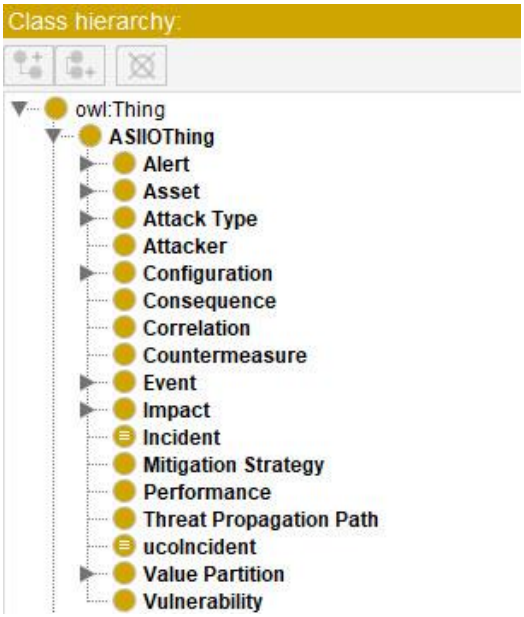

Figure 6.1 – Classes overview

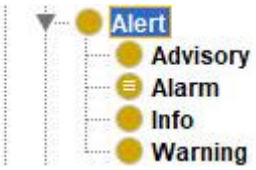

Figure 6.2 – Alert class

Figure 6.3 – Alarm class

- ***Asset***: Asset concept was also defined in Table 5.8 "as the information or resource which has value to an organization or person". Sub-classes for different assets' categories were defined, they are: Data/Document, Engineered System, 'ICARUS Entities', Logical Asset, Personnel, Physical Asset, and Product. **Engineered System** class was added as an extension point with ATMONTO assets. '**ICARUS Entities'** is equal to the ICARUS 'ICARUS Extra Entities' class and represent the extension point with that ontology. Also, **Product** class was added to extend the Ticket ontology's 'Product or service'. Figure 6.4 shows the Asset class and the first level of sub-classes in the hierarchy. The figure reflects Data concept's name modification applied after the validation in section 7.4.

- ***Attack Type***: Attack Type concept was defined as a category or classification of the attack and several sub-classes were defined as per section 5.3. Figure 6.5 shows the first level of the Attack Type hierarchy.

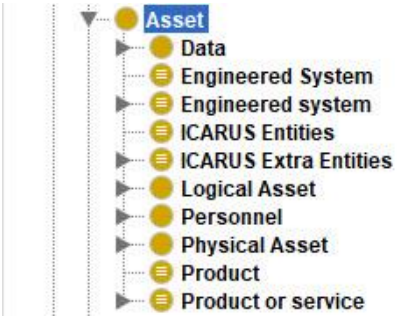

Figure 6.4 – Asset class

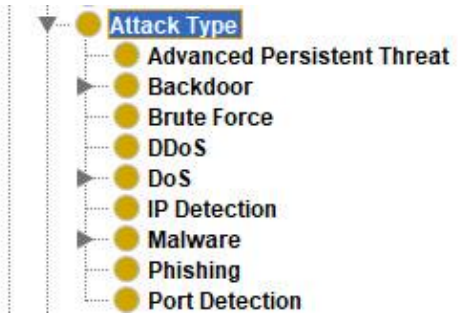

Figure 6.5 – Attack Type class

- ***Event***: In Table 5.8, Event is defined as "a discrete change of state or status of an Asset or group of Assets. Specific Events may trigger Alerts". The sub-classes of Event which are: Attack, Business Process Compromised Event, Threat Propagation Event, and Vulnerability Exposure as shown in Figure 6.6.
- ***Impact***: NIST defines Impact as "the magnitude of harm that can be expected to result from the consequences of unauthorized disclosure of information, unauthorized modification of information, unauthorized destruction of information, or loss of information or information system availability"[1]. The sub-classes of Impact are Business Impact Assessment and Resilience Assessment as shown in Figure 6.7.

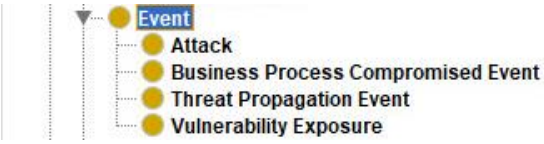

Figure 6.6 – Event class

Figure 6.7 – Impact class

- ***Incident***: Incident is defined in Table 5.8 as "An Event (or group of Events) that compromises an Asset. An Incident may be retroactively classified as an attack. Additionally, it has some sort of impact within the organization, which is described by its severity and completion level". It is equal to the UCO Incident class '**ucoIncident**' and represent the extension point with that ontology.
- ***Value Partition***: is a class created to hold all pre-defined values used with concepts. Value Partitions are not a mandatory part of OWL. They are a 'design pattern' used to model some problems and, in this case, restrict the range of possible values to a pre-defined list. Such value partitions include the following:
  - ***Severity***: Low; Medium; High; Extreme.

[1] NIST Glossary [Online]. Available: https://csrc.nist.gov/glossary/term/countermeasures [Accessed: 20-Aug-2020]

- ***Criticality***: Normal; Escalation; Emergency.
- ***Status***: Enabled; Disabled.

Figure 6.8 shows the implementation of Value Partition class and its sub-classes.

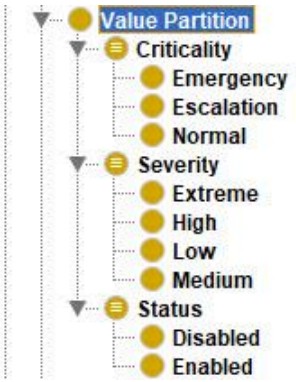

Figure 6.8 – Value Partition class

OWL-Time ontology was imported to Protégé to represent the temporal attributes along with the previously mentioned ATMONTO, UCO, TIO, and ICARUS.

As a result, ASIIO contains 153 core classes including 107 assets and 26 attack types. That is in addition to extending OWL-Time, ATMONTO, and UCO. It is worth mentioning that this count reflects modifications applied after the validation in section 7.4.

6.2 Properties

There are two types of properties in OWL: Object and Data properties. Object properties are used to add some restriction onto the classes and to represent relationships between classes or instances. OWL provides a top-level property 'topObjectProperty' and we will introduce a sub-property for it called 'extTopObjectProperty' to be a parent for all defined object properties within ASIIO. On the other hand, data properties are used to declare characteristics of instances that are simple as data types like numbers or dates. OWL provides a top-level property 'topDataProperty' and we will also introduce a sub-property for it called 'extTopDataProperty' to be a parent for all defined data properties as well.

The value partitions share a simple '***is-a***' relationship to donate inheritance of classes. Figure 6.9 shows the part of ontology representing the value partitions using Protégé plugin OntoGraf[1].

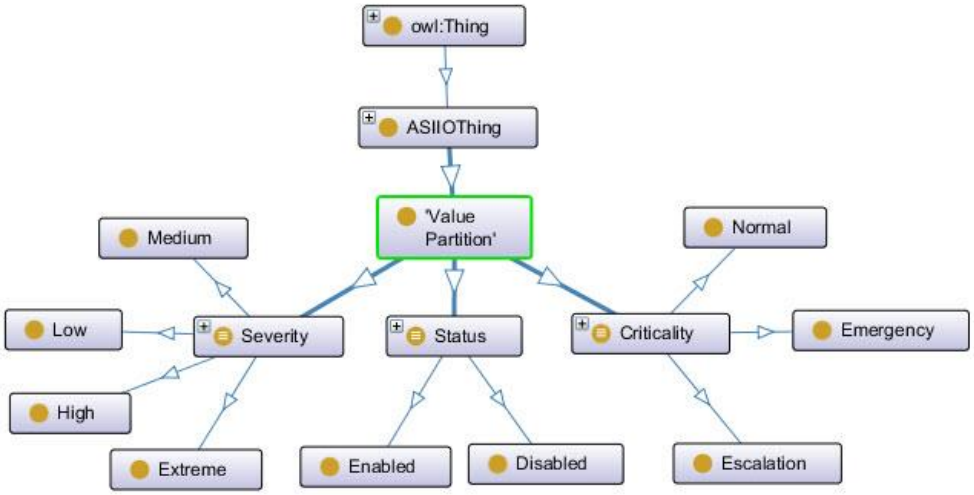

Figure 6.9 – OntoGraf of value partitions

Alert class also shares a '***is-a***' relationship with its sub-class, along with '***hasSeverity***' and '***hasStatus***' relationships with one instance of Severity and Status value partitions, respectively. Alert also has a '***triggeredBy***' relationship with one instance of Event as shown in Figure 6.10.

Figure 6.10 – Alert properties

Event class '***affects***' one instance of Asset and shares a '***hasCriticality***' relationship with one instance of Criticality. An Event '***triggers***' an Alert which is an inverse of the previously mentioned '***triggeredBy***'. To note the start and end time of the event, there are 'has

[1] "OntoGraf: Protégé Wiki" [Online]. Available: https://protegewiki.stanford.edu/wiki/OntoGraf. [Accessed: 20-Aug-2020]

beginning' and 'has end' relationships from OWL-Time ontology to connect the class to the 'Time instant' class. Figure 6.11 depicts the relations of Event class. Vulnerability Exposure shares a '***is-a***' with Event as its sub-class while Correlation '***consistsOf***' one or more Event as shown in Figure 6.12.

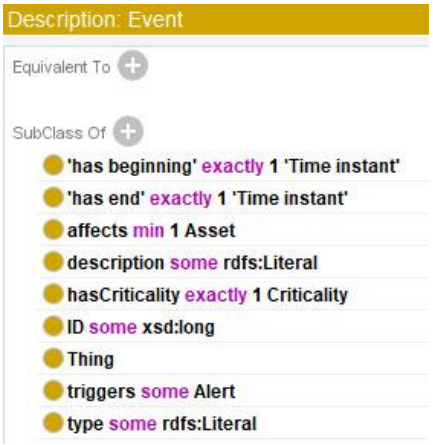

Figure 6.11 – Event properties

Figure 6.12 – Correlation properties

Attack shares an '***is-a***' relationship with Event as its parent class. It also '***exploits***' one or more of Vulnerability instances, and '***has***' an Attack Type. In addition to having a '***hasCriticality***' with one instance of Criticality as shown in Figure 6.13. An Attacker '***launches***' one or more Attacks as shown in Figure 6.14.

Figure 6.13 – Attack properties

Figure 6.14 – Attacker properties

Incident '***consistsOf***' one or more instances of Event and some Impact instances. 'has beginning' and 'has end' relationships link the Incident to 'Time instant' to represent its start and end time. This class is also equal to 'ucoIncident'; therefore, it gets all the relationships from UCO as shown in Figure 6.15.

Vulnerability has an '***affectsConfiguration***' relation with one instance of Configuration class. It '***causes***' some Consequences and is '***exposedBy***' one instance of Vulnerability Exposure event as shown in Figure 6.16. On the other hand, Countermeasure '***prevents***' one or more Consequences as in Figure 6.17.

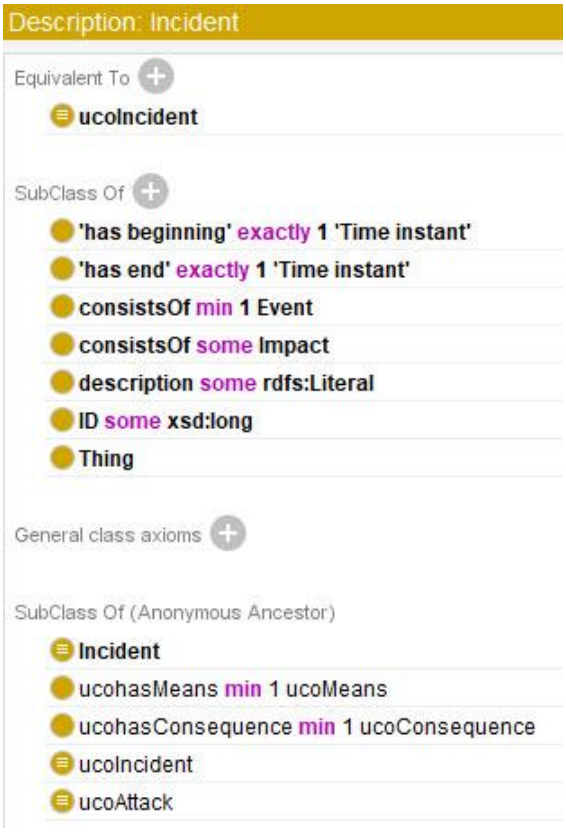

Figure 6.15 – Incident properties

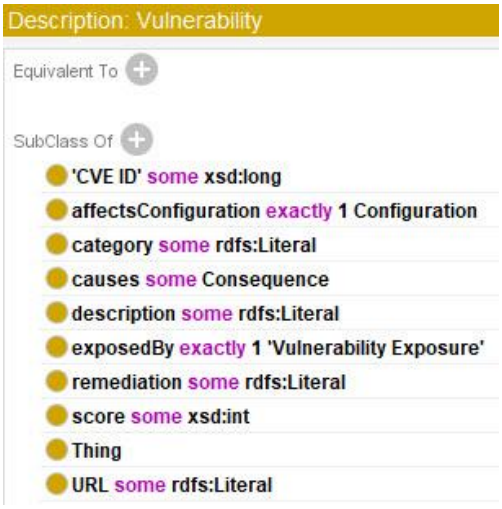

Figure 6.16 – Vulnerability properties

Figure 6.17 – Countermeasure properties

Figure 6.18 shows a focused view on the main concepts described so far.

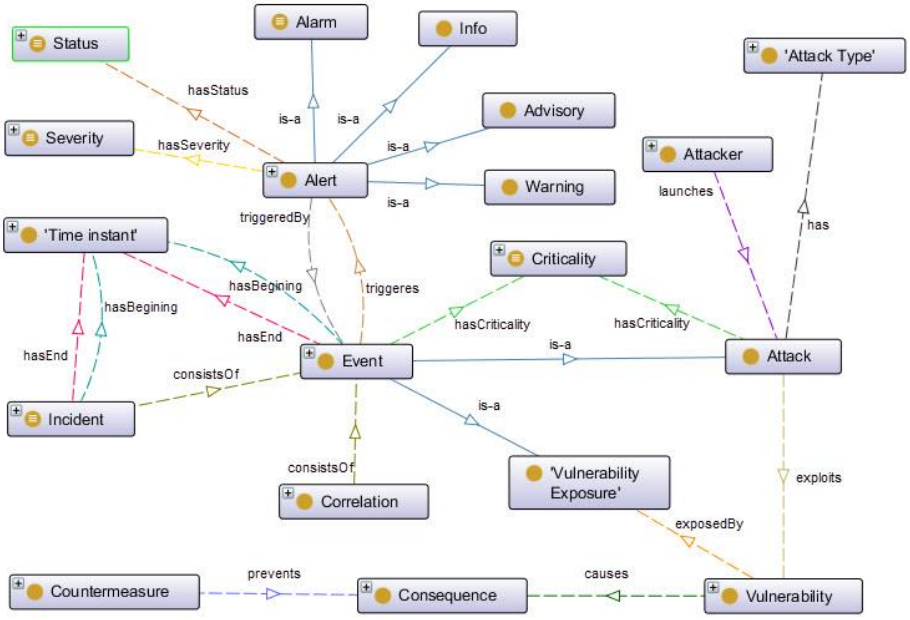

Figure 6.18 – OntoGraf of main concepts

An Asset '***has***' some Vulnerabilities and one instance of Performance. It is '***affectedBy***' an Event, which has an inverse relationship '***affects***'. Also, it shares a '***hasConfiguration***' relationship with one or more instances of Configuration, and a '***hasCriticality***' with one instance of Criticality value partition. Figure 6.19 shows the object properties of the Asset class.

Software is a second level descendant of Asset, as its direct parent is Physical Asset; therefore, it gets all the relationships defined for Asset. It also shares a '***hasVersion***' relationship with at least one instance of Software Version which is a sub-class of Configuration as shown in Figure 6.20.

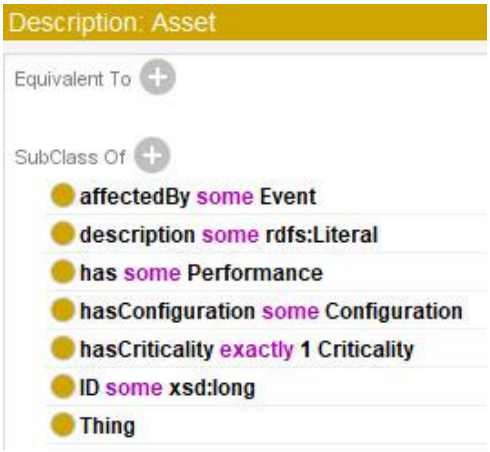

Figure 6.19 – Asset properties

Figure 6.20 – Software properties

Impact shares a '***hasSeverity***' relationship with one instance of Severity and '***hasAffectedAssets***' with one or more Assets. It also '***has***' some Mitigation Strategies as shown in Figure 6.21. Mitigation Strategy '***consistsOf***' one instance of Resilience Assessment as in Figure 6.22.

Figure 6.21 – Impact properties

Figure 6.22 – Mitigation Strategy properties

Resilience Assessment and Business Impact Assessment are sub-classes of Impact. Resilience Assessment '***consistsOf***' at least one instance of Performance which has a link to 'Time instant' via 'has time instant inside'. Figure 6.23 and Figure 6.24 show the relations of Resilience

Assessment and Performance, respectively. Business Impact Assessment '***consistsOf***' at least one Business Process Compromised Evets. In addition to sharing a '***hasSeverity***' with one of Severity instances as shown in Figure 6.25.

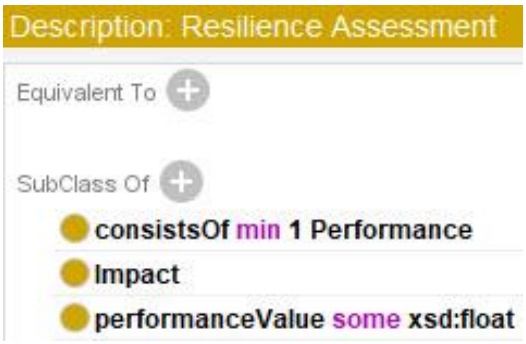

Figure 6.23 – Resilience Assessment properties

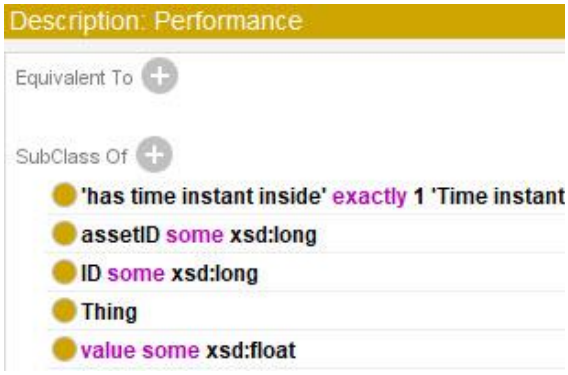

Figure 6.24 – Performance properties

Figure 6.25 – Business Impact Assessment properties

Threat Propagation Event is a sub-class of Event that '***has***' one or more Threat Propagation Path instances. It also '***hasSourceAsset***' and '***hasDestinationAsset***' as shown in Figure 6.26, whereas Threat Propagation Path '***hasSourceEvent***' and '***hasTargetEvent***' as shown in Figure 6.27.

Figure 6.26 – Threat Propagation Event properties

Figure 6.27 – Threat Propagation Path properties

Figure 6.28 shows a focused view on the concepts related to Asset and their relationships.

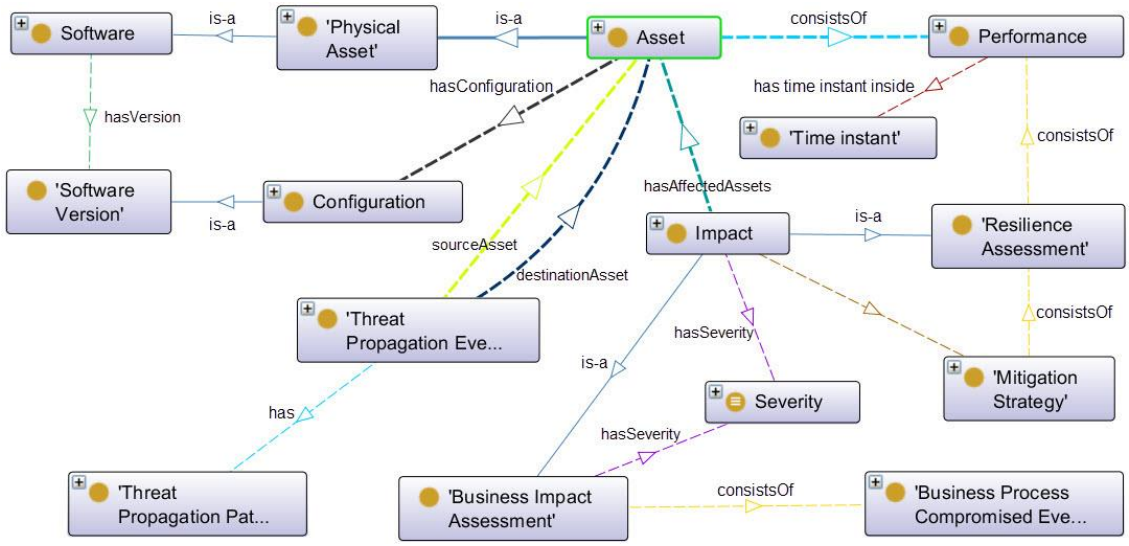

Figure 6.28 – OntoGraf of Asset related concepts

The integrated ontology consists of 40 object properties defined under a local object property 'objectProperty' for organization purposes, including inverse properties added after the validation in section 7.4, as shown in Figure 6.29. While it consists of 23 data properties that are defined under a local property 'dataProperty' as shown in Figure 6.30.

Figure 6.31 shows relationships within ASIIO except for value partitions, attack types, and assets for clarity of the graph.

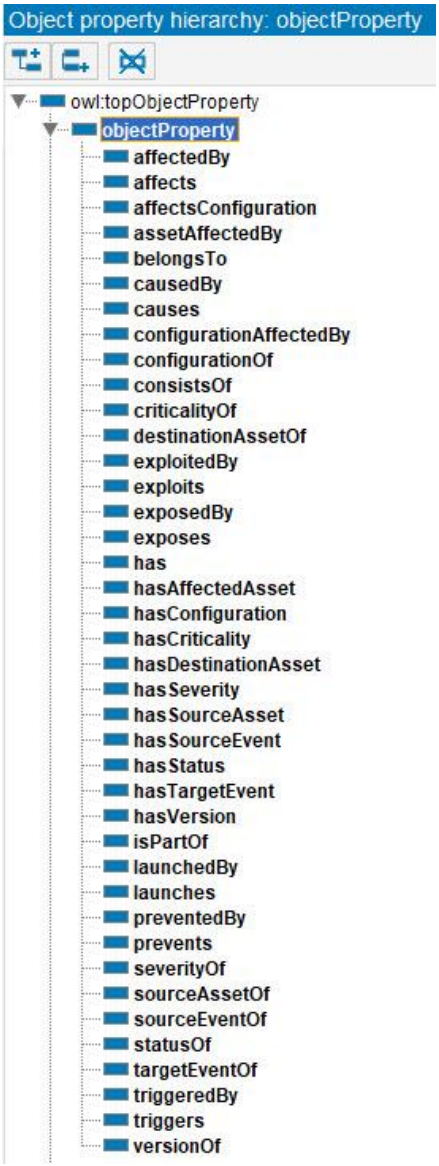

Figure 6.29 – Object properties hierarchy

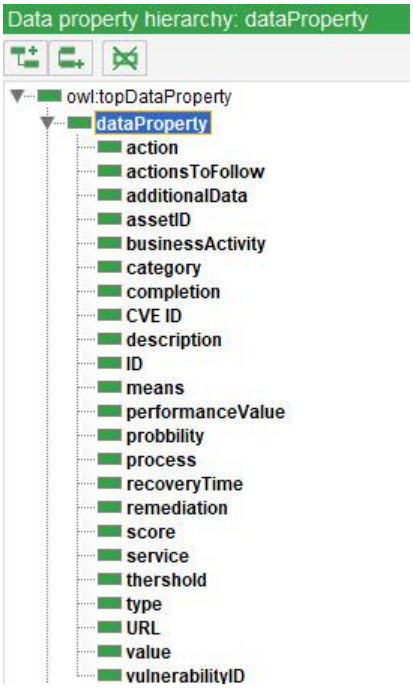

Figure 6.30 – Data properties hierarchy

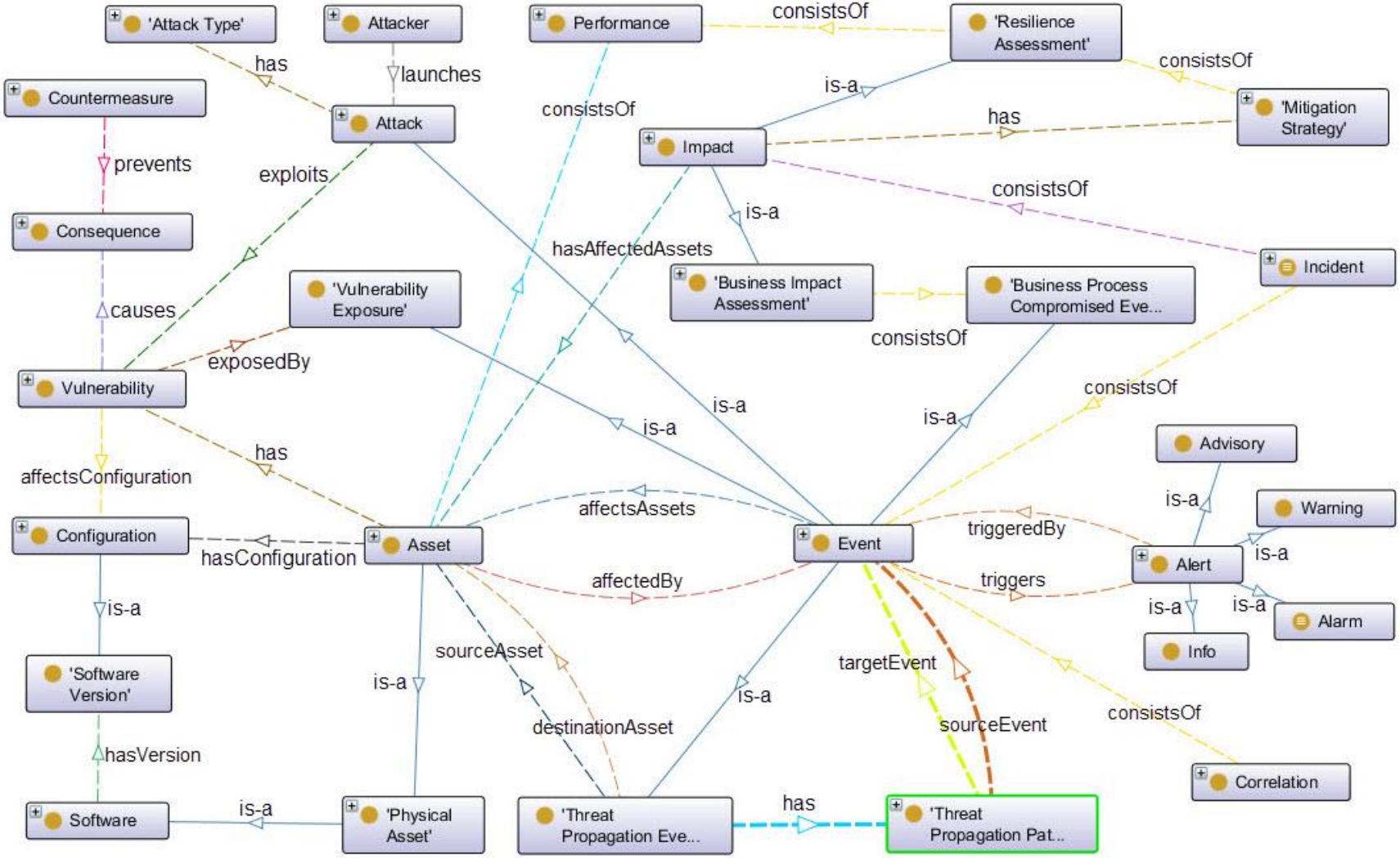

Figure 6.31 – Overall view of ASIIO relations

Figure 6.32 shows a higher level that includes the other imported ontologies with a sample of the concepts available within each one.

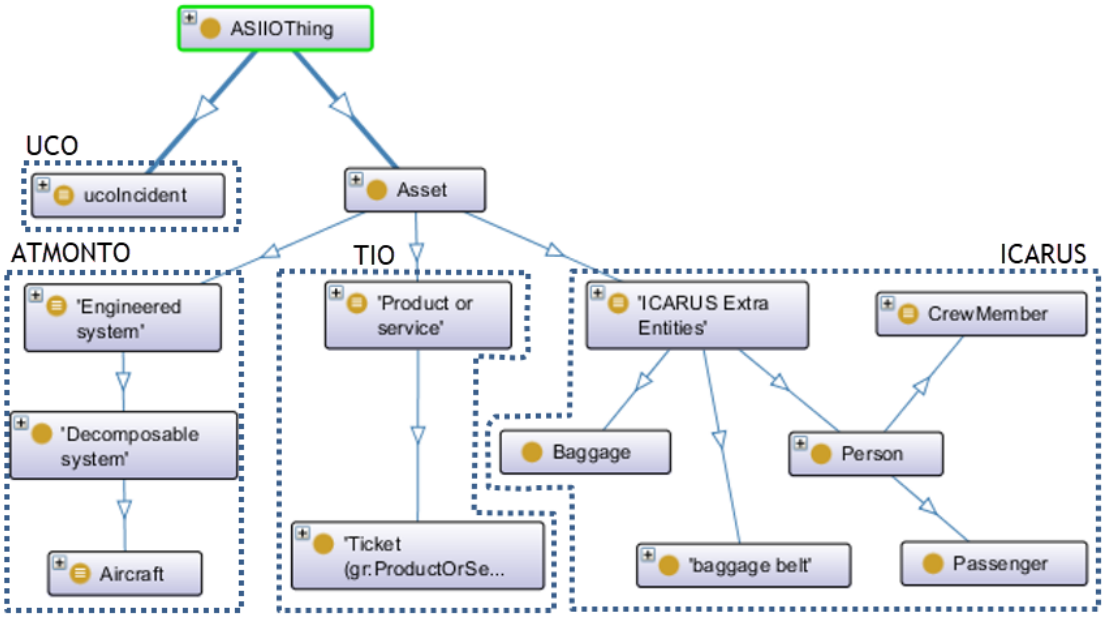

Figure 6.32 – High level OntoGraf

The defined data properties' ranges are from different types: Literal, Long, DateTimeStamp, Int, and Float. Table 6.1 shows the type of each of data properties.

Table 6.1 – Data type of data properties

Literal		Long	DateTimeStamp	Int	Float
action	means	assetID	recoveryTime	score	completion
actionsToFollow	process	CVE ID			performanceValue
additionalData	remediation	ID			probability
businessActivity	service	thershold			value
category	Type	vulnerabilityID			
description	URL				

6.3 Summary

In this chapter, the implementation process was described using Protégé. Main concepts were defined, and a full list of available concepts was made available in the appendices. After implementing ASIIO and importing the extended ontologies, the total count of classes sums up to 575, with 377 object properties and 617 data properties.

Protégé provides several general metrics about the result ontology as shown in Figure 6.33. In this figure, the numbers of class axioms are shown per category: SubClassOf which notes

inheritance, EquivalentClasses which notes similar concepts, and DisjointClasses which notes non-overlapping concepts. These assertions help make the hierarchy between concepts and individuals clear, avoid ambiguity due to different naming, and ensures correct categorization of inferred information.

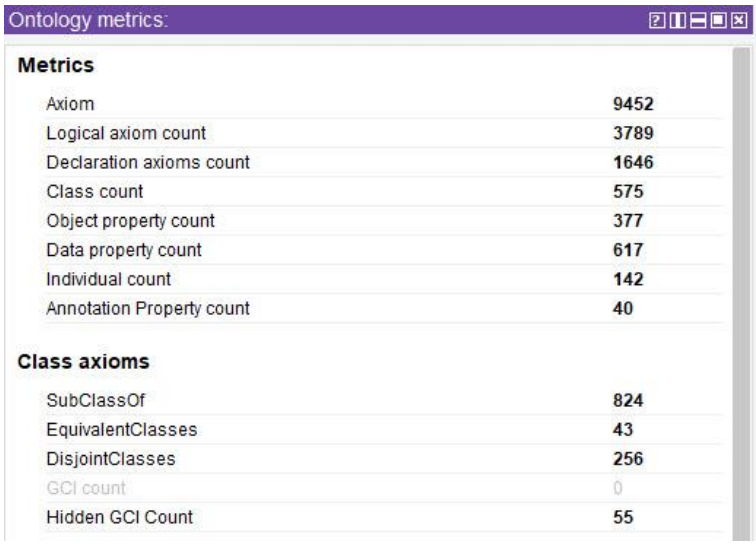

Figure 6.33 – Ontology metrics

7 Evaluation

This chapter will present the details for research hypotheses and the evaluation and experiments that will help prove the success of the developed ontology.

The main proposed methodology of evaluation will follow a task-based approach. Task-based ontology evaluation approaches measure how much the ontology helped with the improvement of the intended task (Raad et al. 2017).

The conducted evaluation includes manual and automated validation, publishing relevant paper about the ontology proposal, analyzing data sample from several systems, and assessing the partners' satisfaction with the developed ontology with an anonymous questionnaire.

7.1 Research Hypothesis

Based on preliminary research findings, different systems use different formats to represent their cybersecurity information. Which makes it harder exchange that knowledge with other systems without an additional work towards mapping and translating between these different formats. The translation process itself might not be successful, as not all content details can be mapped.

According to the research, several questions can be posed, which in turn help to form the corresponding hypotheses:

1) How to avoid ambiguity and confusion due to use of different terminology in different systems in cybersecurity context?
   Hypothesis 1: if using different terminology to refer to the same thing causes confusion and mixed results, composing consensus definitions for the used concepts with a comprehensive semantic layer will reduce the confusion.

2) How to enable systems and solutions to exchange knowledge and interoperate efficiently?

Hypothesis 2: if using different and separate formats causes difficulties in communication and knowledge exchange, using a unified format as in an integrated ontology to represent the overall business requirements will make the process easier.

3) How to make use of multiple systems to extract additional knowledge that exceeds individual systems' boundaries?
   Hypothesis 3: if using an individual system's data limits knowledge extraction to its scope of information, using information shared via the proposed integrated ontology will provide a bigger scope that spans over multiple systems.

7.2 Indicators and Information Source

The indicators that will be used to evaluate the ontology are:

- Number of systems that can interoperate together using the ontology;
- Accuracy in describing some specific scenarios in case studies;
- Efficiency in detecting combined patterns from multiple systems;
- Business partners and concerned entities agreement;
- Scientific community assessment.

This research will acquire information from two types of sources:

1) Public sources: like online publications, ontologies, threat taxonomies, and vulnerability databases;
2) Private sources: basically, the business partners' contributions to the related project. This information cannot be completely disclosed in detail, but some conclusions can be drawn out from it as clear as possible with the approval of the source.

7.3 Proposed Evaluation Methodology

An important assessment of this work will be provided by the scientific community. Conference and journal papers will be prepared and submitted to relevant and recognized events and journals in the area such as Formal Ontology in Information Systems and Advanced Engineering Informatics. The metrics in this case will be, at an early stage, the number of accepted papers and then based on the number of citations.

After the development process is complete and the ontology is on its first stable version, the following evaluation can take place simultaneously.

To prove hypothesis 1, a questionnaire can be sent to concerned parties. That includes business partners as well as external related entities like Francisco Sá Carneiro Airport of Porto. This questionnaire would help get their feedback on the ontology's concepts and relationships.

The main Key Performance Indicator (KPI), to measure the success of the developed ontology in proving hypothesis 2, would be the number of different systems and solutions that become capable of interoperating and communicating easily with each other using the integrated ontology. Keeping in mind that these systems were working separately using different formats for information representation and struggle with the communication with others due to this difference. This would also mean that for any external system, it would be equal to communicating to one compound system rather than several separate ones.

Some case studies for specific scenarios cans be explored and checked against the ontology. Use case diagrams can help depict how the ontology works with different systems.

Other experiments can be conducted to prove hypothesis 3 in detecting combined patterns from different systems' observations. An example of such combined patterns would be patterns that can be extracted from passengers' data and baggage handling system logs. Normally, these systems work separately. However, using the integrated ontology would help extract new knowledge from both of them combined that is not possible to obtain from each one on their own.

7.4 Conducted Evaluation

This section presents several actions which were taken to ensure the correctness and validity of the implemented ontology. These actions include OWL validation, analysis of sample data from relevant systems, and partners' satisfaction assessment.

7.4.1 Manual Validation

OD 101 (Noy et al. 2001), the development methodology selected in section 2.5, provides several guidelines to ensure better result of the developed ontology. This work was manually checked and verified to adhere to them.

- **Ensuring the class hierarchy correctness**

The classes in ASIIO were thoroughly checked to ensure the correctness and the transitivity of the hierarchy, i.e. 'is-a' relations, and the absence of any cycles within the classes. In regards of classes and their synonyms, it was found that some synonyms were defined as sperate classes, such as Trojan and TrojanHorse. This issue was fixed by removing the redundant class and adding the synonym as an 'rdfs:label' in the concept's metadata.

- **Analysing siblings in the class hierarchy**

Regarding the number of sibling classes on a certain level of the ontology, OD101 distinguishes two cases: too little and too many. "If a class has only one direct subclass there may be a modelling problem, or the ontology is not complete. If there are more than a dozen subclasses for a given class then additional intermediate categories may be necessary" (Noy et al. 2001).

ASIIO was scanned looking for these cases and the findings are shown in **Error! Reference source not found.**. Points 1 and 2 are related to attack type hierarchy that was inspired from (Obrst et al. 2014)(Zhao et al. 2018), so there will be no changes to them. Points 3 and 4 will also be kept as-is in anticipation of future expansions, as it is one of the main goals of ASIIO; to be a common base for current and future works. For example, further expansion of ASIIO may include Train and Ship as subclasses of Transportation. But, since it is not our focus now, they are not added in the current work. However, having the general class already available would make later expansion process easier.

Table 7.1 – Analysing sibling classes

Too little (<2)			Too Many (>12)		
1	DoS	<- Tear Drop	5	Thing	<- 16
2	Backdoor	<- Backdoor Trojan	6	Equipment	<- 13
3	Personnel	<- Internal Personnel			
4	Transportation	<- Airplane			

Points 5 and 6 represent the real world categorization which is the meaning of ontologies and OD101 allows it in this case as it is mentioned by (Noy et al. 2001): “However, if no natural classes exist to group concepts in the long list of siblings, there is no need to create artificial classes—just leave the classes the way they are. After all, the ontology is a reflection of the real world, and if no categorization exists in the real world, then the ontology should reflect that”.

- **Disjoint classes**

Disjoint classes mean that their instances do not overlap, there is not an instance that may belong to two disjoint classes. Explicitly declaring this characteristic would assist the tool to validate the populated ontology better for any inconsistencies. Missing the disjoint specification is “the most common modelling error found in OWL ontology” (Rector et al. 2004). ASIIO was reviewed to make sure all the disjoint classes are declared as such. In the developed ontology, all classes on the same level are disjoint with each other, i.e. an Alert cannot be an Event. An example of classes those were not declared as disjoint is Strategic Data sub-classes: Confidential Data, Digital Credential, and Intellectual Property whereas some Digital Credential can be Confidential Data.

- **Naming conventions**

Protégé, the ontology editor selected in section 2.4, is case-sensitive and holds all classes and properties in the same namespace which affected the naming convention used in this work. The followed convention for naming classes in ASIIO is ‘camelCase’ for the Internationalized Resource Identifiers (IRIs) starting with uniform prefix ‘ext’ to group all elements for the ontology. As for the labels, more relaxed convention is followed to enhance readability of the classes’ names with capitalized and spaced words. In the meanwhile, properties maintain the ‘camelCase’ as it is more common with verbs. In general, the prefix is dropped out of all labels for clarity, nouns are in singular form, and no symbols are allowed. The only exception would be abbreviations like URL, CVE ID, etc.

For example: class's IRI is extResilienceAssessment and its label is 'Resilience Assessment', property's IRI is extHasConfiguration and its label is hasConfiguration.

7.4.2 Automated Validation

(Poveda-Villalón 2016) introduced the OntOlogy Pitfall Scanner! (OOPS!) as a tool[1] to detect common pitfalls of ontology development. OOPS! categorizes the outcome into three levels: Minor, Important, and Critical. It uses some lingual analysis, which might lead to false alarms. That is why it is needed to evaluate each point, but it still helps highlight some issues that manual check may overlook.

Figure 7.1**Error! Reference source not found.** shows the result of running ASIIO through OOPS! And it shows 5 minor, and 4 important pitfalls along with 16 suggestions.

Results for P04: Creating unconnected ontology elements.	1 case | Minor
Results for P07: Merging different concepts in the same class.	2 cases | Minor
Results for P08: Missing annotations.	172 cases | Minor
Results for P11: Missing domain or range in properties.	53 cases | Important
Results for P13: Inverse relationships not explicitly declared.	39 cases | Minor
Results for P22: Using different naming conventions in the ontology.	ontology* | Minor
Results for P24: Using recursive definitions.	2 cases | Important
Results for P30: Equivalent classes not explicitly declared.	3 cases | Important
Results for P41: No license declared.	ontology* | Important
SUGGESTION: symmetric or transitive object properties.	16 cases

Figure 7.1 – Initial OOPS! analysis

In this section, the pitfalls found by OOPS! will be closely verified and fixed if possible.

- **Important Pitfalls**

The first pitfall is "P11: Missing domain or range in properties" and Figure 7.2**Error! Reference source not found.** shows a sample of the 53 listed properties corresponding to this point, some of which are from imported ontologies. OOPS! notes that a property needs a defined domain and range to be complete.

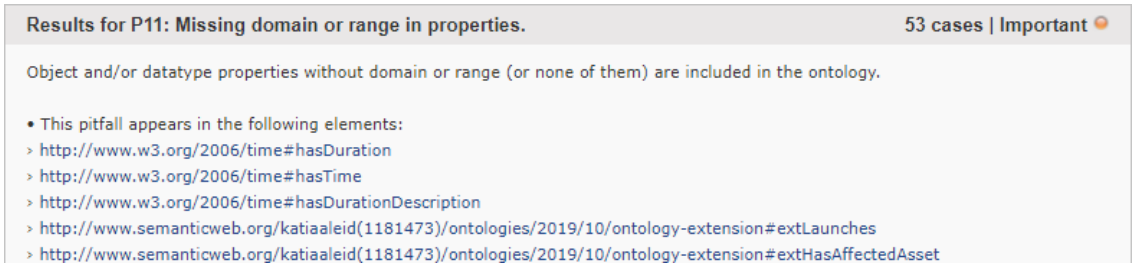

Figure 7.2 – Important pitfall #1

[1] OOPS Tool [Online]. Available: http://oops.linkeddata.es/response.jsp [Accessed: 05-August-2020]

However, several contributions warned about this and advised against it (Rector et al. 2004)(Horridge 2011)(Allemang et al. 2011)(Yusof et al. 2019). That is due to unexpected reasoner behaviour, especially in large and complex ontologies. Therefore, the domain and range – of both object and data properties – were not assigned to follow this best practice notice.

The second pitfall is "P24: Using recursive definitions" which is only found in an imported ontology as shown in Figure 7.3**Error! Reference source not found.**.

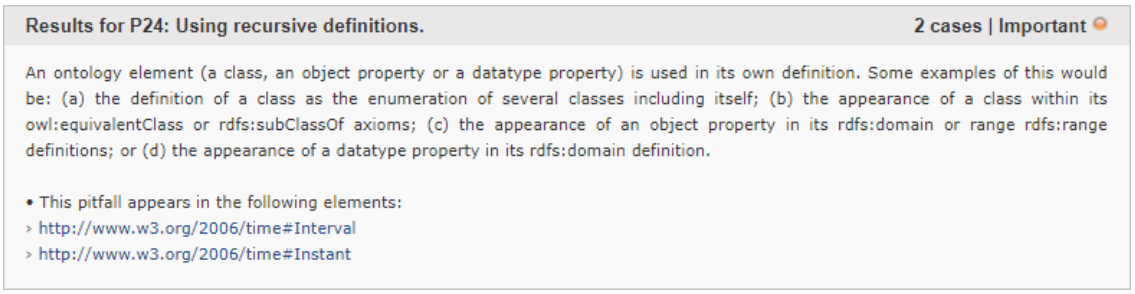

Figure 7.3 – Important pitfall #2

The third pitfall is "P30: Equivalent classes not explicitly declared" in which OOPS! suggests pairs of classes that thought to be duplicated. As mentioned by the creator of OOPS! (Poveda-Villalón 2016), "For each pair of classes that are not defined as equivalents, it is checked whether the concepts they represent could be synonyms according to WordNet and possible equivalences between classes are proposed".

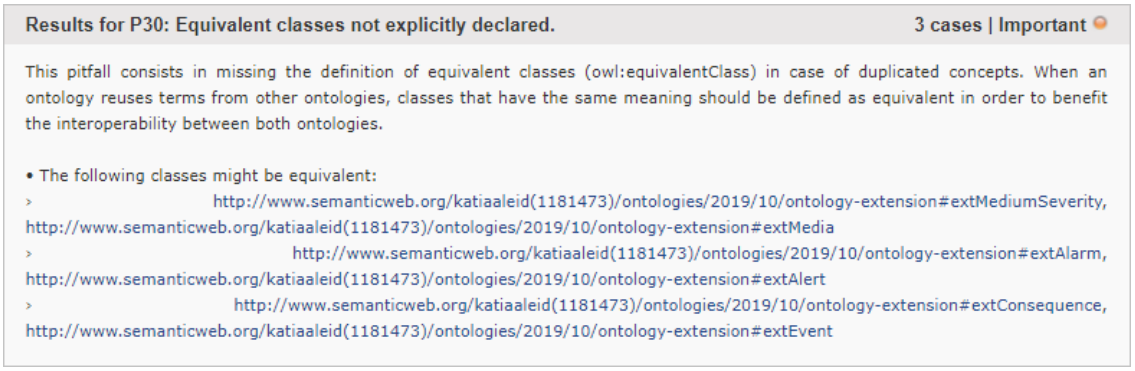

Figure 7.4 – Important pitfall #3

As listed in Figure 7.4**Error! Reference source not found.**, the suggested pairs are:

- Medium Severity (descendant of Severity Level) and Media (descendant of Asset);
- Alarm (descendant of Alert) and Alert;
- Consequence and Event.

After thorough consideration of the intended meaning and usage of these classes in cybersecurity domain, it was decided that they are not equal, and this pitfall will be ignored considering it as a false alarm.

The last pitfall is “P41: No license declared” which is related to the ontology in general as shown in Figure 7.5**Error! Reference source not found.**. This pitfall was fixed by adding ‘dcterms:license’ in the ontology’s metadata.

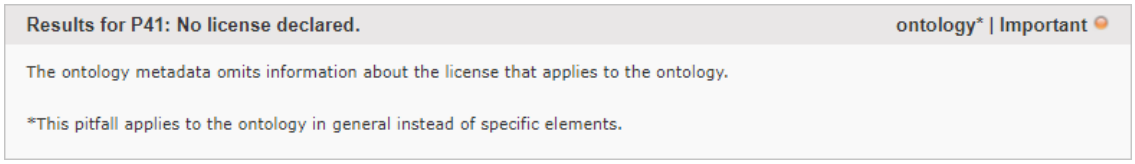

Figure 7.5 – Important pitfall #4

As for the selected license, it was decided to use Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)[1]. The chosen license allows future sharing and adaptation of this work, with the condition of giving appropriate credit and making the new work available under the same license. This is to help other researchers and interested parties make use of this ontology, while encouraging them to keep their work available to others as well.

- **Minor pitfalls**

It is worth mentioning that minor pitfalls do not pose a problem, nor they are mandatory to fix. However, it is suggested by the creator of OOPS! that working on them will make “the ontology in better form and understandable” (Poveda-Villalón 2016).

The first pitfall is “P04: Creating unconnected ontology elements” shown in Figure 7.6**Error! Reference source not found.** due to having the local class ASIIOThing. OOPS! considers it to be isolated for not having any relations to upper level concepts, even though it is a sub-class of ‘owl:Thing’. The class ASIIOThing was made for an organizational purpose as noted in the implementation chapter. Actually, we faced this case with the imported ontologies. For example, UCO has a local class ‘ucoUCOThing’ that holds all the ontology’s concepts. While OWL-Time and ATMONTO do not; their classes are spread out under ‘owl:Thing’. This sometimes makes the ontology unclear and uneasy to deal with, especially in an integrated ontology that is supposed to grow more and include more in the future. Therefore, it was decided to ignore this pitfall and keep the local ASIIOThing class in place.

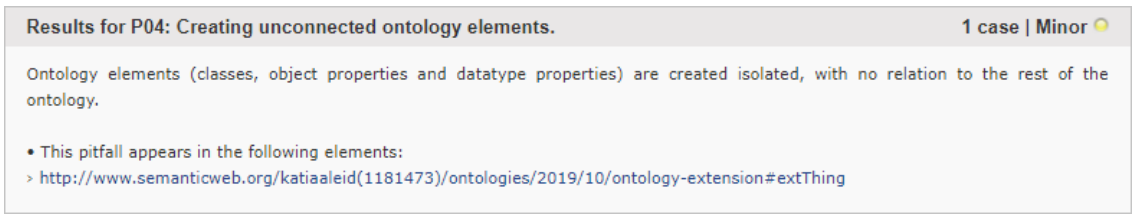

Figure 7.6 – Minor pitfall #1

[1] CC BY-SA 4.0 [Online]. Available: https://creativecommons.org/licenses/by-sa/4.0/ [Accessed: 05-August-2020]

The second pitfall is “P07: Merging different concepts in the same class” relating to DataOrDocument and EnvironmentalAndTechnologialEquipment classes as shown in Figure 7.7**Error! Reference source not found.**. As mentioned before, OOPS! uses lingual analysis with the help of WordNet. Therefore, the use of ‘And’ and ‘Or’ in the classes’ IRIs raised the merging flag.

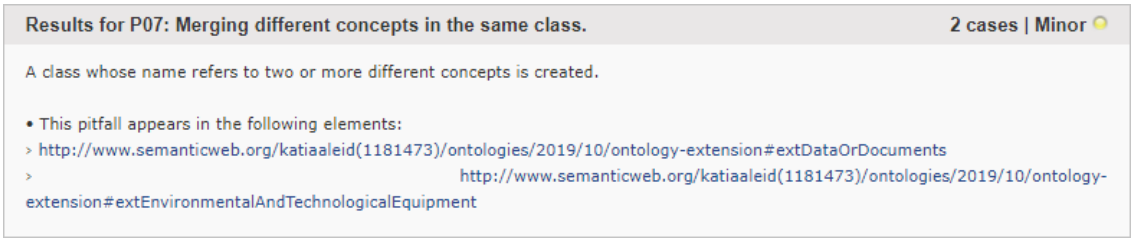

Figure 7.7 – Minor pitfall #2

For DataOrDocument case, the class and its sub-classes were checked. All the sub-classes are considered as Data e.g. Administrative Data, Commercial Data, etc. Therefore, it is possible to rename the class to just Data without affecting its meaning within the ontology. For the EnvironmentalAndTechnologicalEquipment case, the class does not have any descendants. Therefore, it was safely split into two separate classes.

The third pitfall is “P08: Missing annotations” in relation to missing metadata.**Error! Reference source not found.** Figure 7.8 shows a sample of the 172 entities from different ontologies. As for the classes within ASIIO, all the classes and properties have ‘rdfs:label’ filled out. Nonetheless, this pitfall brought to our attention that it is better to have, not just a clear readable label, but also a comprehensive definition of each class. Having this information available within the ontology would make work easier later, rather than going back to documents to look for definitions. Therefore, ‘skos:defintion’ was filled out for classes which need it.

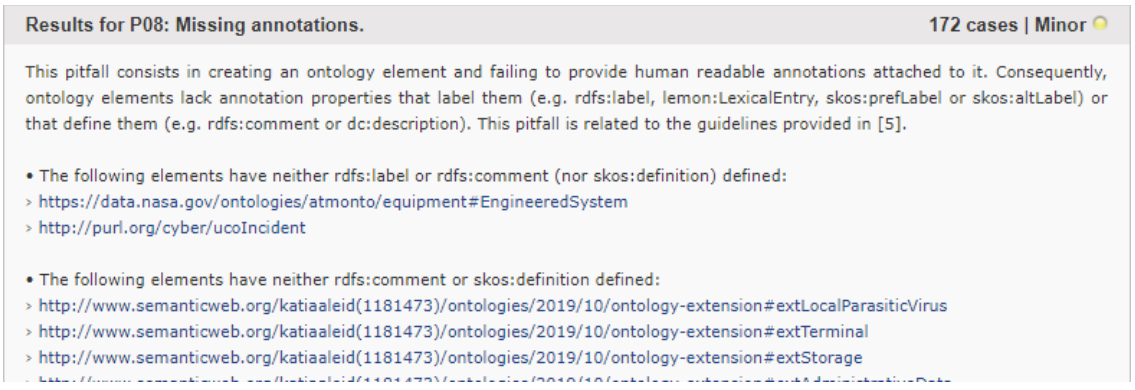

Figure 7.8 – Minor pitfall #3

The fourth pitfall is “P13: Inverse relationships not explicitly declared” as shown in Figure 7.9**Error! Reference source not found.**. New object properties were added and defined as inverse of the existing ones, except for ‘extTopObjectProperty’ which is just for organizational purpose.

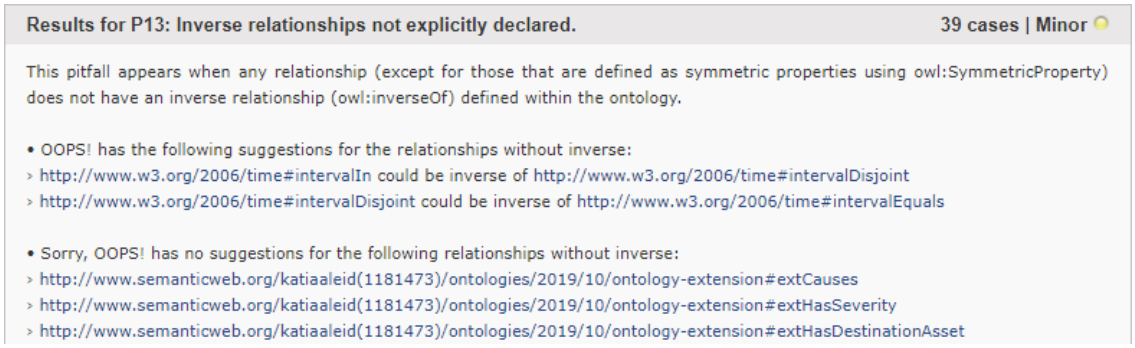

Figure 7.9 – Minor pitfall #4

The last pitfall is "P22: Using different naming conventions in the ontology" as shown in Figure 7.10**Error! Reference source not found.**. ASIIO was already checked to comply with the naming convention described in section 7.4.1. Moreover, this pitfall does not provide enough details to clearly identify the problem, so it was ignored.

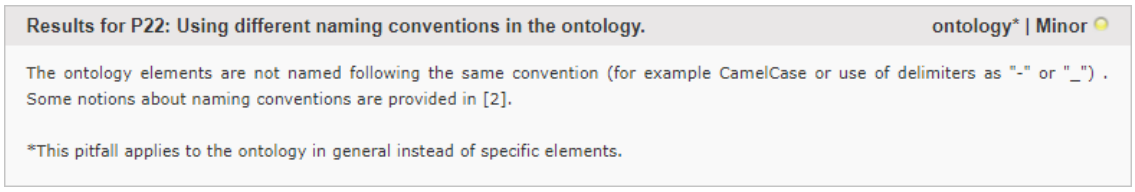

Figure 7.10 – Minor pitfall #5

- **Suggestions**

OOPS! offered 16 suggestions related to object properties as shown in Figure 7.11**Error! Reference source not found.**. All 16 of the listed cases are in OWL-Time ontology. Since all these suggestions are not in ASIIO, there is nothing that can be done about them.

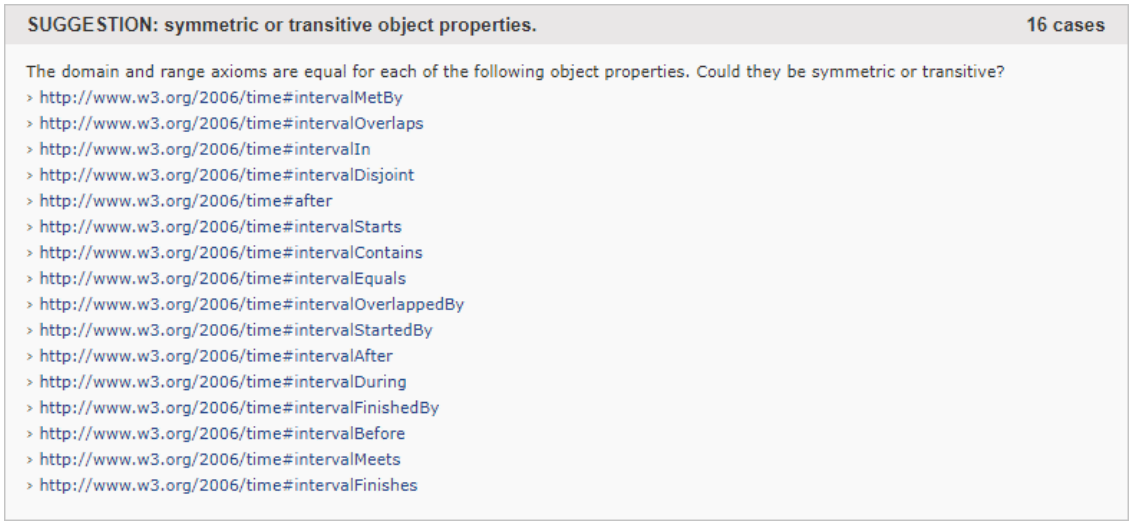

Figure 7.11 – OOPS! suggestions

7.4.3 Scientific community contribution

Following the proposed evaluation plan presented in section 7.3, a paper was submitted to the IEEE 6th International Conference on Computer and Communications (ICCC)[1]. ICCC will take place in Chengdu, China on December 11 – 14, 2020. The paper was titled "An ontology to promote interoperability between cyber-physical security systems in critical infrastructures". It explored the current state of cybersecurity domain and presented a proposal for the integrated ontology being developed. It is worth mentioning that as the paper was submitted at an early stage of this work, the ontology presented in the paper was still at its initial phase. The paper was accepted on September 9, 2020 with ID of IC121.

7.4.4 Relevant systems samples

Several data samples from involved systems were provided by the business partners. The received data came from separated systems that do not share anything in common in terms of assets nor events. Therefore, it was not possible to detect any correlation between them. Unfortunately, it was not possible to obtain data from other related systems in time for this analysis. The received data was in the form of text system logs containing some monitored events and recorded alerts. Each file contained different information according to the application that generated it. These files were analysed and pre-processed before importing the data into the ontology. ASIIO was found to cover all concepts for all the systems with different levels of coverage for the attributes present in the logs. Table 7.2 shows each of the received systems, concepts included in the log files, and the percentage of ASIIO provided coverage.

Table 7.2 – Data sample coverage

	System concepts	ASIIO coverage	Notes
TRAMICS	Alert, Asset, Correlation, Event, Info	Concepts 100% Attributes 90%	Event has 3 types which are not available in ASIIO Event and Correlation both may trigger Alerts, while in ASIIO only Event triggers Alerts
BP-IDS ComSEC	Alert, Asset, Event, Sensor	Concepts 100% Attributes 57.69%	"Sensor detects an Event" relation is not available in ASIIO
GLPI	Alert, Asset, Incident, Impact	Concepts 100% Attributes 57.14%	Some attributes exist but with different data types

[1] ICCC Website [Online]. Available: iccc.org [Accessed: 11-Sept-2020]

7.4.5 Assessment questionnaire

Satisfaction of the business partner with the resulted ontology was assessed via a questionnaire[1] made using Microsoft Forms. The questionnaire consisted of 14 questions with different types and focus points from the business field, important concepts, and the expected improvement of overall performance after using ASIIO. The questionnaire was sent to the consortium and received 10 responses, as just one partner from each organization fulfilled it, within a period of 2 weeks.

The responses showed a variety of application fields by the organizations that use ontologies to support their business, with the main areas being airports and cybersecurity. It is noted also that the participants have not used ontologies for physical security aspects so far. This would present a place for ASIIO to help with achieving interoperability among physical and cyber security systems. Figure 7.12 shows the ontologies usage sectors as per the participants' responses to the questionnaire.

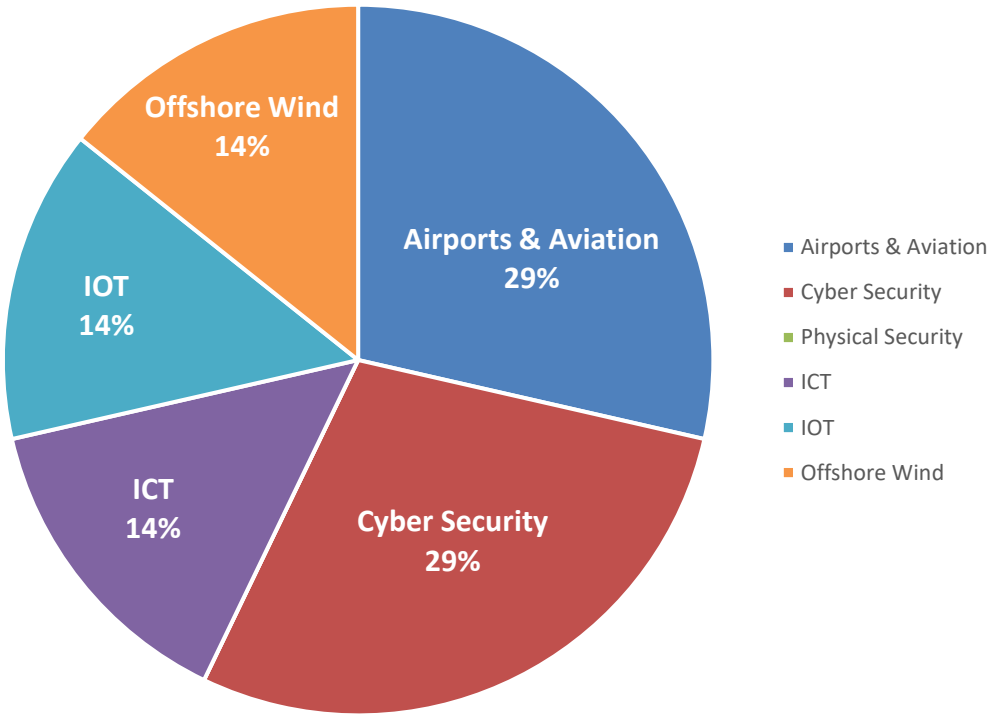

Figure 7.12 – Participants usage of ontologies

[1] Assessment questionnaire results [Online]. Available: https://forms.office.com/Pages/AnalysisPage.aspx?id=KaZgeh_mDEOcYyZmU7K4mdUeo3dAtz9Hm8Mts2EVprNUMkFZNEREM0JRQVlTM1daWFZCWlVGN0hCNC4u&AnalyzerToken=7kJYxGmmBWQrB81nqdQxSyVfwuW65XzB [Accessed: 08-Oct-2020]

In what regards to the developed ontology ASIIO, the participants thought that it would help make the exchanged data clearer and easier to read and understand in addition to improving the extensibility compared to previous practices. They also thought that ASIIO helped them to understand the roles and responsibilities of other systems and improve interoperability as shown in Figure 7.13.

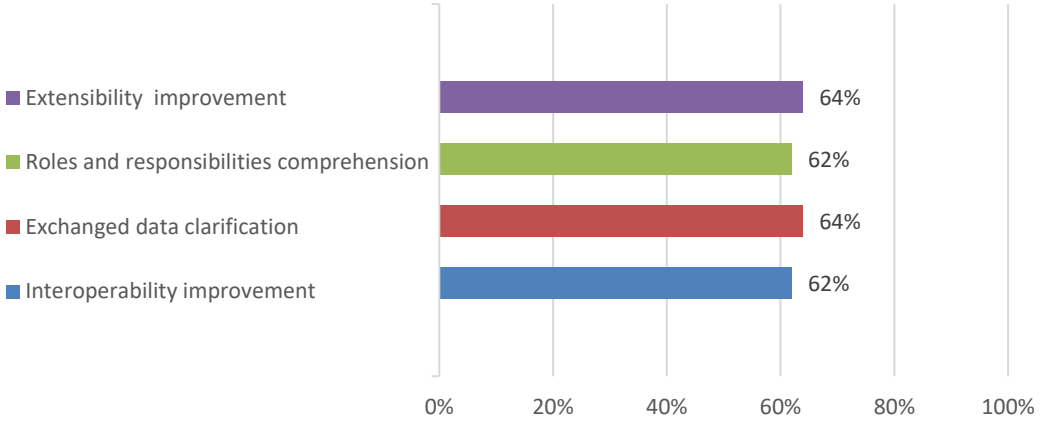

Figure 7.13 – Feedback about ASIIO

As for enabling communication with different systems, ASIIO was also found to help clarify communication with up to 3 systems as the responses show in Figure 7.14.

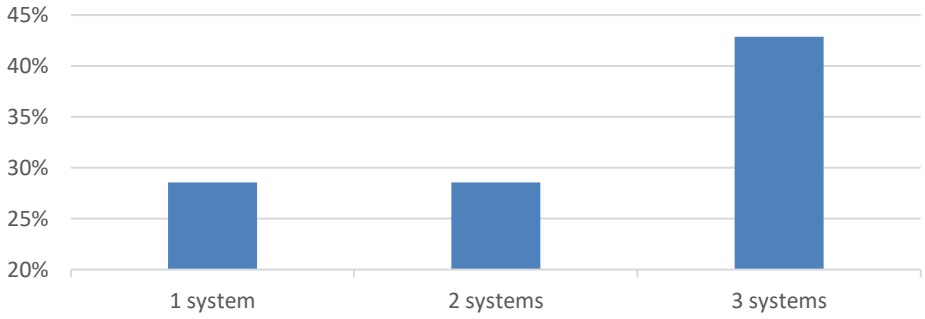

Figure 7.14 – Enabled communication via ASIIO

In the questionnaire, a selection of concepts was presented to collect some specific feedback from the partners in terms of importance. In general, the concepts were positively evaluated, and Attack appears to be the most important one for the majority of the partners as shown in Figure 7.15. Some participant gave suggestions for additional concepts that they are interested in, such as Risk, Regulation, Communication, and Compliance. These concepts can be studied and considered to be added to the ontology in future versions.

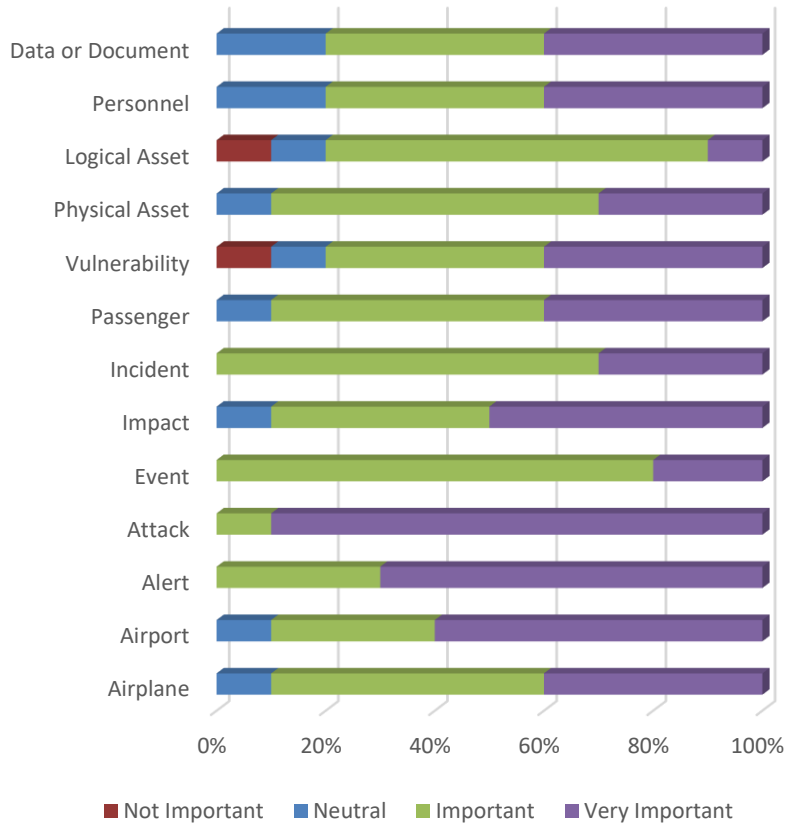

Figure 7.15 – Main concepts feedback

The perception of ASIIO was positive and promising in general. It appears to provide the essential features to cover partners' needs with a room for improvement in some aspects. The majority of participants in this questionnaire stated that they would use ASIIO in their projects as shown in Figure 7.16, which is a good indicator of their satisfaction with the current state of the ontology.

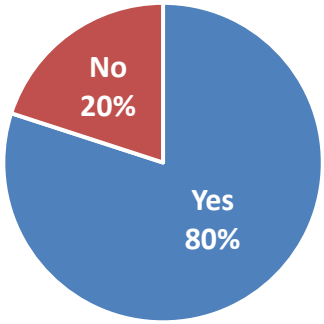

Figure 7.16 – Participants' answer to using ASIIO

7.5 Summary

In this chapter, the actions taken to validate ASIIO were described. The developed OWL file was verified to follow the OD101 guidelines. It also was run through the online tool (OOPS!) to detect any pitfalls which were then discussed and fixed. Data sample files were received from 4 partners, which were analysed, pre-processed, and imported into the ontology with 100% concept coverage rate. An assessment questionnaire was formed and sent to the partners to collect their feedback regarding ASIIO. The feedback was good overall, and 80% of them showed interest in using ASIIO into their future projects.

8 Conclusion

In modern days, online services are taking over more aspect of our lives. Transportation is one them. It is very important to ensure the security and safety of the users, their life, and their information, against possible threats and harmful usage. Despite having some work done in the scope of cybersecurity, it is noted that airport cybersecurity still needs further enhancement. SATIE is a European project concerned with airport security as a part of the European Union's Horizon 2020 research and innovation programme. The cooperation with the consortium of SATIE provided a valuable source of information related to the needs and requirements of an ongoing use case. Having SATIE as an application study was helpful towards forming the domain boundaries and obtaining real-life feedback throughout the development process.

The use of ontologies is thought to facilitate the integration of several systems together by using a common vocabulary. Therefore, this work explored the principals of ontologies and their categorization and methodologies. Which in turn helped make the decision of what level the developed ontology will be, and what methodology and tool will be used.

In order to get a better understanding of the current state of this domain, a state-of-the-art review was conducted in relation to both cybersecurity and airports domains. With a clear set of criteria, the relevant publications were filtered, studied, and summarized. 28 available approaches were found till the time of conducting the review. Through the study, and with feedback from SATIE, a domain set was formed by selecting concepts which were thought relevant to the airports' cybersecurity and SATIE. The set included 413 concepts related to cyber and physical security and airports infrastructure, which most of them were available in one ontology at a time. Also, the studied approaches were found to use different formats and have different focus within the overall domain. The presented analysis of these approaches compared their interests and usage, and it showed how separated they are despite of some overlapping. Thus, highlighting the need for the integrated ontology for better communication.

Regarding the need of the intended ontology in the market, a business value analysis was performed. It is important for this work to be useful in practical application fields, and not just

as a theoretical research. In this context, Value Proposition Canvas helped clarifying the customers' pains and the gains they would get from such ontology and SWOT analysis helped to shed light on the main advantages and disadvantages this work provides, in addition to the opportunities available to be seized and threats to be aware of.

The conducted review gave a clear idea about the current state and what is needed for a better domain support. Therefore, an initial set of concepts were made to be the starting point for the development process. The concepts were selected from the previous study and feedback of SATIE, and a survey was made to get a deeper look on how these concepts are defined by different organisations. Then, new definitions were made for these concepts. The new definitions take into consideration the conducted survey along with SATIE use case. Therefore, it is thought that the consensus definition would help fulfill the needs of many systems.

With that in mind, the requirements for the new ontology were formed. The OSRD was made to describe the characteristics desired from the developed ontology. The design of the ontology started from an initial set of 4 concepts and grew bigger with each explored use case scenario. As a result, the ontology now consists of 153 concepts including assets and attack types. Several ontologies were extended to make use of publicly available efforts like ATMONTO for airports and UCO for cybersecurity. At this point, and due to time limitations, the ontology is simply a preliminary core thought to expand in future works. As the design progressed, the domain concepts' set was enriched with additional 100 concepts. According to the updated domain set, the designed ontology along with the previously studied 28 approaches were compared. It was found out that the developed ontology provides the highest coverage of the domain so far. And with more elaborating in the future, the ontology is expected to evolve to cover more of the domain. A description logic representation of the ontology was provided for the main concepts and relations.

The intended ontology was named Airport Security Interoperability Integrated Ontology (ASIIO). ASIIO was implemented in OWL ontology language using Protégé editor. The main classes were described, and a full list of definitions was also provided. Object properties were also illustrated using OntoGrafs.

Different actions were taken for the evaluation of ASIIO. First, for the validation of the developed OWL structure, the ontology was checked against the guidelines of OD101. The OWL file was also scanned by an online tool "OOPS!" to help discover any other mistakes. Any issues highlighted by OD101 and OOPS! were either fixed or refuted.

Second, a scientific contribution was made in a form of a paper submitted to ICCC 2020, China; where the paper was peer-reviewed and accepted.

Third, several data samples from relevant systems were provided by the business partners. After analysis and pre-processing the data, it was imported into the ontology with 100% concept coverage rate. Thus, ensuring that ASIIO is well-matched with the said systems' requirements.

Lastly, a questionnaire was created to gauge partners' satisfaction with ASIIO and to collect their feedback. The general feedback was positive and confirms that ASIIO provides most of the desired features it was developed for. The feedback also included some suggestions to improve certain aspects of the ontology. Despite ASIIO being an initial core for the airport cybersecurity domain ontology and the available room for improvement, majority of the participants stated that ASIIO would help them better understand and communicate several systems as well as improve interoperability and extensibility. Furthermore, 80% of them showed the will to use ASIIO in their projects.

Future prospect for ASIIO include monitoring its performance in several projects to get a real-life feedback for any issues or shortcomings, then work on solving these issues and develop the ontology into a better version. The feedback received from the partners via the questionnaire may also be studied and considered to be included later.

References

Abdelghany, A.S., Darwish, N.R., & Hefni, H.A. 2019. An agile methodology for ontology development. *International Journal of Intelligent Engineering and Systems* 12(2): p.170–181.

Aime, M.D., & Guasconi, F. 2010. Enhanced vulnerability ontology for information risk assessment and dependability management. *Proceedings - 3rd International Conference on Dependability, DEPEND 2010*: p.92–97. Available at: https://ieeexplore.ieee.org/abstract/document/5562843/ [Accessed January 16, 2020].

Allemang, D., & Hendler, J. 2011. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Second Edition*.

Bergner, S., & Lechner, U. 2017. Cybersecurity ontology for critical infrastructures. In *IC3K 2017 - Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 80–85. Available at: www.itskritis.de [Accessed January 15, 2020].

Brunk, B.K., & Porosnicu, E. 2004. A Tour of the AIXM Concepts. 424. Available at: http://www.schematron.com [Accessed February 14, 2020].

Chen, Y., Peng, X., Zhong, B., & Luo, H. 2016. Application of ontology in vulnerability analysis of metro operation systems. *Structure and Infrastructure Engineering* 12(10): p.1256–1266. Available at: https://www.tandfonline.com/doi/abs/10.1080/15732479.2015.1110602 [Accessed January 16, 2020].

Choraś, M., Kozik, R., Flizikowski, A., & Hołubowicz, W. 2010. Ontology applied in decision support system for critical infrastructures protection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6096 LNAI(PART 1): p.671–680.

Cuppens-Boulahia, N., Cuppens, F., Autrel, F., & Debar, H. 2009. An ontology-based approach to react to network attacks. *International Journal of Information and Computer Security* 3(3–4): p.280–305.

Danyliw, R.J. 2007. The incident object description exchange format, IETF RFC 5070. 62(1):

p.27–40. Available at: http://www.rfc-editor.org/info/rfc7970. [Accessed February 14, 2020].

Doynikova, E., Fedorchenko, A., & Kotenko, I. 2019. Ontology of metrics for cyber security assessment. *ACM International Conference Proceeding Series*. Available at: https://dl.acm.org/doi/abs/10.1145/3339252.3341496 [Accessed January 16, 2020].

Fernandez, M., Gómez-Pérez, A., & Juristo, N. 1997. Methontology: from ontological art towards ontological engineering. *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering* (May 2014): p.33–40. Available at: http://speech.inesc.pt/~joana/prc/artigos/06c METHONTOLOGY from Ontological Art towards Ontological Engineering - Fernandez, Perez, Juristo - AAAI - 1997.pdf.

Gao, J.B., Zhang, B.W., Chen, X.H., & Luo, Z. 2013. Ontology-based model of network and computer attacks for security assessment. *Journal of Shanghai Jiaotong University (Science)* 18(5): p.554–562.

Gerhards, R., & GmbH, A. 2009. RFC 5424 The Syslog Protocol. *Network Working Group, IETF*: p.10–11. Available at: http://trustee.ietf.org/license-info [Accessed February 14, 2020].

De Giacomo, G., & Lenzerini, M. 1996. TBox and ABox Reasoning in Expressive Description Logics. In *Proceedings of the Fifth International Conference on the Principles of Knowledge Representation and Reasoning (KR'96)*, 37–48.

Gonzalez Granadillo, G., Ben Mustapha, Y., Hachem, N., & Debar, H. 2012. An ontology-based model for SIEM environments. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering* 99 LNICST: p.148–155.

Greitzer, F.L., Lee, J.D., Purl, J., & Zaidi, A.K. 2019. Design and Implementation of a Comprehensive Insider Threat Ontology. *Procedia Computer Science* 153: p.361–369. Available at: https://www.sciencedirect.com/science/article/pii/S1877050919307495 [Accessed January 16, 2020].

Helle V. Dam, Jan Engberg, H.G.-A., & Gruyter, W. de. 2012. *Knowledge Systems and Translation* H. V. Dam, J. Engberg, & H. Gerzymisch-Arbogast (eds). Berlin, Boston: DE GRUYTER. Available at: https://books.google.pt/books?id=IL2E9xuJLAAC&pg=PA113&redir_esc=y#v=onepage&q&f=false [Accessed February 14, 2020].

Horridge, M. 2011. A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3.

Islam, C., Babar, M.A., & Nepal, S. 2019. Automated Interpretation and Integration of Security Tools Using Semantic Knowledge. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 513–528. Springer Verlag

Jafer, S., Chhaya, B., Durak, U., & Gerlach, T. 2016. Formal scenario definition language for aviation: Aircraft landing case study. *AIAA Modeling and Simulation Technologies Conference, 2016* (June).

Kenaza, T., & Aiash, M. 2016. Toward an Efficient Ontology-Based Event Correlation in SIEM.

Procedia Computer Science 83: p.139–146. Available at: www.sciencedirect.com [Accessed January 15, 2020].

Kenaza, T., Machou, A., & Dekkiche, A. 2018. Implementing a Semantic Approach for Events Correlation in SIEM Systems. In *IFIP Advances in Information and Communication Technology*, 648–659. Available at: https://link.springer.com/chapter/10.1007/978-3-319-89743-1_55 [Accessed January 15, 2020].

Koen, P.A., Bertels, H.M.J., & Kleinschmidt, E. 2014. Managing the front end of innovation-part I: Results from a three-year study. *Research Technology Management* 57(2): p.34–43.

Krauß, D., & Thomalla, C. 2016. Ontology-based detection of cyber-attacks to SCADA-systems in critical infrastructures. In *2016 6th International Conference on Digital Information and Communication Technology and Its Applications, DICTAP 2016*, 70–73. Available at: https://ieeexplore.ieee.org/abstract/document/7544003/ [Accessed January 15, 2020].

Leung, N.K.Y., Lau, S.K., Fan, J., & Tsang, N. 2011. An integration-oriented ontology development methodology to reuse existing ontologies in an ontology development process. *ACM International Conference Proceeding Series*: p.174–181.

Lim, S.Y., Song, M.H., & Lee, S.J. 2004. The construction of domain ontology and its application to document retrieval. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3261: p.117–127. Available at: http://link.springer.com/10.1007/978-3-540-30198-1_13 [Accessed May 17, 2020].

López, F. 1999. Overview Of Methodologies For Building Ontologies. *Proceedings of the IJCAI99 Workshop on Ontologies and ProblemSolving Methods Lessons Learned and Future Trends CEUR Publications* 1999(2): p.1–13. Available at: http://iwayan.info/Research/Ontology/Tutor_Workshop/Tutorial_4_Analysis.pdf.

Malone, J., & Parkinson, H. 2010. Reference and application ontologies. *Ontogenesis* (6): p.22–25. Available at: http://ontogenesis.knowledgeblog.org/295.

Martins, A.F., & De Almeida Falbo, R. 2008. Models for representing task ontologies. In *CEUR Workshop Proceedings*,

Menzel, C., & Menzel, C. 2003. Reference Ontologies - Application Ontologies: Either/Or or Both/And? *Proceedings of the KI2003 Workshop on Reference Ontologies and Application Ontologies*: p.1–10. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1.2490.

Michal, K., Michal, Š., & Zdeněk, B. 2012. Interoperability through ontologies. *IFAC Proceedings Volumes (IFAC-PapersOnline)* 11(PART 1): p.196–200.

Moliner, M.A. 2009. Loyalty, perceived value and relationship quality in healthcare services. *Journal of Service Management* 20(1): p.76–97. Available at: www.emeraldinsight.com/1757-5818.htm [Accessed February 10, 2020].

Mundie, D. et al. 2014. *An incident management ontology*.

Noy, N.F., & McGuinness, D.L. 2001. Ontology Development 101: A Guide to Creating Your First Ontology. In *Stanford Knowledge Systems Laboratory*, 25.

Obrst, L., Chase, P., & Markeloff, R. 2014. Developing an ontology of the cyber security domain. *CEUR Workshop Proceedings* 966: p.49–56. Available at: https://duck.franz.com/agraph/cresources/white_papers/STIDS2012_T06_ObrstEtAl_CyberOntology.pdf [Accessed January 17, 2020].

Oltramari, A., Cranor, L.F., Walls, R.J., & McDaniel, P. 2014. Building an ontology of cyber security. *CEUR Workshop Proceedings* 1304: p.54–61.

Oltramari, A., Cranor, L.F., Walls, R.J., & McDaniel, P. 2015. Computational ontology of network operations. *Proceedings - IEEE Military Communications Conference MILCOM* 2015-Decem: p.318–323.

Onwubiko, C. 2018. CoCoa: An ontology for cybersecurity operations centre analysis process. *2018 International Conference on Cyber Situational Awareness, Data Analytics and Assessment, CyberSA 2018*.

Peroni, S. 2016. SAMOD : an agile methodology for the development of ontologies. : p.1–11.

Pershing, J.A. 2006. *Handbook of Human Performance Technology Third Edition*.

Poveda-Villalón, M. 2016. Ontology Evaluation: a pitfall-based approach to ontology diagnosis. : p.236. Available at: https://core.ac.uk/download/pdf/148679315.pdf%0Ahttp://oa.upm.es/39448/.

Raad, J., Raad, J., & Cruz, C. 2017. A Survey on Ontology Evaluation Methods A Survey on Ontology Evaluation Methods. (November 2015).

Radziwill, N. 2015. *Value Proposition Design*.

Rector, A. et al. 2004. OWL pizzas: Practical experience of teaching OWL-DL: Common errors and common patterns. In *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 63–81.

Risk Based Security. 2015. *Data Breach QuickView*.

Risk Based Security. 2019. Data Breach QuickView Report Year End 2019. *Cyber Risk Analytics*: p.1–23. Available at: https://pages.riskbasedsecurity.com/2018-ye-breach-quickview-report.

Seganti, A., Kapłański, P., & Zarzycki, P. 2016. Collaborative Editing of Ontologies Using Fluent Editor and Ontorion. In V. Tamma, M. Dragoni, R. Gonçalves, & A. Ławrynowicz (eds) *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.Lecture Notes in Computer Science, 45–55. Cham: Springer International Publishing Available at: http://link.springer.com/10.1007/978-3-319-33245-1.

Singh, V., & Pandey, S.K. 2019. Cloud Security Ontology (CSO). *Springer*: p.81–109. Available at: https://link.springer.com/chapter/10.1007/978-3-030-03359-0_4 [Accessed January 17, 2020].

Smith, B. 2003. Ontology. *Blackwell Guide to the Philosophy of Computing and Information, Oxford: Blackwell* (1964): p.155–166.

Suárez-figueroa, M.C., & Gómez-pérez, A. 2009. How to Write and Use the Ontology Requirements Specification Document How to Write and Use the Ontology Requirements. (November 2009): p.966–982.

Syed, Z., Pädia, A., Finin, T., Mathews, L., & Joshi, A. 2016. UCO: A Unified Cybersecurity Ontology. *AAAI Workshop - Technical Report* WS-16-01-(Figure 1): p.195–202.

Tamea, G., Cusmai, M., Palo, A., Priscoli, F.D., & Cimmino, A. 2014. Situation awareness in airport environment based on Semantic Web technologies. *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support, CogSIMA 2014*: p.174–180. Available at: https://www.researchgate.net/publication/269306945 [Accessed January 16, 2020].

Ulicny, B.E., Moskal, J.J., Kokar, M.M., Abe, K., & Smith, J.K. 2014. Inference and ontologies. *Advances in Information Security* 62: p.167–199.

UNIVERSITY OF JYVÄSKYLÄ. 2015. Semantic Web and Linked Data Web of People Internet of Things Web of Data Evolution of the Web.

Wang, J.A., & Guo, M. 2009. OVM: An ontology for vulnerability management. *ACM International Conference Proceeding Series*.

Wang, J.A., Guo, M.M., & Camargo, J. 2010. An ontological approach to computer system security. *Information Security Journal* 19(2): p.61–73. Available at: http://www.tandfonline.com/doi/abs/10.1080/19393550903404902 [Accessed January 16, 2020].

Wita, R., Jiamnapanon, N., & Teng-amnuay, Y. 2010. An ontology for vulnerability lifecycle. *3rd International Symposium on Intelligent Information Technology and Security Informatics, IITSI 2010*: p.553–557. Available at: https://ieeexplore.ieee.org/abstract/document/5453687/ [Accessed January 16, 2020].

Woodall, T. 2003. *Conceptualising 'Value for the Customer': An Attributional, Structural and Dispositional Analysis*. Available at: https://www.researchgate.net/publication/228576532 [Accessed February 23, 2020].

Yusof, N.M., & Noah, S.A.M. 2019. Malaysian food composition ontology evaluation. *International Journal of Machine Learning and Computing* 9(5): p.700–705.

Zhao, Y., Lang, B., & Liu, M. 2018. Ontology-based unified model for heterogeneous threat intelligence integration and sharing. *Proceedings of the International Conference on Anti-Counterfeiting, Security and Identification, ASID* 2017-Octob: p.11–15. Available at: https://ieeexplore.ieee.org/abstract/document/8285734/ [Accessed January 16, 2020].

Appendices

Appendix A – Asset Hierarchy

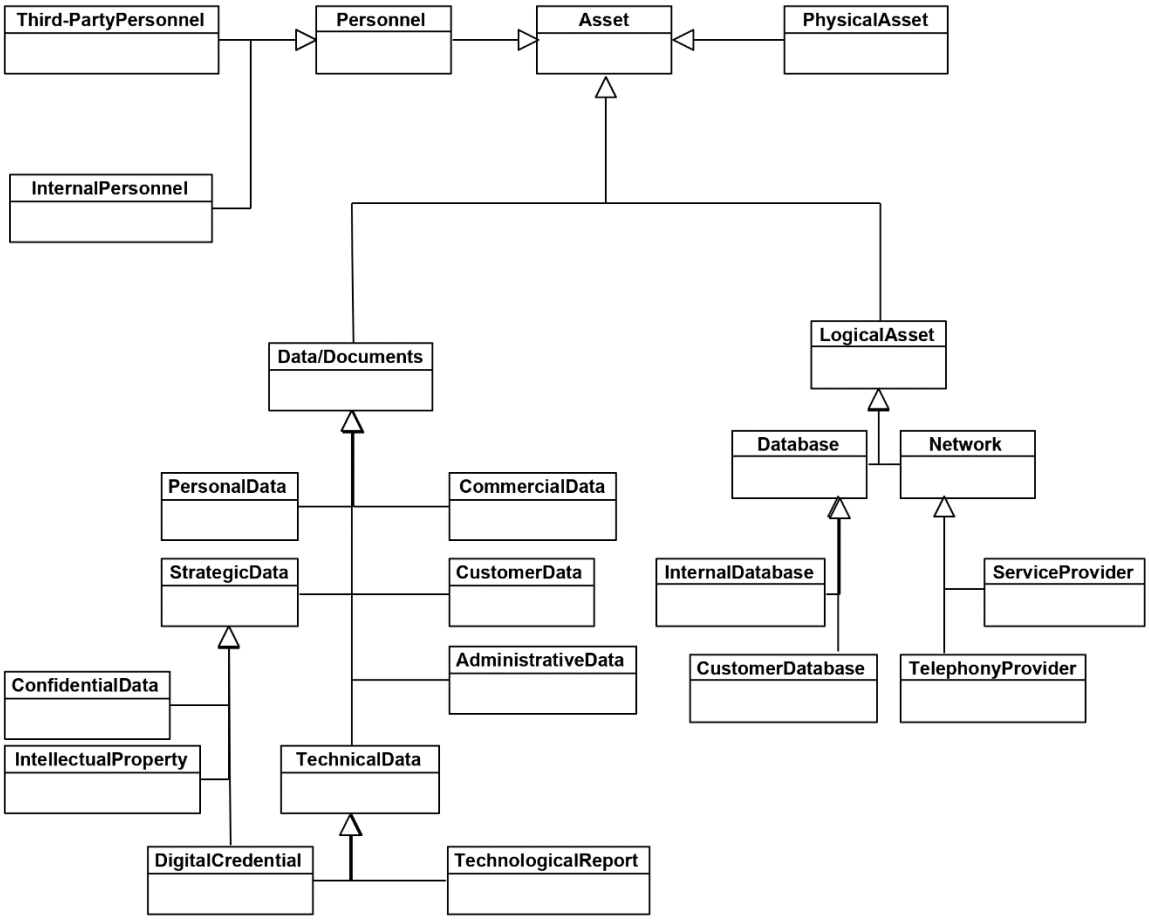

Figure A.1 – View of assets classes v 1.4

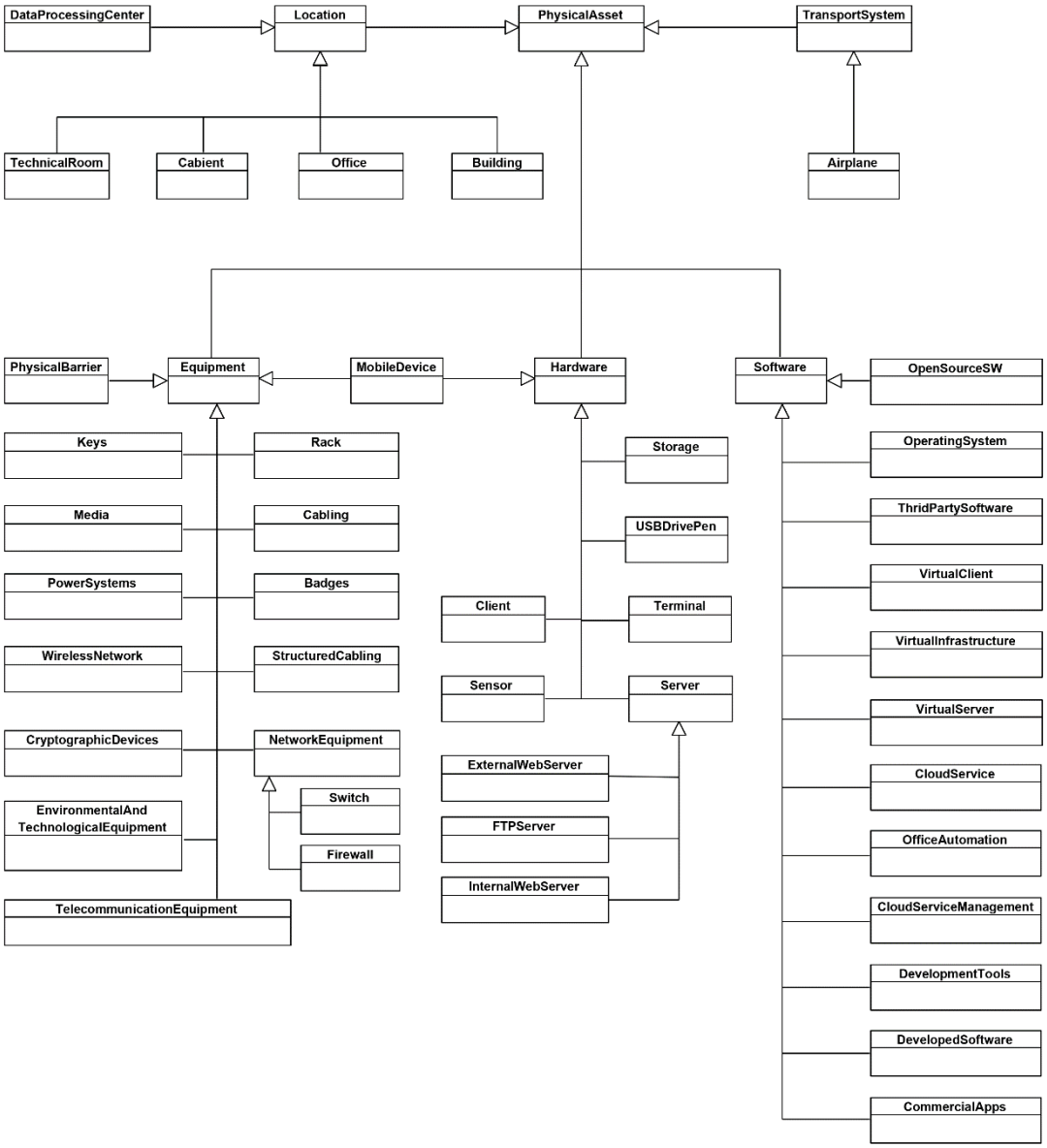

Figure A.2 – View of physical assets classes v 1.4

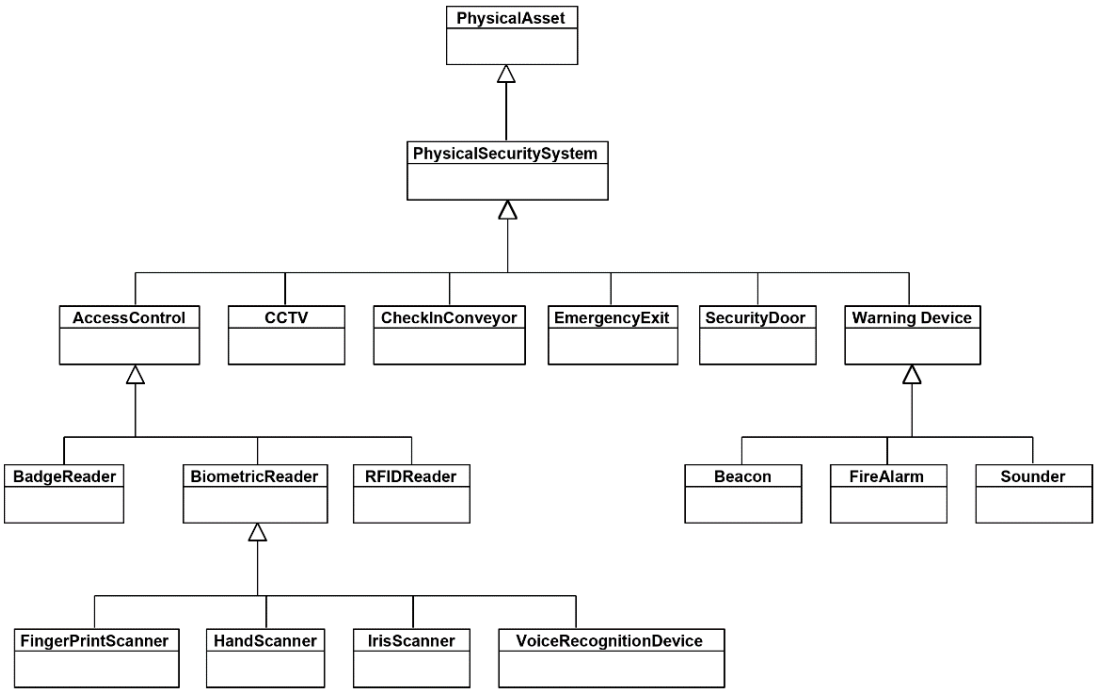

Figure A.3 – Physical security assets v 1.7

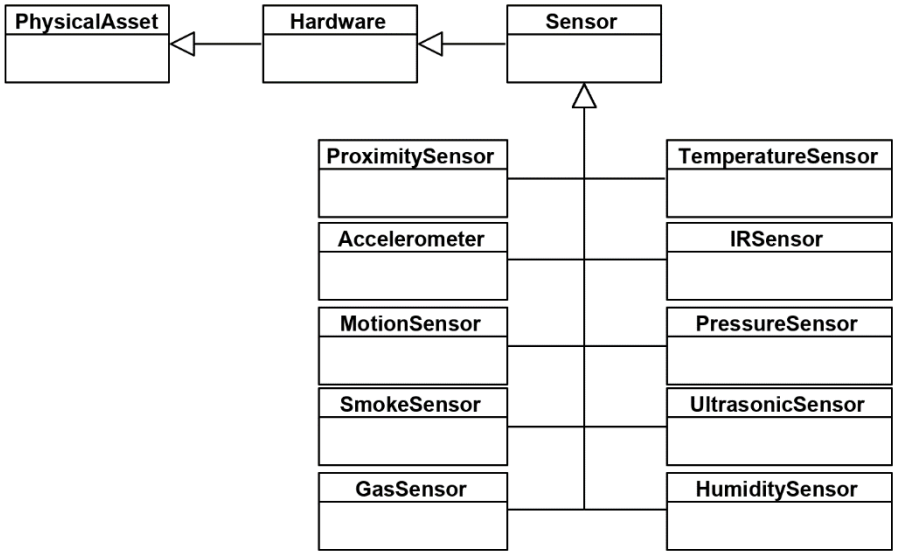

Figure A.4 – Sensors hierarchy v 1.7

Appendix B – Attack Type Hierarchy

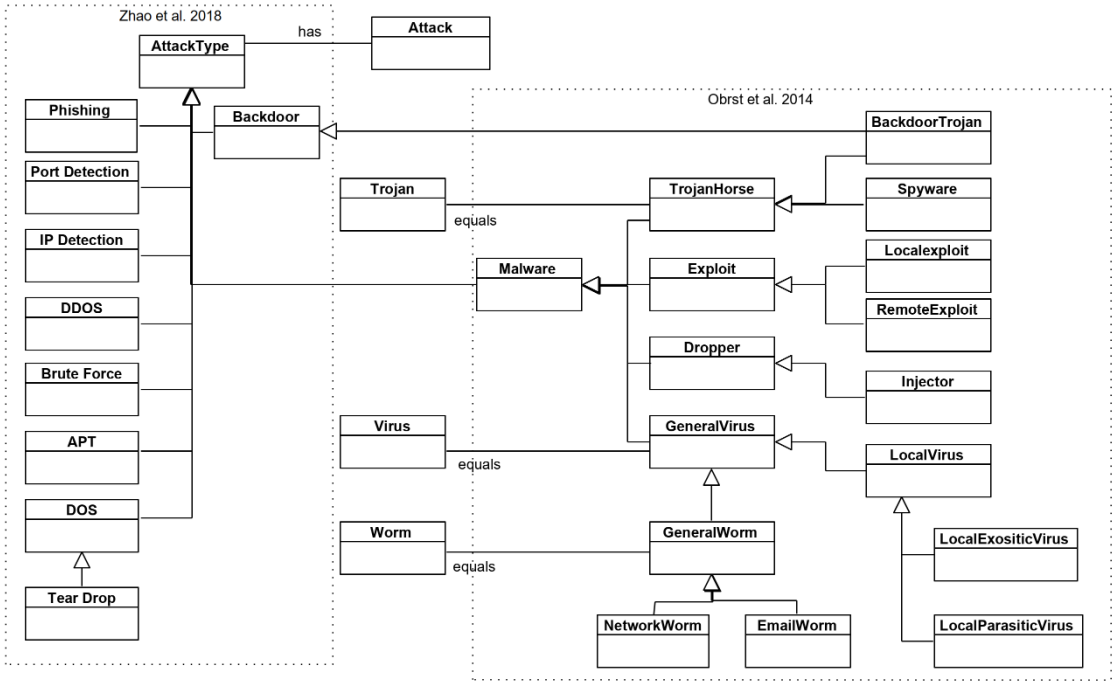

Figure A.5 – AttackType hierarchy v 1.6

Appendix C – List of Concepts

Accelerometer	A tool that measure proper acceleration
Access control	The selective restriction of access to a place or resource
Advanced Persistent Threat	An attack in which an unauthorized user gains access to a system or network and remains there for an extended period of time without being detected
Advisory	A type of Alert fulfilling a specific criterion to be determined later
Airplane	A powered flying vehicle with fixed wings and a weight greater than that of the air it displaces
Alarm	An alert with a Security Level that is either High or Extreme
Alert	A notification that a specific attack has been directed at an organization's information systems.
Asset	Information or resource which have value to an organization or person.
Attack	An attempt by an unauthorized individual to fool a Verifier or a Relying Party into believing that the unauthorized individual in question is the Subscriber
Attack Type	A category or classification of the attack
Attacker	a party, including an insider, who acts with malicious intent to compromise a system
Backdoor	A covert method of bypassing normal authentication or encryption in a computer, product, embedded device, or its embodiment
Backdoor Trojan	An attack that gives malicious users remote control over the infected computer
Badge	A small piece of metal, plastic, or cloth bearing a design or words, typically worn to identify a person or to indicate membership of an organization
Badge Reader	A data input device that reads data from a **Badge**
Beacon	A block that projects a light beam to attract attraction or provide a status
Biometric Reader	A device that reads the identity of a person by comparing some attributes or their physiological being or behavioural traits against a sample database
Brute Force	An attack consists of an attacker submitting many passwords or passphrases with the hope of eventually guessing correctly
Building	A structure with a roof and walls
Business Impact Assessment	A way to predict the consequences of disruption of a business function and process and gathers information needed to develop recovery strategies
Business Process	An attack that targets specific business processes and attempts

Compromised Event	to compromise them for some financial gain
Cabinet	A cupboard with shelves or drawers for storing or displaying articles
Cabling	An electrical or electronic cable
CCTV	A video camera used to transmit a signal to a specific place on a limited set of monitors
Check-in conveyor	A device used to weigh baggage and dispatch it onto the collector conveyor without the need for manual lifting
Client	A desktop computer or workstation that is capable of obtaining information and applications from a server
Cloud Service	A service delivered on demand to companies and customers over the internet
Cloud Service Management	A collection of activities that an organization does to plan, design, deliver, operate, and control the IT and cloud services that it offers to customers
Commercial App	A software or program that is designed and developed for licensing or sale to end users or that serves a commercial purpose
Configuration	The possible conditions, parameters, and specifications with which an information system or system component can be described or arranged
Consequence	An effect (change or non-change), usually associated with an event or condition or with the system and usually allowed, facilitated, caused, prevented, changed, or contributed to by the event, condition, or system
Correlation	Finding relationships between two or more log entries
Countermeasure	A protective measure prescribed to meet the security requirements (i.e., confidentiality, integrity, and availability) specified for an information system. Safeguards may include security features, management constraints, personnel security, and security of physical structures, areas, and devices
Criticality	A value partition that controls the level of criticality of an event
Cryptographic Device	A device capable of accelerating and decrypting Secure Sockets Layer (SSL)
Customer Database	A database related to customers' information
Data	Data collections that are used or produced from the work process
Data Processing Center	A place where various electronic equipment, especially computers and telecommunications equipment
Database	An organized collection of data stored and accessed on computers
Denial of Service DoS	A cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host

	connected to the Internet
Developed Software	A program that was conceived, specified, designed, programmed, documented, and tested
Development Tool	A tool that supports the rapid implementation of software applications upon a programming platform
Disabled	A state of inactivity
Distributed Denial of Service DDoS	A malicious attempt to disrupt normal traffic to a web property
Dropper	A kind of Trojan that has been designed to "install" some sort of malware to a target system
Email Worm	A worm spread via email
Emergency	A criticality value that the related event impacts to safety/breach of critical systems
Emergency Exit	A special exit for emergencies such as fire for faster evacuation if the regular exit in blocked.
Enabled	A state of activity
Engineered System	A subclass used to link to ATMONTO/equipment
Environmental Equipment	A system, product, equipment, or technology responsible for the mitigation of environmental damage and minimization of the environmental impact
Equipment	Necessary items for a particular purpose
Escalation	A criticality value that the related event requires the application of corrective measures
Event	A discrete change of state or status of an Asset or group of Assets. Specific Events may trigger Alerts
Exploit	A code that takes advantage of a software vulnerability or security flaw
External Web Server	A third-party web server where companies outsource their website hosting
Extreme	The highest severity level
Finger Print Scanner	A device that optically scans the fingerprint when the user touches the glowing window
Fire Alarm	A device used to detect and warn people through visual and audio appliances when smoke, fire, carbon monoxide or other emergencies are present
Firewall	A network security device that monitors incoming and outgoing network traffic and permits or blocks data packets based on a set of security rules
FTP Server	A software application which enables the transfer of files from one computer to another
Gas Sensor	A tool that detects gases like oxygen, carbon dioxide, nitrogen, etc
General Virus	A malicious code that replicates by copying itself to another program, computer boot sector or document and changes how

	a computer works
General Worm	A type of malware that spreads copies of itself from computer to computer. A worm can replicate itself without any human interaction, and it does not need to attach itself to a software program in order to cause damage
Hand Scanner	See **Palm Scanner**
Hardware	Tools, machinery, and other durable equipment
High	A high severity level
Humidity Sensor	A tool that detects and measure water vapor
Impact	The magnitude of harm that can be expected to result from the consequences of unauthorized disclosure of information, unauthorized modification of information, unauthorized destruction of information, or loss of information or information system availability
Incident	An Event (or group of Events) that compromises an Asset. An Incident may be retroactively classified as an attack. Additionally, it has some sort of impact within the organization, which is described by its severity and completion level.
Info	A type of Alert fulfilling a specific criterion to be determined later
Injector	A type of Trojan that insert malicious code into processes running on a computer in order to perform various actions, such as downloading additional malware, interfering with web browsing activities or monitoring the user's actions
Internal Database	A database related to industry operations
Internal Personnel	A group of personnel related to the business
Internal Web Server	A web server where the organization itself sets up their own to host the website
IP Detection	An attack that examine a network to see which network devices are on that network
IR Sensor	A tool that emit and detects infrared radiation
Iris Scanner	A device that measures the unique patterns in irises
Key	A small piece of metal used to operate a lock
Local Exositic Virus	A type of virus that does not modify the targeted file but there exists a dependency on the pre-existence of it
Local Exploit	An exploit that requires prior access to the vulnerable system and usually increases the privileges of the person running the exploit past those granted by the system administrator
Local Parasitic Virus	A type of virus that spreads by attaching itself to another program
Local Virus	A virus that requires prior access to the vulnerable system
Location	A particular place or position
Logical Asset	An intangible possession related to business domain

Low	A low severity level
Malware	A software intentionally designed to cause damage to a computer, server, client, or computer network
Media	Means of communication like broadcasting, publishing, and the internet
Medium	A medium severity level
Mitigation Strategy	A set of steps taken to reduce the risk (the severity of the impact and/or probability of the occurrence)
Mobile Device	A portable computing device
Motion Sensor	A tool that detects nearby movement
Network	A group of interconnected systems
Network Equipment	Electronic devices which are required for communication and interaction between devices on a computer network
Network Worm	A worm spread via the network
Normal	A criticality value that the related event does not impact critical components and does not require intervention
Office	A room, set of rooms, or building used as a place for commercial, professional, or bureaucratic work
Office Automation	A software used to digitally create, collect, store, manipulate, and relay office information needed for accomplishing basic tasks
Open Source SW	A software that uses an open development process and is licensed to include the source code
Operating System	A system software that manages computer hardware, software resources, and provides common services for computer programs
Palm Scanner	A device that veins in the user's palm which are as distinctive as fingerprints
Performance	An action or process of performing a task or function
Personnel	A group of people employed in an organization or engaged in an organized undertaking
Phishing	A fraudulent attempt to obtain sensitive information or data, such as usernames, passwords, and credit card details, by disguising oneself as a trustworthy entity in an electronic communication
Physical Asset	An item of economic, commercial, or exchange value that has a material existence
Physical Barrier	A structural obstacle in natural or manmade environments that prevent or block mobility
Physical Security System	A collection of equipment that provides security and safety
Port Detection	An attack that sends client requests to a range of server port addresses on a host, with the goal of finding an active port and exploiting a known vulnerability of that service

Power System	A network which consists of generation, distribution, and transmission system. It uses the form of energy (like coal and diesel) and converts it into electrical energy
Pressure Sensor	A tool that measures pressure of gases or liquids
Proximity Sensor	A tool with the ability to detect the presence of nearby object without any physical contact
Rack	A framework, typically with rails, bars, hooks, or pegs, for holding or storing things
Remote Exploit	An exploit that works over a network and exploits the security vulnerability without any prior access to the vulnerable system
Resilience Assessment	A short self-assessment to help identify the scores relating to Resilience
RFID Reader	A device that uses radio frequency waves to wirelessly transfer data between itself and a RFID tag
Security Door	A gate with any range of measures to provide a safe and secure access control
Sensor	A device, module, machine, or subsystem whose purpose is to detect events or changes in its environment and send the information to other electronics
Server	A piece of computer hardware or software (computer program) that provides functionality for other programs or devices
Service Provider	A service provider provides organizations with consulting, legal, real estate, communications, storage, processing
Severity	A value partition that controls the level of severity of an alert
Smoke Sensor	A tool that uses both photoelectric and temperature sensor for maximum protection from flaming
Software	A collection of data or computer instructions that tell the computer how to work
Software Version	A unique set of version numbers assigned to unique state of computer software
Sounder	An audio device used with **Beacon** to represent warnings
Spyware	An unwanted software that infiltrates your computing device, stealing your internet usage data and sensitive information
Status	A value partition that controls the status of an alert.
Storage	A space available for storing something, in particular allocated space in a warehouse
Structured Cabling	A building or campus cabling infrastructure that consists of a number of standardized smaller elements
Switch	A networking hardware that connects devices on a computer network by using packet switching to receive and forward data to the destination device
Tear Drop	A denial-of-service (DoS) attack that involves sending fragmented packets to a target machine. Since the machine receiving such packets cannot reassemble them due to a bug in

	TCP/IP fragmentation reassembly, the packets overlap one another, crashing the target network device
Technical Room	An enclosed space with doors for access/egress inside or outside the tunnel with safety installations
Technological Equipment	A collection of technology assets including mainframe computers, servers, general computer equipment, printers, monitors, hard drives, memory, storage devices and call centres/ACD systems but excluding Flight Simulators
Telecommunication Equipment	A hardware which is used for the purposes of telecommunications
Telephony Provider	A software interface to a physical telephony device (such as a modem) that can be accessed programmatically to perform actions such as dialling a phone number or logging a call
Temperature Sensor	A tool that measures the amount of heat or coldness that is generated by an object or system
Terminal	A device to help accomplish and automate tasks on a computer without the use of graphical user interface
Thing	Local class to represent OWL:Thing within the ontology for organizational purpose
Third Party Software	A reusable software component developed to be either freely distributed or sold by an entity other than the original vendor of the development platform
Threat Propagation Event	A detection of malicious web destinations
Threat Propagation Path	A sequence of steps towards the detecting malicious web destinations
Transport System	A system for the movement of humans, animals, and goods from one location to another
Trojan	A type of malicious code or software that looks legitimate but can take control of your computer
Trojan Horse	See **Trojan**
Ultrasonic Sensor	A tool that measures the distance of a target object by emitting ultrasonic sound waves and converts the reflected sound into an electrical signal
USB Drive Pen	A data storage device that includes flash memory with an integrated USB interface
Value Partition	A logical class to contain all enumeration values used for other concepts
Virtual Client	A computing model that provides desktop virtualization solution to improve limitations associated with the traditional distributed desktop environment
Virtual Infrastructure	A way to share physical resources of multiple machines across your entire infrastructure
Virtual Server	A server that shares hardware and software resources with

	other operating systems
Virus	See **General Virus**
Voice Recognition Device	A device with the ability to decode the human voice
Vulnerability	A weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source
Vulnerability Exposure	A system configuration issue or a mistake in software that allows access to information or capabilities that can be used by a hacker as a stepping-stone into a system or network
Warning	A type of Alert fulfilling a specific criterion to be determined later
Warning Device	A device that signals the occurrence of some undesirable event
Wireless Network	A computer network that uses wireless data connections between network nodes
Worm	See **General Worm**
Hardware	The physical parts of a computer

Appendix D – Assessment Questionnaire

Questionnaire title: Assessment of Airport Security Interoperability Integrated Ontology (ASIIO)
Platform: Microsoft Forms
Responses: 10 anonymous
Average time to complete: 11 minutes and 3 seconds
Status: Closed
Questions:

1. ASIIO utilizes semantic capabilities to enhance interoperability between the different systems working together in airports security environment. Therefore, it aims to cover essential requirements from all involved business fields from security to airports. What is your business field?

   Type: choices with multiple answers
   Choices:

   - Airports and Aviation
   - Cyber Security;
   - Physical Security;
   - Information Technology;
   - Other.

2. Ontologies, among other formats, are generally used to provide a standard way of communication between systems. Several ontologies are already available for cybersecurity and airports separately.
   Do you have a previous experience with any of the following ontologies?

   Type: choices with multiple answers
   Choices:

   - NASA Air Traffic Management Ontology (ATMONTO);
   - Unified Cyber Ontology (UCO);
   - Ticket Ontology;
   - ICARUS Ontology;
   - None;
   - Other.

3. Have you ever used ontologies for supporting your business?
   Type: choices with single answer
   Choices:

   - Yes;
   - No.

4. If you had used ontologies in a business context, what was the application field?

   Type: choices with multiple answers
   Choices:

   - Airports;
   - Cyber Security;
   - Physical Security;
   - ICT;
   - IoT;
   - Other.

5. The following list consists of some concepts that studies showed to be mostly used in the airport cybersecurity domain. Which of these concepts do you think are important to represent your business requirements?

   Type: likert scale
   Points:

   - Not important;
   - Neutral;
   - Important;
   - Very Important.

   Statements:

   - Airplane;
   - Airport;
   - Alert;
   - Attack;
   - Event;
   - Impact;
   - Incident;
   - Passenger;
   - Vulnerability;
   - Physical asset;
   - Logical asset;
   - Personnel;
   - Data or document.

6. Is there any other concept that is very important to your business and not mentioned in the list?

   Type: optional text input.

7. Has ASIIO helped you clarify communication with a particular system? If so, how many systems has ASIIO enabled communication with?

   Type: number input

8. How would you rate the improvement in systems' interoperability when using ASIIO?

   Type: rating

9. How much has ASIIO made the exchanged data clearer and easier to read and understand?

   Type: rating

10. How did ASIIO help understanding the roles and responsibilities of other systems?

    Type: rating

11. As a result of integrating several information sources, a more comprehensive knowledge can be extracted and deduced, which was difficult using the separated systems. Has ASIIO and its extended set of concepts helped clarify implicit relationships?

    Type: choices with single answer
    Choices:

    - Yes;
    - No.

12. As the business grows and adapts to environment changes, new needs might emerge and it would be necessary to add corresponding concepts. Extensibility of the ontology helps with facilitating growth of the domain by being ready to accommodate any further expansions. How would you rate the improvement in extensibility when using ASIIO?

    Type: rating

13. Would you use ASIIO in your future projects?

    Type: choices with single answer
    Choices:

    - Yes;
    - No.

14. Is there any other feedback, suggestions, or remarks you would like to give regarding the developed ontology ASIIO?

    Type: optional text input.