Data protection regulation ontology for compliance

Master of Science in Technology Thesis
University of Turku
Department of Computing
Cyber Security (EIT Digital Master School)
June 2022
Vinko Mlačić

Supervisors:
Seppo Virtanen
Antti Hakkala

UNIVERSITY OF TURKU
Department of Computing

Vinko Mlačić: Data protection regulation ontology for compliance

Master of Science in Technology Thesis, 53 p., 3 app. p.
Cyber Security (EIT Digital Master School)
June 2022

The GDPR is the current data protection regulation in Europe. A significant market demand has been created ever since GDPR came into force. This is mostly due to the fact that it can go outside of European borders if the data processed belongs to European citizens. The number of companies who require some type of regulation or standard compliance is ever-increasing and the need for cyber security and privacy specialists has never been greater.

Moreover, the GDPR has inspired a series of similar regulations all over the world. This further increases the market demand and makes the work of companies who work internationally more complicated and difficult to scale.

The purpose of this thesis is to help consultancy companies to automate their work by using semantic structures known as ontologies. By doing this, they can increase productivity and reduce costs. Ontologies can store data and their semantics (meaning) in a machine-readable format.

In this thesis, an ontology has been designed which is meant to help consultants generate checklists (or runbooks) which they are required to deliver to their clients. The ontology is designed to handle concepts such as security measures, company information, company architecture, data sensitivity, privacy mechanisms, distinction between technical and organisational measures, and even conditionality.

The ontology was evaluated using a litmus test. In the context of this ontology, the litmus test was composed of a collection of competency questions. Competency questions were collected based on the use-cases of the ontology. These questions were later translated to SPARQL queries which were run against a test ontology. The ontology has successfully passed the given litmus test. Thus, it can be concluded that the implemented functionality matches the proposed design.

Keywords: Data Protection, Knowledge Modeling, Ontologies, GDPR, Regulatory Compliance, Privacy, Cyber Security, ISO/IEC 27001, OWASP ASVS, Business process

Acknowledgements

I would like to extend my sincerest thanks to my university supervisors Seppo Virtanen, PhD and Antti Hakkala, PhD for their guidance and patience. Also, many thanks to my company associates Toby Moncaster, PhD and Jovan Stevović, PhD for their feedback, support, and understanding.

Additionally, this endeavor could not have been possible without the generous support from EIT Digital Master School who financially supported my studies.

I am also grateful to my girlfriend Carolina Gutierrez Bolaños, MSc for her editing skills and inspiration. Thanks should also go to my classmates and friends for their moral support.

Lastly, I would be remiss in not mentioning my family, especially my parents, sister, and grandparents who impacted and motivated me.

Contents

List of Figures

List of Tables

List of acronyms

ASVS OWASP Application Security Verification Standard

AWS Amazon Web Services

BYOD Bring Your Own Device

CCPA California Consumer Privacy Act

DPA Data Processing Agreement

DPIA Data Protection Impact Assessment

GDPR General Data Protection Regulation

ISMS Information Security Management System

ISO International Standardization Organization

LGPD Lei Geral de Proteçao de Dados

OWASP The Open Web Application Security Project

OWL Web Ontology Language

PIPL Personal Information Protection Law

RDF Resource Description Framework

STIG Security Technical Implementation Guide

1 Introduction

Nowadays, data protection and privacy are becoming increasingly important where they once were an afterthought. As a result, we are seeing countries globally introducing new data protection legislations. The new laws aim to protect the rights of the individual, but they can vary greatly between each other.

Most notably, **GDPR** [1] (the "toughest privacy and security law in the world" [2]) has made a significant impact on the world of data processing and data protection. Although an EU regulation, GDPR affects every company around the globe as long as they process personal data of EU citizens.

Not long after GDPR, California Consumer Privacy Act (**CCPA**) bill was signed by the California's governor Jerry Brown [3]. The CCPA gives Californian citizens similar rights as GDPR. However, CCPA is very different and more limited than GDPR.

The awareness of people about privacy risks has, surprisingly, even arrived to China where the Personal Information Protection Law (**PIPL**) [4] was finally adopted on August 31st, 2021.

It has already been a couple of years since GDPR came into effect, and companies have shifted course to take more care about the data protection and privacy rights of an individual. In the beginning, GDPR was widely ignored, but after the Data Protection Authorities started fining these companies, they have swiftly changed their strategies [5]. Now, the companies are aware that this is something that is not

going to go away, and to maintain their success they need support. To follow these regulations many companies turn to cyber security consulting companies.

1.1 What is Compliance in Cyber Security?

Consultancy in cyber security compliance is an amalgam of two domains: **tech** and **legal**. This symbiosis is necessary for making the companies satisfy all of the requirements imposed by a regulation. The requirements are seldom simply signing documents that say that you accept the risk in case a breach happens. They are much more complex than that.

First and foremost, companies need to ensure that they have implemented proper technical measures. Defending systems is more difficult than attacking them. The so-called "attack surface" [6] is relatively big and cyber security experts who are tasked with protecting it have to protect the whole surface. On the other hand, the attackers who are targeting a system only need to find one weak spot to exploit it. For this reason, a palette of security measures need to be implemented. These include (among others) using state-of-the-art encryption mechanisms, using firewalls, following various security principles such as "least privilege" principle [7] or "need to know" principle [8].

Solely technical security measures cannot be relied upon to protect the data. Organisational measures are equally important. According to a 2021 cyber security report from Proofpoint [9]: "Nearly 25% of all attack campaigns hid malware in compressed executable files, which run only after the recipient interacts with it.". This means that almost a quarter of all attackers in 2021 relied on the chance that a human will make the mistake of executing a malicious file. Technical security measures cannot stop insider threats like these. Organisational measures like having password length policies, not accepting emails from outside of the organisation, using password managers, having periodical cyber security trainings, "bring your

own device" (BYOD) policies, and others are the key to support technical security measures and truly protect the data in the system.

Finally, being compliant to certain data protection regulations and standards is not something companies do only because they have to. Being compliant with these regulations builds trust between the company and its customers - who can be other businesses or individuals. Businesses see that the value in compliance is in avoiding fines and delivering products and services that are more trustworthy. This makes it easier for them to find new customers and to market themselves better. That being said, depending on the regulation, cyber security compliance can be a hard goal to achieve. For example, GDPR requires the fore mentioned security measures to correspond to the current state-of-the-art. Furthermore, to-be-compliant companies are required to fill out their record of processing activities which enumerates all of the data they process and the reasons why they are processing them. Also, they need to list their data processors (providers - e.g. MailChimp) and sign Data Processing Agreements (DPAs) with each of them. Then, depending on the type of data they process, a Data Protection Impact Assessment (DPIA) which includes a risk assessment, might have to be made. All of these things require time and money, especially if it turns out that the company has been doing a lot of things incorrectly, and the risks are unacceptable. In this case, the company would have to fix these things as soon as possible or risk a fine.

1.2 Problem Statement

According to a report from Fortune [10], the cyber security market is going to triple by the year 2028. Fortune lists as one of the driving factors the "Increasing Government & Private Investments in Advanced Cyber Security Solutions". This government and private interest is what resulted in new regulations and more work for companies that deal with cyber security compliance. As the market grows, the

consulting companies need to scale their capabilities or risk opportunity costs. This is why automation and efficient knowledge sharing are paramount.

Automation in compliance consultancy is a complicated task. Consultants need a way to organize knowledge to reuse it. As the knowledge in this case is cross-domain, it is no surprise that it is usually unstructured. For consultancy companies, an organized knowledge base is part of their knowledge capital. The main problems to tackle are communication (having a single source of truth), reusability, maintainability, and extensibility. According to an excellent guide from Noy and McGuinnes [11], an ontology solves exactly these problems. Although it might seem like an over-engineered solution at first, only an ontology actually solves all these problems. Solutions as databases or files in which you store company's knowledge can be instances of a knowledge model, but they are, essentially, designed to store data - not knowledge.

Often, many copies of one single statement are spread across several files without any linking or referential integrity. This type of system does not ensure that there is only one single source of truth. One could update a statement in a particular place, but how can we be sure it only appears where we found it? This type of unstructured knowledge is not reusable. Usually, the same knowledge is a part of many services a company offers. It is a terrible waste to have consultants do work that is mostly copy-paste.

1.3 The Solution

In this thesis, a new knowledge model is proposed. This knowledge model (or ontology) should contain the knowledge possessed by the data security expert who assesses the state of a company which processes some kind of data. The outcome of the thesis work is a definition of the ontology. The domain of the ontology refers to data protection regulations, security standards, and best practices. It is

important to stress that the aim is not to create a solution that helps companies become compliant. Rather, it is to develop a solution for consulting companies that optimizes their process.

We explore the creation of a flexible ontology which can answer questions about which security measures are requirements of a certain regulation and why are they needed. A possible application of this could be to create checklists (sometimes called runbooks) which consultants deliver to the customer. The solution streamlines their work and reduces the time they waste. Moreover, a knowledge base also creates a single source of truth. Additional benefits include: improved communication between consultants, faster onboarding of new employees, and the ability of hiring junior consultants with confidence.

The rest of the thesis is organized as follows. Firstly, relevant data protection regulations and standards are presented in chapter 2. This chapter is important because exactly these regulations and standards are the content of the ontology - the main work of the thesis. From these documents, the vocabulary and the semantic relationships have been analysed.

Then, an overview of knowledge modeling is given in 3. This is the second part of the theoretical background that needs to be presented so the reader understands the following chapters.

In chapter 4, an overview of related work is given. This is to show the differences and similarities between the related work and the thesis work.

After this, the methodology of how this ontology has been created will be discussed in chapter 5. This chapter will explain how was the data which was presented in the chapter 2 translated into a machine-readable format. It will also clarify how can the technical solution of this thesis be evaluated. This will be the base premise for drawing conclusions in chapter 8.

Then, the focus is taken to the core of this thesis by explaining the ontology

structure in chapter 6. This chapter explains the technical solution created as the work of this thesis. Also, the chapter draws a parallel to the previous chapter (chapter 5).

The evaluation of the knowledge model is presented in chapter 7. The used evaluation framework was already presented in chapter 5.

The final part is the summary of the results and a reflection on the current state of the art (already presented in chapter 4). This also includes some insights about compliance and ontology development that have come up during the thesis work.

2 Overview of Data Protection Regulations and Standards

In this chapter, I will explain which frameworks were used as content of the ontology and why. The main building blocks of the ontology include the knowledge about technical security measures, organisational security measures, and a regulation. The ontology is meant to be flexible enough to support any number of standards and regulations. A fully working application would probably include several regulations and several security measure standards to be as complete as possible. However, in the scope of this work, including all of these is impractical and it would take too much time.

Therefore, the focus was on GDPR [1] as the regulation, ISO 27001:2013 [12] as the framework which defines organisational security measures in a business context, and OWASP Application Security Verification Standard 4.0.3 [13] framework as the technical security measure guideline.

Each of these building blocks serve a purpose. To answer the question of which security measures are requirements for a certain regulation, consulting security standards is the only trustworthy option. Therefore, they have to be the content of the ontology. Similarly, answering why is a certain requirement required for this regulation needs to reference some legal text from the regulation. So, regulation content, at least partially, needs to be included into the ontology.

2.1 Literature Review of Regulations and Standards

As explained before, focus on the right ontology content was required to create a practical solution. How to choose which regulation or standard to model?

2.1.1 Regulation

Regarding the regulation, there is not a clear choice which one to choose to model. The choice of the regulation affects mostly the application of the ontology, not its core design. Therefore, there are no right regulations to model, they are modeled according to the need. I have chosen GDPR, primarily because the need for this application seems abundant. Secondly, GDPR is applicable anywhere in the world as long as some party collects personal data of EU citizens.

Furthermore, there are various regulations that are very similar or based after GDPR. These include: Australia's Privacy Act and Brazil's Lei Geral de Proteçao de Dados (LGPD) among others [14]. This makes GDPR a good choice to demonstrate the functionality of the current thesis work and a good ground for extension in the future work. Finally, the choice of GDPR was also biased by my experience of working with it. I am more familiar with GDPR than any other regulation.

2.1.2 Technical Security Measures

Second basic block of the ontology is the technical security guideline. Technical security guideline should contain practical security measures which are related to a particular technology. This could be an Android smartphone, Django Web server, React.js website, AWS deployment, and so on. For the same reasons as listed above, it is impractical to model all of these. Thus, I was searching for a technical guideline which is broad and complete to achieve maximum utility and conciseness of the ontology.

The only constant in the world of technical security measures is that they are always changing - not unlike the technology for which they are written. That makes the work of writing and maintaining these guidelines difficult. To make a comparison with the security standards which are focused on organisational security measures - ISO 27001 is making a new version this year after 9 years. On the other hand, technical security measures get changed on a daily basis. US's Department of Defense has a myriad of products which are meant to track exactly these changes and give best guidelines (so-called STIGs [15]) on how to mitigate new vulnerabilities.

In the course of research, OWASP Application Security Verification Standard [13], US Department of Defense's Security Technical Implementation Guides (STIGs) [15] and Mitre's ATTACK framework [16] were reviewed. The only framework that fit the criteria was the OWASP ASVS. The DoD STIGs were too detailed and tied to specific technologies. They could be a good source of data for extending the ontology, but for the first model, they are an overkill. The Mitre's ATTACK framework is as its name says - attack oriented. It maps the security measures as a thing of secondary importance.

2.1.3 Organisational Security Measures

The final block of the ontology is the organisational security measure standard. The organisational security includes writing policies (and sticking to them), managing people, and managing risk. The main purposes of these standards is to remove the human error. As such, the knowledge of organisational security does not change as frequently as the knowledge of the technical security measures. Therefore, there is a plethora of standards which are complete and thorough for use in this ontology. Throughout the literature overview part of the thesis, several standards were reviewed to figure out which one would fit best as part of the ontology. As it was already discussed, the ontology should support any standard or at least be adapt-

able enough so new standards can be appended to the existing structure. Thus, the choice of the standard which we model first is relevant because the abstract model will be extracted from this first standard. The reviewed standards include SOC 2 [17], HITRUST Cyber Security Framework [18], ISO/IEC 2700X family of standards [12], and NIST Cyber Security Framework [19].

The first part of the review process included searching for the trends as the modeled standard should be something that is needed. The trends in the last five years of the reviewed standards are shown in the figure 2.1 below.

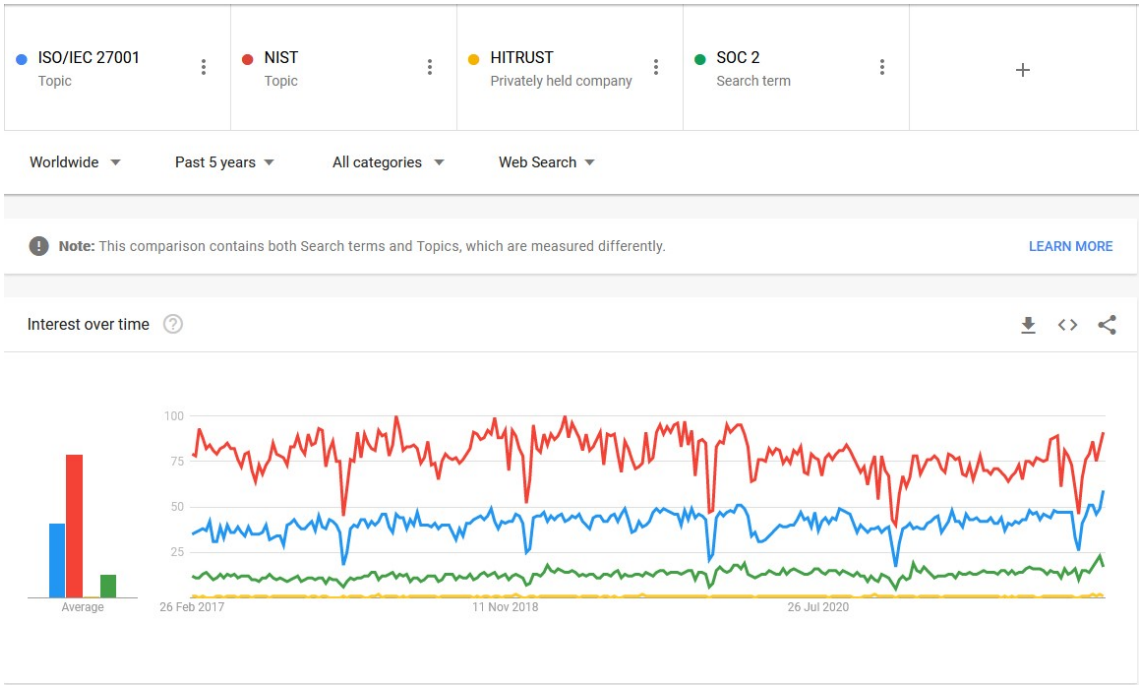

Figure 2.1: Comparison of organisational standards trends. Source: Google Trends

As the figure shows, the most growing interest among the reviewed standards is the NIST Cyber Security Framework. However, after deeper review, the NIST CSF does a lot of mix and match between technical security measures and organisational security measures. This would mean that if separate security measures in the ontologies are separated as technical and organisational, separating them as disjoint

classes is not possible because some of the security measures would overlap between categories. To allow the reasoner to work optimally, having as many disjoint classes as possible is important. The reasoner works on an Open World Assumption and by using disjoint classes we manage to create a "closed world" which is more useful. For more information about the Open World Assumption, please refer to subsection 3.3.5.

The other two security standards: SOC 2 and HITRUST were not as popular as ISO/IEC 27001 so they were dropped as the first standard. However, they have advantages on their own. For instance, HITRUST cyber security framework is completely mapped to other frameworks and would easily fit into the ontology.

2.2 GDPR

In this section, I will present the GDPR as it is defined, explain how complicated is the process of becoming GDPR compliant, and go over the most important concepts that are modeled in the ontology. Explaining this is important for understanding the abstract structure of the knowledge model which is not only particular to GDPR.

The GDPR compliance process starts with a step that is, actually, not included in the ontology - the record of processing activities [20]. Record of processing activities requires the company to list all data points they process, why they are processing them, and on what scale are they processing them. This part is not included in the ontology because the domain of the ontology does not include a legal component. However, knowledge about company type and data sensitivity is included which is analogous to the knowledge in the record of processing activities albeit not so exhaustive. The record of processing activities is the basis for the assessment of what level of security is required for the protection of the data. The GDPR intentionally vaguely states: "... the controller and the processor shall implement appropriate technical and organisational measures to ensure a level of security appropriate to

the risk ..." [1]. This is done because the risk of the data breach changes in regards to data sensitivity and the amount of data. For this reason, the record of processing activities is the starting point for the GDPR compliance process.

After this, the company needs to implement adequate security measures to protect the data.

Generally, this is the final step for many companies. However, there are some types of processing that require additional assessments. These are what GDPR refers to as "... likely to result in a high risk to the rights and freedoms of natural persons..." [1]. If the company belongs in this category, then it needs to do a Data Processing Impact Assessment (DPIA). The final output of this assessment is a document which describes what data is considered risky to process, explanations on why this data is necessary to process, assurances that the data is protected well enough to reduce the risk of processing, and the demonstration of the legitimate interest made by the controller. This is, of course, a simplified version of what needs to be in the document. In this case, the company that is trying to get GDPR compliant, needs to take special care about what data are absolutely necessary to process and to protect them adequately.

This reinforces the argument that the type of data the companies process affects the security measures that need to be implemented. For this reason, having type of data processed in the ontology is imperative. Otherwise, answering the question of which security measures the company needs can be very difficult.

None of the security standards or guides require privacy mechanisms which are required by the data protection regulations. However, these mechanisms are at the least encouraged by the regulations. For instance, GDPR [1] even explicitly lists the pseudonymisation technique in the article 32. as something that increases the security level of the stored data. Not only is pseudonymisation encouraged, but GDPR makes a difference between pseudonymised data and anonymised data. This

makes a huge impact on the security measures that need to be applied and the overall software architecture.

Moreover, GDPR does not apply to anonymised data, but applies to pseudonymised data. On the other hand, the US data protection regulation for health systems, HIPAA [21], does not apply to de-identified data. The definition of de-identified data in the context of HIPAA [21] follows the definition of pseudonymised data under GDPR. Strictly speaking, they are different concepts. Deidentification is the process of removing identifiers from a data set, and pseudonymisation is replacing the sensitive data with a pseudonym. For this reason, they are modeled as separate classes.

Reviewing the GDPR process has been very insightful in the way that I discovered three new concepts that need to be represented in the ontology. These are the **type of data processed**, the **type of company**, and the **privacy mechanisms applied**. To correctly answer the competency questions set out by the ontology requirements, it is important to model these concepts as well.

2.3 ISO/IEC 27001

The ISO/IEC 27001 [12] was chosen as the model standard because it is focused purely on the organisational measures. In my research, I have found that a number of companies are misguided into thinking that the standard makes any assurances security-wise. Later, they get disappointed because the certification officials mostly make sure that certain policies are in place and that they are applied according to the risk company is facing. This makes it good for the organisational part of the ontology. Another point why ISO/IEC 27001 was chosen is because the certification process is less dependent on the type of company and the auditors opinion.

Currently, a new edition of ISO/IEC 27001 is being developed. Its updated code of practice ISO/IEC 27002:2022 has already been released and the new ISO/IEC

27001:2022 is set to follow. However, these documents were not available to me at the time of writing and because of this I have had to use ISO 27001:2013.

The ISO/IEC 27001 helps companies create an Information Security Management System (ISMS). An ISMS is a set of policies and governance over the information security of an organisation with a top-bottom approach. Top-bottom approach means that the responsibility of ensuring the claims in the standard is primarily on the upper management.

The first part of the standard contains the requirements for the company to be certified as ISO 27001. These are separated into 10 clauses. Clauses 0 to 3 define the common language, and others are mandatory requirements. The other part of the standard provides 114 security controls a company has to implement to be compliant with the standard.

2.4 OWASP ASVS Framework

OWASP ASVS framework is a basis for testing application security. It is made to be used as a measure of the level of security confidence in the application. This is why the security measures are separated into several levels. The first level is meant to be ensured by any application. The consequent levels are needed by applications that handle data with more risk.

The first level in the standard has around 130 requirements. Therefore, the standard is quite difficult to satisfy, even for the basic part.

Unlike the organisational security measure standard that was used (ISO/IEC 27001), the OWASP ASVS is very technical and specific. This means that it changes much more frequently.

The standard was chosen because it covers a wide range of security weaknesses in a comprehensive and reasonably rigorous way. It also can be exported in a machine-readable format (JSON) which was very useful for programatically creating

ontological structures from this.

In this thesis, the version 4.0.3 was used which was the latest at the time of writing.

3 Knowledge Modeling

Knowledge modeling is the backbone of the thesis work. Thus, explaining the theoretical background is very important for understanding the relevance of the thesis. To understand the theory of knowledge modeling, we start by integrating definitions of knowledge, breaking it down to its types, and finally specifying what type of knowledge is being represented in this thesis work.

In this chapter, first we go over some format definitions of knowledge modeling and ontologies. As the words "knowledge" and "ontology" are used in different context, it is necessary to clarify exactly what is being discussed in this thesis. Later, some ontology metrics are presented. Then, a description of the Web Ontology Language is given to provide theoretical background for the following chapters. Finally, an overview of some applications which are made using knowledge modeling and ontologies is presented.

3.1 What is Knowledge?

The Blackwell Guide to Epistemology [22] defines knowledge as a relation between a conscious subject and the reality [22]. Davies [23] gives a disambiguation between different types of knowledge: explicit, tacit, and implicit. According to Davies [23]: "Explicit knowledge is knowledge that the knower can make explicit by means of a verbal statement; implicit knowledge is knowledge that is not explicit." The definition of tacit knowledge is slightly more complicated; tacit knowledge is knowledge

that could be made explicit, but it is not because of some reason. Tacit knowledge highly depends on the context. Davies [23] gives an example of tying a shoelace to explain tacit knowledge: we need to know that we need to move our hands and the shoelaces in *this way*. So, in this example, part of the knowledge is made verbal, and part is indicated by the context.

Ontology is an example of explicit knowledge.

From a philosophical perspective, an ontology is a study of all being and existence. Smith [24] gives a definition of ontology from this perspective as: "Ontology seeks to provide a definitive and exhaustive classification of entities in all spheres of being." This discipline focuses on the fundamental questions on the meaning of entities, their similarities and differences.

The definition of ontology as used in this thesis is more narrow. It views ontology (or a knowledge model) as a product which explains a domain of knowledge in an explicit way. The most common definition of this type of ontology is the one given by Studer et al. [25]: "Ontology is a formal explicit specification of a shared conceptualization."

3.2 Ontology Metrics

Course [26] presents three language metrics of the ontology: expressivity, semantics, and mathematical rigour.

Expressivity refers to the number of entities and relations defined in the ontology. For example, a small taxonomy could be considered an ontology of very low expressivity. Taxonomies by definition do not contain any relations except the subsumption relations. Therefore, they are usually low expressivity ontologies.

The semantics refer to the clarity of meaning of the ontology. Naturally, this is a highly subjective metric, which is a common problem in measuring ontologies.

Mathematical rigour defines the coherence of representation which depends on

semantics. Ontologies which are highly mathematically rigorous are called heavy-weight ontologies.

3.3 Web Ontology Language (OWL)

There are different ways to represent an ontology. The representation can be more or less formal. Less formal representation are easier for humans to understand (e.g. graphs, mind maps), but it is hard or impossible for the computers to understand them. By contrast, more formal ontologies are more easily understood by the computers. These include coded representations of ontologies.

In this section, I will go over some ontology concepts and how are they represented in Web Ontology Language (OWL).

3.3.1 Classes, Individuals, and Inheritance

Classes (sometimes also called kinds, types, or concepts) represent abstract models of something that we're collecting knowledge about. An ontology that collects knowledge about different types of boats might have classes such as a yacht, sailboat, speedboat, etc. as its classes.

Class is a basic building block of an ontology. Classes can have individuals. These are instances of a class. In the boat ontology example, a sailboat might be a class while a particular sailboat model might be an individual of that class. Whether a concept is an individual or a class depends on the application and design of the ontology.

3.3.2 Relations

Another basic building block of an ontology is a relation. Relations connect classes or individuals. Relations are normally defined by the ontology designer except in the

case of subsumption relations. Subsumption relations are "is a" relations. Sometimes, they are also called inheritance relations. For example, class *Sailboat* has a subsumption relation with class *Boat*. This means that all individuals of class *Sailboat* are also individuals of class *Boat* through inference.

Custom relations are represented in OWL with object properties. In OWL, all relations have a direction. This means that the ontology can be represented using a directed graph where the classes are nodes and relations are edges. Incidentally, this is the most common way of visually representing ontologies. The direction of a relation is represented in OWL by specifying its domain and range. Domain is the origin of the relation and range is the target of the relation.

3.3.3 Properties

Relations can be further specified using property characteristics and restrictions (explained in the subsection 3.3.4). In OWL, there are seven property characteristics that can be specified for an object property:

- Functional. Properties marked as functional specify that an individual can relate to at most one individual using this property. An example of a functional property would be *hasMainSail* between classes Sailboat and Sail. Since a sailboat can only have one main sail, *hasMainSail* property has to be functional.

- Inverse functional. Properties marked as inverse functional specify that their inverse counterparts are functional.

- Transitive. If a property is marked transitive, that means that if class X relates to Y and Y relates to Z using that same property, X is related to Z through inference. Note that if a property is transitive, it's inverse property is also transitive. Moreover, transitive properties cannot be functional at the same time because through inference the origin of the relation might become

related to other individuals which could lead to unwanted inconsistencies in the ontology.

- Symmetric. Properties marked symmetric specify that if a relation is applied in one direction, it also applies in its inverse direction. For example *hasSynonym* relation applies in both ways.
- Asymmetric. Properties marked as asymmetric cannot be applied in both directions.
- Reflexive. Properties marked as reflexive can relate to the same class.
- Irreflexive. Properties marked as irreflexive can only relate individuals of different classes.

Object properties are not the only type of property. There are also annotation properties and datatype properties.

Annotation properties specify the metadata (data about data). These are commonly used for giving comments and describing in more words the meaning and intention of why a certain part of the ontology was designed in a certain way.

Datatype properties are properties which relate an individual of a class to a certain data value. For example, class *Boat* might have a datatype property *hasBoatLength* which could describe in meters the length of the boat. Datatype properties can only have the functional property characteristic (see above).

3.3.4 Restrictions

Other than properties, relations can be further described using restrictions. There are four types of restrictions that are important to explain: existential restrictions, universal restrictions, *has-value* restrictions, and cardinal restrictions.

Existential restrictions are specified in Protegé using the "some" keyword. Existential relations define that if a class has individual, then that individual must have

properties defined by existential relations. However, if an individual has properties defined by existential relations, we cannot infer that this individual is part of that class (due to open world assumption, see subsection 3.3.5).

Universal restrictions are specified in Protegé using the "only" keyword. Universal restrictions defined that if an individual has certain properties defined by the universal restrictions on a class, then that individual must be part of that class. In this way, reasoner can automatically infer the members of a class and see mistakes where individuals have been categorized erroneously. Having this ability of using the reasoner is one of the major benefits of using OWL language.

has-value restrictions are restrictions which constrain the relation to only be applicable to a specific set of individuals.

Cardinal restrictions constrain the relations to have only a certain amount of relations between specified classes. There are three types of cardinality restrictions: "min", "max", and "exactly".

3.3.5 Open World Assumption

Open world assumption is one of the principles of OWL. In a nutshell, an individual cannot be classified to belong to a certain class because it has all properties defined on a class. This needs to be specially specified (by using universal restrictions). The open world assumption states that there might be other classes that can also defined these properties and the individual can belong there, too.

Simply put, if a statement is not true, it cannot be inferred that it is false. In terms of ontologies, the reasoner does not have the complete knowledge of what things might be in the ontology. Even future additions or changes are taken into account. This is why, a good practice in ontology development is to apply closure axioms whenever possible. For more information on closure axioms and how were they used in this ontology, see section 5.4.

3.4 Ontology Applications

Ontologies can be used for any application that has clearly defined vocabulary and/or process.

Applications can be very diverse and their purpose can vary, too. Ontologies can be used for formalization - creating a standard vocabulary which can serve as a consensus between the words and their meaning in a particular context. An example of this are medical applications. Medical applications usually create vocabularies in an ontological format which can be used for medical applications. In chapter "Semantic Technologies in Drug Discovery" of "Systems Medicine" [27], a survey of numerous medical ontologies has been made. This just speaks to the sheer number of available ontologies for a single field in medicine.

Knowledge modeling can also be used for recommendation systems. For example, knowledge modeling is intensely used by the Netflix team to run their recommendation system.

4 Related Work

In this chapter I will go over some of the work that is related in some way to this thesis. Firstly, it is necessary to present the guides which are the fundamental part of how this work got made. The two most important guides used are A Practical Guide To Building OWL Ontologies by Matthew Horridge from Manchester University [28] and Ontology Development 101: A Guide to Creating Your First Ontology [11].

Next, there are some other examples of ontologies which dealt with similar domains as this one. There are several works which deal with cyber security, compliance, and using knowledge modeling for solving problems in these domains.

4.1 Earlier Attempts at using Ontologies in Cyber Security

In [29], Herzog et al. describe an ontology which connects technical terms with general terms related to cyber security like threats, vulnerabilities, and countermeasures. This type of ontology is meant to be consulted as an encyclopedia. While it is somewhat related to this work, it is more important to present it as a contrast because it is more focused on describing the knowledge than on possible reasoning features of the ontology. Fenz and Ekelhart make a similar attempt [30] in formalizing information security knowledge with a highly detailed ontology (500 concepts and 600 formal restrictions). This ontology is based on ISO 27001, IT Grundschutz

Manual, and the NIST handbook.

In [31], [32], Fenz et al. describe an ontology for information security but more focused on the risk management part of the domain knowledge. This work has been motivated by similar ideas. Fenz et al. started moving away from abstract vocabulary definitions of information security towards a more practical approach which is widely supported through this work as well.

Works by Herzog, Shahmehri, and Duma from Linköpings Universitet, Sweden [29] and some early works by Fenz et al. from the Technische Universität Wien, Austria are attempts at building ontologies which try to model security domain knowledge. This work gives a hint of what is done in those papers and the thesis is strongly focused on modeling the regulatory compliance related to data protection regulations such as GDPR or HIPAA, and standards such as ISO 27001. This thesis relies on these standards and regulations as the foundational knowledge of the ontology. More importantly, the application of the ontology is completely directed for use in compliance processes.

In the next section, I will go over some attempts to model different regulations and standards using ontologies.

4.2 Ontologies with Similar Domains

In [33], Fenz et al. proposes an ontology which is made to support the compliance process for compliance with ISO 27002 (the more technically descriptive standard in the ISO 2700x standard family). In this ontology, Fenz et al. also introduce a formal representation of the organization's assets as part of their knowledge model. This idea is also used in this work in a slightly different way. In this ontology, we simply collect some information about the organization, not the whole list of assets. However, the introduction of this method means that the ontology only works if this information about the organisation is added, too. This adds a certain complexity in

formalizing the ontology, but improves the reasoning features that the ontology can deliver once this data has been added. It also means that updating the ontology (at least the individuals) will be a standard part of its operation.

In [34], Fenz and Neubauer present an extension to earlier work [33] (presented above) which is an ontology aimed at aiding the compliance determination. This shows how the ontology needs to be designed starting from the application. Furthermore, having ontologies which are difficult to extend is not good enough. It is necessary to be able to modularize new parts and have a core knowledge part which does not depend on anything else.

In [35], Joshi et al. describe an ontology which describes HIPAA privacy and security rules. The ontology is meant for privacy and security control selection similarly to the ontology in this thesis.

Bartolini et al. in [36], make an attempt to represent the duties of a data controller in what was at the time, upcoming GDPR regulation. This is yet another example of an ontology with similar domain.

Maybe the biggest influence of all in terms of the methodology of constructing the ontology and testing it, was the GConsent ontology [37]. The use of litmus test as the testing method was completely inspired by this work. Furthermore, the ontology is based in a similar domain - consent. Consent is a very important part of GDPR and privacy in general. Usually, it is the most practical approach for obtaining legal basis for processing personal and/or sensitive data.

5 Knowledge Modeling Methodology

In this chapter, methodology of how was the ontology created will be explained. Practical knowledge modeling theory was employed for the development of this thesis [26]. The work of Noy and McGuinnes was used as reference [11]. The following sections describe everything that was done in each stage of the development.

The course [26] provided a modern and practical approach on developing ontologies, while the guide [11] was a good reference for any doubts, such as, choosing when to make a certain term in the ontology an instance or a class.

The main tool used in the development of the ontology is the Protegé 5.5.0 [38] and the HermiT reasoner [39] which was used for consistency checking.

There are several phases in the knowledge model development. As a first step, goal and scope of the ontology needs to be defined. Here, the domain and vision of an application are added. In this phase, the list of competency questions was created. This list will be used with the objective of describing the possible application of the ontology and for evaluating it later. Second phase is the information gathering. The information gathering is already described in chapter 2 where the knowledge sources have been presented. The concepts were gathered and taken into an initial list of terms that need to be included in the ontology. In the third phase, an attempt is made to reason out which concepts and relations should be added as class, property, or discarded completely. By doing this, a simple ontology is created which is depicted using a radial diagram. In the fourth phase, the knowledge model is formalized

through the OWL language. The Protegé knowledge modeling environment was used along with the HermiT reasoner. After formalizing the ontology, it needs to be made available for wide audience so they can re-use, implement, and give feedback to it. The last phase is the evaluation. Here, the list of the competency questions that was created in the first phase is consulted and tested against the created ontology.

Throughout the development of the knowledge model, the requirements are constantly discovered or invalidated. Most of the requirements were collected in the information gathering phase.

All of the mentioned phases are connected with the requirements management.

5.1 Goal and Scope Definition

The goal of this phase is giving purpose to the knowledge model. To do this, it is important to describe some of the most important aspects of the ontology in the early stages of development. These include the ontology's domain of interest, high-level aim, key stakeholders, and the scope of the ontology. The competency questions were defined for the first time in this phase. Some of them include:

- What security measures does a company need to implement in order to be compliant with regulation X?
- Given that the company does not process sensitive data, what kind of security measures does it need to implement to be compliant with regulation X?
- Why does a company have to implement these security measures?

From this sample of the competency questions the main classes can already be extracted. This list only contains some competency questions that were used in the process. For a full list of competency questions, refer to the section 7.3.

5.1.1 Domain of Interest

The domain of the ontology are the security measures and data protection regulations. The knowledge base is expected to answer questions such as "What security measures I need to implement in order to satisfy regulation X?" and similar.

Note that the domain of the regulation does not cover everything that is necessary to be compliant with a data protection regulation. Data protection regulations are vastly more complex than that. Even though this might be related to the ontology that is being created, every aspect of compliance cannot be covered in one single ontology. However, having an extendable and adaptable ontology will build a foundation for future work.

5.1.2 Aim of the Ontology

The aim of this ontology is to be able to determine the security measures a company needs to implement before it can claim it is compliant with a specific regulation. Consultants need to determine which security measures are mandatory, which measures are simply nice to have, and which policies need to be created and implemented on the technical road to compliance. Furthermore, company size and the type of data companies process can have implications on the required security measures. The aim of the ontology is to understand these implications.

5.1.3 Key Stakeholders

Listing stakeholders of the ontology is important because all stakeholders should understand the domain language in the ontology. Furthermore, it is important to avoid confusing terms which could mean different things for different stakeholders.

The main stakeholders of the ontology or the main potential users are the **consultants** and the **companies** trying to get compliant (the customers of the consultants). The former are probably more important because the consultants will

not only refer to the knowledge in the ontology, they will also curate maintain its content.

5.1.4 Scope of the Ontology

The work of this thesis is to create an extendable middle-level ontology. Middle-level ontologies are dependencies of concrete application-level ontologies. It is challenging to test the middle-level ontologies because it cannot be known if the ontology satisfies all functionalities if they are not used.

This is why, as part of the work, an example application ontology has also been created to test the current model.

Also, because of this, the ontology does not assume any security standards or regulations even though the questions that the ontology should answer need to be concrete.

5.2 Information Gathering and Elicitation

This phase requires the gathering of all of the knowledge that will be used in the ontology. From there, content is carefully selected and first classes and properties are listed.

There are several methods of gathering information and they all depend on the nature of the knowledge one is trying to model. The course Practical Knowledge Modeling [26] mentions several: brainstorming, survey, interview, focus group, and content analysis. Given the nature of the data collected in the ontology, brainstorming and content analysis are the only two methods that apply for the current study.

The knowledge sources used in the information gathering phase are described in detail in section 2. In this section, the knowledge sources are analysed from a

different perspective. They are examined from a semantic standpoint. Also, the concrete classes and properties were extracted from the raw unstructured text and a mind map of the preliminary ontology was created.

5.3 Initial Structuring

In the initial structuring phase, a list of the most important terms found in the reviewed material is created. This list is used to classify which terms will become classes, instances, relationships, or completely discarded. The outcome of this phase is to curate this list, and create a visual representation of the first ontology in the form of a graph.

Please find the complete list of the terms used in this ontology work in the appendix A. An overview is also available which shows the design decisions made throughout the development the ontology.

5.4 Formalisation

Formalisation of the ontology was done using the OWL (Web Ontology Language) language.

The OWL language is based on RDF (Resource Description Framework) syntax. The RDF supports logical statements in the form of *Subject - Predicate - Object*. OWL language builds the semantic relationships on top of this. The main concept in the OWL language is called *Thing* and all other concepts inherit from it. Please find below the listing 5.4 of an example of OWL language syntax (XML) which is taken from this ontology.

```
<EquivalentClasses>
    <Class IRI="#SensitivityValuePartition"/>
    <ObjectUnionOf>
        <Class IRI="#Confidential"/>
        <Class IRI="#Private"/>
        <Class IRI="#Proprietary"/>
        <Class IRI="#Public"/>
        <Class IRI="#Sensitive"/>
    </ObjectUnionOf>
</EquivalentClasses>
```

Listing 5.1: Example of OWL language

The formalisation process applied on this ontology can be described in the following steps:

1. Model all classes and subsumption relationships (these are gathered from the initial structuring phase).

2. Model all object properties on all classes. In this step, it is important to mark properties as functional (where possible).

3. Model datatype properties on all classes.

4. Add existential restrictions to all classes.

5. Try to convert as many classes to *defined classes* by adding universal restrictions on all of their object properties.

6. Try to add closure axioms to as many classes as possible.

Steps 1-4 are pretty straight forward. This part basically comes down to inputting the data from the documents created by the previous phases. Steps 5 and 6 are more difficult and opinion-based.

In step 4, existential restrictions are added. According to a practical ontology guide from the Manchester University [28], existential relationships define the necessary conditions of a class. Primitive classes only have their necessary conditions defined. This means that if an individual is part of class A, it must have the necessary properties defined by these restrictions on the class. However, this does not infer that if an individual has these properties, it can be implicitly considered part of class A. Primitive classes require more attention from the knowledge model editor. Also, the reasoner can infer less about primitive classes which can allow logical mistakes to slip in the ontology.

The guide [28] also mentions defined classes. Defined classes have their existential and universal restrictions defined. Universal restrictions are used to define sufficient conditions of a class. A universal restriction defines that if an individual has certain properties, then it must be part of a certain class. A class which has its sufficient conditions defined can have its individuals implicitly assigned to it. This is where the reasoner becomes very useful in automatically assigning individuals to classes, and in checking for logical mistakes when assigning individuals to wrong classes.

In the final step (6), closure axioms (sometimes called closure restrictions) are applied to as many classes as possible. Closure axiom is mentioned as best practice in the guide referred above [28]. Closure axiom is syntactically equivalent of using a universal restriction which is made of a union of all object properties in a class. The purpose of closure axioms is to enable the reasoner to create subsumption relationships for us. This also helps in checking if mistakes have been made in defining a certain inheritance relationship.

Other than these steps, ontological "design patterns" have been applied when

there was a chance to do so. The use of design patterns increases maintainability and extensibility of the ontology. Ontology design patterns are similar to design patterns in software engineering. They are solutions for problems that have been appearing over and over again. More about this and other technicalities will be described in chapter 6.

5.5 Deployment

Deployment is the next phase of the ontology design process. Before anyone can use the ontology, it needs to be made available. The work in this thesis has been split into two ontologies: one middle-level ontology, and one applied ontology based on GDPR, ISO 27001, and OWASP ASVS framework. The point of the separation is to give the ontology best chances to be employed repeatedly. The ontology can be found in the following GitHub repository: `https://github.com/vinkomlacic/cyber-security-compliance-ontology`.

5.6 Evaluation

To assess the functionality and the quality of the ontology, it is necessary to use it. However, evaluation of the thesis work can be done using the *litmus test* [11]. As mentioned before, in the design phase, a list of competency questions was defined. This list is not meant to be exhaustive but it serves the purpose of demonstrating what capabilities the ontology has. We "ask" the ontology these competency questions using the SPARQL [40] query language. If the responses are satisfactory, we can say that the ontology has been evaluated successfully. Another tool used was the Protegé [38] ontology development IDE and the HermiT [39] reasoner to check inconsistencies while developing the ontology. Using the reasoner helps identify logically impossible statements in the ontology.

6 Ontology Structure

In this chapter, the ontology structure is described. Firstly, a concept analysis from the information gathering phase is presented. This is important to understand the core of the domain knowledge that is being modeled in this ontology. Then, a graph is presented which shows the final formalized ontology. These design choices are explained the following section. Finally, ideas are given on how could this ontology be implemented to serve as an actual application.

6.1 Concepts Analysis

In this section, we list statements in natural language which are the core of this ontology's knowledge.

Security measure is a way of protecting a company asset. The assets companies protect are usually data, profits, computers, and so on.

Security measures can be implemented on people (through policies and other types of commitment), or on technology (computers, servers, programming code, etc.). These are jointly called security measure implementation places.

Security measures can be technical or organisational. Technical security measures are implemented on technology, and organisational security measures are implemented on people.

Security measures can come from standards, guides, and regulations. These can jointly be called security measure sources. Each of security measure sources

have their own categorizations of the security measures proposed in their respective documentations.

Security measure is necessary to implement for a certain regulation because that regulation provides some kind of text which requires this security measure to be implemented. This text can be provided through the actual regulation document or one of its associated documents. For example, this would be the technical manual BSI-TR-03161 for the German BSI-100 standard. The security measure can be directly or indirectly required. This comes down to the opinion of the consultants who will interpret the regulation and decide whether a particular security measure is deemed necessary. They can provide a reason why they think a certain security measure is deemed necessary.

Security measures can be necessary only under certain conditions. Such conditions could include the company employee number, if the company processes sensitive data, the company target market size, and others.

A security measure can also be optional to implement. Some regulations make certain security measures simply "nice to have", but not required. In times of auditing, these "additional" security measures get perceived as a sincere attempt of securing users' data. In particular, in GDPR, this could lead to reduced fines in case of a data breach.

Technical security measures are implemented in parts of the company's architecture. It is important to know the company's architecture and how the data flows to be able to determine what security measures are necessary.

The architecture is composed of several components. The architecture component can be a node or a channel. Each of these components can have various security measures applied to them. Also, architecture components process (create, read, update) some kind of data.

The data set processed by the architecture components can be protected by some

kind of privacy mechanism. These include anonymization, pseudonymization, and others. The fact that a privacy mechanism has been applied to a data set makes the security measures less relevant or not necessary at all.

6.2 Preliminary Ontology

After analysing the semantic definitions above, a preliminary ontology mind map was extracted. The mind map is shown below in the figure 6.1.

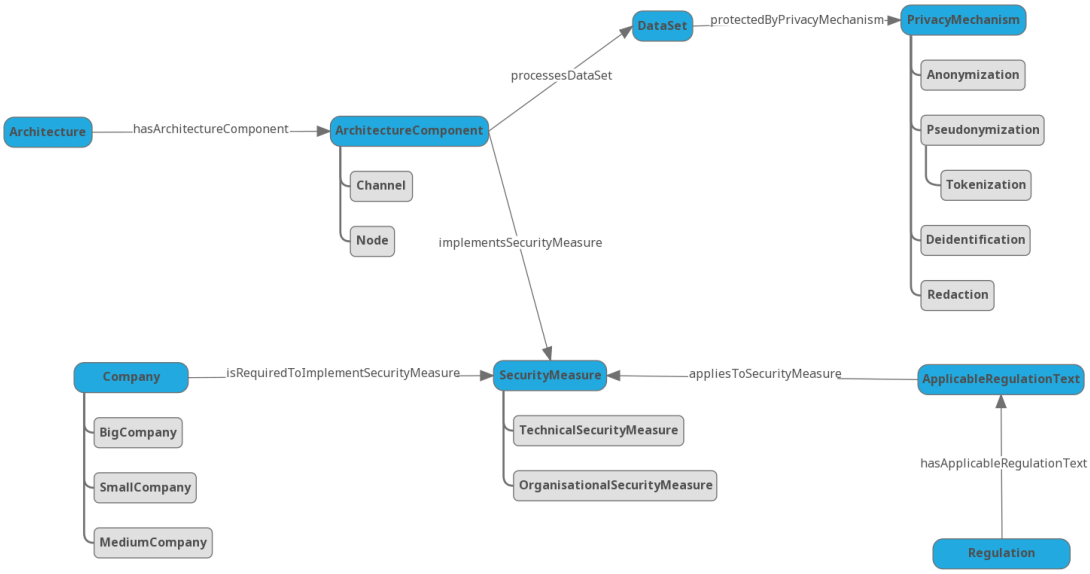

Figure 6.1: Mind map of the preliminary ontology

This preliminary ontology has a lot of shortcomings which will be addressed in the following paragraphs.

In this initial draft of the ontology, the center of the ontology is the *SecurityMeasure* class. Individuals of this class will be the actual security measures the company needs to implement.

The disambiguation between the technical and organisational security measures is not semantically sound because the technical and organisational security mea-

sures will have the same properties as the *SecurityMeasure* class. However, I felt that the disambiguation is needed because these two categories are the common way of separating the security measures. Furthermore, the two security measures come from different sources. It is rare for the same authority to describe both technical and organisational security measures. The reason is usually flexibility of the standard/guide. It is hard to manage changes if both domains are mapped in the same document. Also, the knowledge in these two domains is not exactly the same. It is possible that in the future, changes in the knowledge model are introduced because of this distinction.

Moreover, the distinction between different company types seems oversimplified. It is hard to draw a line between a small, medium, and a big company. Do we separate it using the revenue or employee count? Moreover, the different regulations differently discern what is a small and what is a big business.

6.3 Ontology after Initial Structuring

As it was already described in chapter 5 about methodology, in the initial structuring phase, all of terms and concepts from the concepts analysis above are taken and curated in a table.

In this table, it was decided which of the terms will be kept and which terms need more detailing for the ontology to work. The full list and the decision choices can be observed in appendix A. After this was done, a graph was created which displays all of the concepts and relationships between them. The graph is displayed in the figure 6.2 below.

As it can be observed from the graph, a couple of things changed since the preliminary mind map. After all of the concepts have been thoroughly analysed, a better understanding of how will the ontology look is gained. Then, we can create the initial structure.

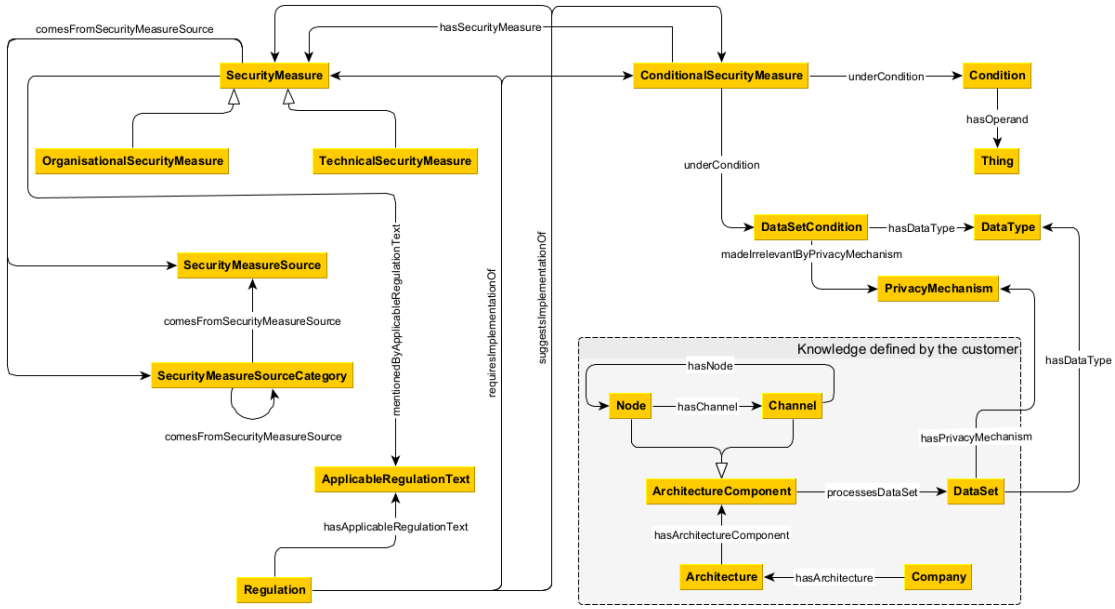

Figure 6.2: Ontology graph after initial structuring (created using yED graphing tool 3.2.21 [41])

The main revelations gained through this process are that a more flexible conditional mechanism is going to be necessary for the ontology to work. Conditions like the company employee number are only *instances* of what the conditions could be. There might be a variety of these conditions that will have to be encoded in the knowledge and it is necessary to give the consultant a way to represent it using the ontology.

This is the whole point behind the *ConditionalSecurityMeasure* class. This class can only have one *SecurityMeasure* because it is conceptually a subsumption of the *SecurityMeasure* class. An actual subsumption could not be used because one *SecurityMeasure* individual can have different conditions, and it can be used without conditions, too. Using a subsumption relation, we would apply a condition to a security measure, and could not use it anymore without this condition for other regulations.

This class has been created to replace the company type information represented in the preliminary mind map of the ontology. It gives much more robustness to the ontology. Also, this class was necessary because OWL can only represent binary relations, not n-ary. This problem has been discussed more in depth in section 6.5 of this chapter.

Another revelation was that the source of this information will have to be the client of the consultancy companies. The client part has been specially marked in the figure 6.2. We cannot encode all possible combinations of architectures the customer might have, nor the possible data sets and privacy mechanisms applied to them. To answer some of the competency questions, this type of knowledge is required.

6.4 Ontology after Formalisation

From the initial structuring, it is possible to proceed to the formalisation phase of ontology design. the ontology was formalised using the OWL language and Protegé ontology editor.

After formalising, a graph can be made. This graph is shown in the figure 6.3 below.

Please note that in the graph, only subsumption (parent-child) relations are shown. This is a radial representation which helps to display how different classes in the ontology are grouped. For example, *ConditionalSecurityMeasure* is very close to *Condition* class which means that they likely work together somehow or there is a dependency between them somewhere.

In the graph, some changes in regards to the initial structuring graph are visible. As it is shown, the *DataType* class now has a sensitivity property which can disambiguate different data types. It was already argumented why this is important for the ontology to work in section 2.2 of chapter 2. From the sensitivity property

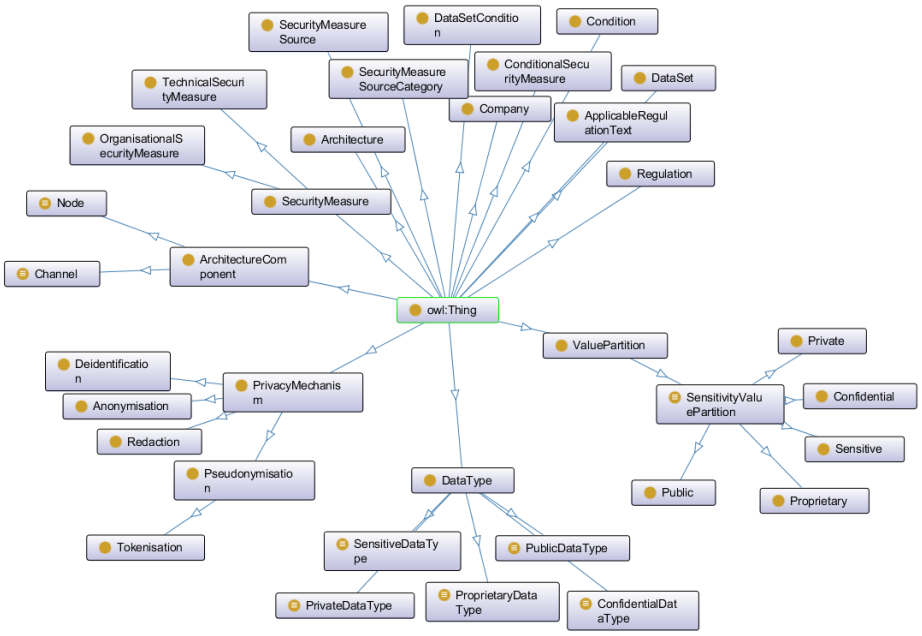

Figure 6.3: Tree radial representation of the ontology after formalisation (only subsumption relations shown)

data types now also have specialized classes so we can use the reasoner to check instances of classes. Having both the sensitivity property and defined classes which can automatically place individuals into the correct class makes the ontology more expressive and easier to work with when creating queries.

The sensitivity property has been modeled using a design pattern called "Value Partition". More on this design pattern will follow in section 6.5 of this chapter.

6.4.1 Cardinal Relations

In the formalised version of the ontology, all relations have cardinality. Cardinality defines the number of individuals that can be on either end of the relation (origin or target). This is analogous to entity relation graphs in database design. Most of the object properties defined on the classes use the existential restriction (keyword

"some" in Protegé). Existential restriction has an implicit cardinality of at least 1 individual.

For example, in this ontology, class *ArchitectureComponent* has an existential restriction on the property *hasChannel* which relates it to the *Channel* class. This means that if an individual is of type *ArchitectureComponent*, it necessarily needs to have at least one *hasChannel* property.

Most relations in this ontology simply use the existential restriction on their properties because it usually makes the most sense and it is not as constraining. However, there are exemptions from that rule. For example, *Node* and *Channel* are two classes that have symmetric relation between them. This symmetric relation is expressed in OWL as two object properties: *hasNode* property on *Channel* class, and *hasChannel* property on *Node* class. These two object properties are inverse to each other, but they are not equivalent. This is because *hasChannel* property on the *Node* class has an existential restriction while *hasNode* property on the *Channel* class does not. *Channel* class has an explicit cardinal relation with the *Node* class. These classes represent the most abstract parts of a computer network architecture - a node and a channel which connects nodes. A node can be connected with as many nodes as necessary through channels. Therefore, one node can have many channels. However, a single channel can only connect two nodes. This is why the *Channel* has cardinality restriction which restricts it to have exactly two *hasNode* properties.

6.4.2 Data Type Sensitivity

Data types sensitivity was taken from [42]. It is more important that the data types classification is context-based (e.g. data sensitivity), rather than content-based. Classification of data is a problem of its own. Content-based data classification requires that all possible content types are enumerated. This is a never-ending exhaustive work. Having context-based data classification is slightly inaccurate and

more subjective, but it can be managed more easily. This is why in this ontology, data types are defined by the consultant according to their needs, but they need to set a certain sensitivity level for every data type they list.

6.5 Used Design Patterns

Design patterns follow the same definition as in object oriented programming. These are solutions to problems that have occurred over and over again.

In this ontology, two ontology design patterns have been used. One of them is related to the n-ary relation problem, and the other to the value partitioning.

6.5.1 N-ary relations in OWL

Throughout the modeling of the ontology, designing conditions on security measures was a particular challenger. The result of this was a ternary relation between *SecurityMeasure*, *Condition*, and *Regulation*. This relation would say that a certain security measure is required by the regulation only under certain conditions. But, how can this be modeled in OWL?

According to [43], OWL exclusively supports binary relations. However, it is possible to create a class which encapsulates an n-ary relation. This is why the *ConditionalSecurityMeasure* was created. This class has connections to *ApplicableRegulationText*, *Condition*, and *SecurityMeasure*.

You can see how this is connected in the figure 6.4 below.

As the graph shows, the regulation can suggest or require either security measure or conditional security measure. Conditional security measure is only suggested/required under the defined condition. These exclusivity constraints are not shown on the graph, but are implemented in OWL.

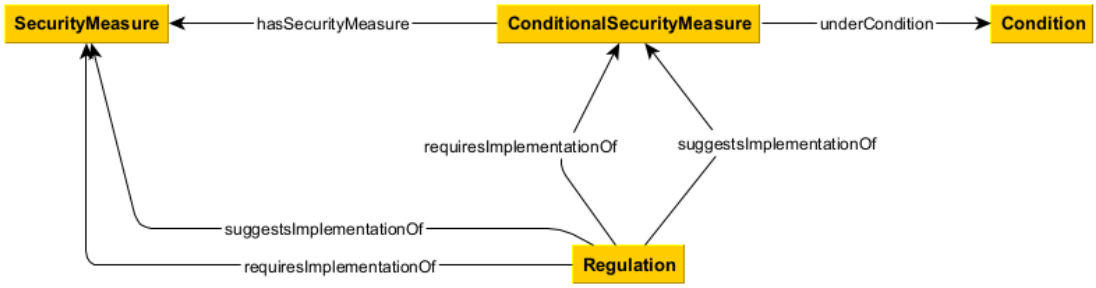

Figure 6.4: Ontology subset: representation of conditionality problem

6.5.2 Value Partition

Value partition is another design pattern used in the development of the ontology. According to [28]: "Value Partitions restrict the range of possible values to an exhaustive list ...". This design pattern can be used to better describe values a certain property can take.

In this ontology this has been used on the sensitivity property to distinguish different data type sensitivities. How this class was implemented is demonstrated in the figure 6.5 below.

This graph shows only the relations between the sensitivity value partition and the data type, but another important part that was done in OWL is the *Covering axiom*. Covering axiom is the main part of the value partition design pattern. This applied covering axiom in this part of the ontology states that the *SensitivityValuePartition* has to be one of its sub-classes. If this is not true, the reasoner complains. This is done to restrict the usage of the value partition class and to make all of the value partitions defined classes.

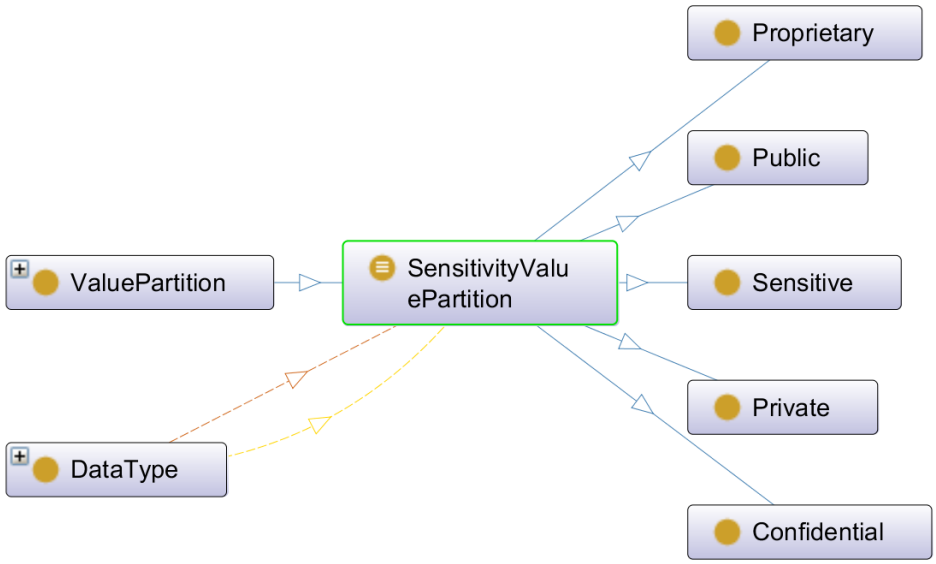

Figure 6.5: Ontology subset: sensitivity Value Partition

7 Evaluation of the Ontology

In this chapter, the method of evaluation of the ontology will be explained in detail. Also, an overview of the competency questions will be given. It is necessary to provide the reasoning behind why were these questions chosen for evaluating the ontology. This chapter will connect the intended use of the ontology with the evaluation methodology.

7.1 SPARQL Query Language

SPARQL Query Language was used in the evaluation of the ontology.

SPARQL Query Language is formatted similarly like the RDF - in triplets of subject, predicate, and object. Through these three values, we can query the contents of the whole knowledge base.

According to the W3C documentation [40]: "SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports aggregation, subqueries, negation, creating values by expressions, extensible value testing, and constraining queries by source RDF graph."

In the listing 7.1, an example of a SPARQL query is shown. The query finds all individuals, their type, and their class. Note that two clauses were used in the format of subject - predicate - object.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT * WHERE {
        ?individual rdf:type ?type .
         ?type rdfs:subClassOf ?class   .
}
ORDER BY ?class
```

Listing 7.1: Example of a SPARQL Query

7.2 Converting Competency Questions to SPARQL Queries

To validate the work of this thesis, it was necessary to prove that the possible application of the ontology would work. To do that, it is common to use a *litmus* test which is composed of competence questions. Similar evaluations were done in other works such as [37].

Converting competency question to SPARQL was at length discussed in [44] by Potoniec et al. In this work, we apply a similar approach. All of the competency questions (see section 7.3) were converted to SPARQL queries and the knowledge base was queried.

To test the ontology, in total 7 broad competency questions were posed. These were converted to SPARQL queries and executed against an extended version of the ontology which was filled with individuals from GDPR as the regulation, ASVS

framework as our source of technical security measures, and ISO 27002 as the source for organisational security measures.

To properly test the knowledge base, it was also necessary to load it with some dummy data about the company and its architecture. Even though the data has been inserted for testing purposes only, they do not affect the outcome of the test.

7.3 Overview of competency questions

In order to properly test the ontology, a litmus test was proposed. Ontology design can be difficult to validate. Article [45] demonstrated the complexity of evaluating the quality of ontologies. According to the article [45], there are many different metrics that can be used to classify the evaluation methods. These include the stage of development, completeness and precision, performance, etc.

The goal of this work is not to formally prove the proposed ontology. Also, the performance of the ontology is not particularly interesting either. The main goal is to prove that the ontology can be used in an application.

This is done using the litmus test. In the context of ontologies, litmus test is a collection of competency questions (listed below). The competency questions were collected based on the use-cases of the ontology. They prove that the information stored in the ontology can be effectively used in an application. A very similar approach was used in an article [37].

The choice of competency questions included the possible use-cases that might be required by the users of the ontology. Please note that the competency questions are meant to be user oriented. Thus, they are simplified versions of the SPARQL queries that are derived from them.

1. What security measures does a company need to implement in order to be compliant with regulation X?

2. Given that the company does not process sensitive data, what kind of security measures does it need to implement to be compliant with regulation X?

3. Given that the company has a huge user base, what kind of security measures does it need to implement to be compliant with regulation X?

4. What policies a company needs to implement to be compliant with regulation X?

5. Given company architecture, where do the security measures need to be applied?

6. Why does a company have to implement these security measures (what part of the regulation X applies)?

7. Where are the security measures coming from (what sources)?

8. Given that the data set is protected by a privacy mechanism, what security measures are required to be implemented?

9. What security measure are suggested to implement for the user given their company information?

The most important use case is to get a list of security measures for a given company. Here, the notion of company includes several sub-concepts like the company architecture, the type of data they process on the architecture components, the flow of the data in between the architecture components. This is all covered by the first question.

Question #2 covers the specific case that might interest the user. However, it is based on the same data that was covered in the question #1.

Question #3 covers the conditionality part of the ontology. It is possible to encode different conditions that might be interesting for a regulation. One of the

examples of this is requiring some security measures only in case the company is big enough. For example, a regulation might consider the company as "high risk" if it has a certain number of employees. This could mean that another, stricter level of security measures needs to be implemented. The conditionality is implemented according to regulation requirements.

Question #4 covers the organisational measure part of the ontology. In section 6.2, it was argued that the distinction between the technical and organisational security measures was needed. This distinction is tested by competency question #4. Organisational measures include the internal and external policies.

Detailed data about the company is also part of the ontology. This enables the user to retrieve very specific information about where to install specific technical security measures. The competency question #5 tests this functionality. The expected result is a list of architecture components where particular security measures need to be implemented. Along with knowledge about security measures and the company architecture, this question tests the knowledge about the sensitivity of the data and the processing places in the user's architecture.

The user might also be interested in the reason why certain security measures are required. Applicable regulation text is also stored in the knowledge base for this reason. This can demonstrate to the user the real need for the security measure. Otherwise, the users might be suspicious of implementing a technically complex security measure. This functionality is tested by the competency question #6.

Another functionality provided by the ontology is integrity of the security measure that are proposed to the user. Instead of creating a custom security measure list, already known and respected security standards are used in the ontology. That being said, it is possible to use completely custom security measure sources instead of the official ones. The question #7 tests this functionality. By giving the source of the security measure, the user can have more confidence in implementing a security

measure which can be technical or organisational.

The question #8 tests the fact that some data sets stored on the user's premises might be protected by privacy mechanisms such as anonymisation. It is worth mentioning that this changes the required security measures in most regulations. The knowledge about a limited set of privacy mechanisms is encoded in the ontology. Combined with the conditionality feature, it is possible to implement conditional security measure that only apply if the privacy mechanism is not implemented.

Some security measures might be simply suggested. This means that they are not required, but help when the time of auditing or a breach happens. It is possible to store this knowledge in the ontology. The competency question #9 tests this part of the ontology.

In conclusion, these competency questions cover all main domains of the ontology: company information (architecture, data sensitivity, processed data), security measures (includes the distinction between technical and organisational security measures), privacy mechanisms, conditionality, and the difference between requiring or suggesting a security measure.

8 Conclusions and Further Work

8.1 Conclusions

This thesis aimed at providing an optimisation tool for regulatory compliance business processes. By analysing GDPR [1], ASVS framework [13], and ISO 27001 standard [12], an ontological structure was designed whose application is to answer questions about the requirements posed by a certain regulation under conditions such as the size of the company, sensitivity of the data that is being processed, and the overall architecture of the analysed product.

It is hard to measure the performance gains a compliance consulting company would have by using this type of solution because a company would have to adopt it first.

However, the utility of the solution can be tested using the competency questions. Having passed the complete *litmus* test, it can be concluded that the ontology serves the purpose laid out in the specification phase.

This research shows how the use of semantic technologies can be used to accelerate business processes in regulatory compliance.

Dealing with ontology design can prove to have some limitation mostly due to the availability of the tools. The tools used for developing the ontology in this work were mainly developed by universities and research institutes and some of them are not actively maintained. This speaks to the problem of traction the ontological

researches have and to the fact although that the field has been around for decades, it is still slowly developing.

Another important finding regarding the ontological development is that having the right knowledge development process is very important for developing an ontology that is relevant and answers the user need.

8.2 Future Work

This section will cover possible extensions of the created ontology.

Firstly, this ontology does not cover at all the concept of risk which is one of the core concepts in security. The ontology could be extended adding this part so we can order the security measures according to the risk assigned to them. Although this is a common request, technicalities around the risk assessment and assignment made it too complex to be added in this ontology. This is why it is mentioned here as a possibility for future work.

Furthermore, the user might be interested about the difficulty of implementation of different security measures. This is as impactful as the risk part presented in the previous paragraph. Product managers always have to weigh the complexity of building a feature versus the value that the feature will add. Having this kind of data could wean them off to a less data risky feature that does not require a security measure that needs a lot of effort to build.

This ontology can be expanded in legal parts, too. The same way security measures are listed according to the regulation and the company architecture, the legal requirements can also be laid out in a similar fashion. For example, the record of processing activities can be derived from the company part of the ontology. We can shine more light on the sensitive parts of the processed data set and give the user a more accurate picture of what security measures are needed. Also, we could identify cases where security measures are necessary because of third-country transfers and

similar situations while they would not have been required otherwise.

Finally, the extended ontology contains the partial notions of GDPR, ASVS framework, and the ISO 27002 standard. A fully working application would require complete body of knowledge regarding these standards and regulations. Furthermore, the knowledge base could be extended to use other regulations and standards. This would increase the chance of introducing new features to the base ontology because new knowledge would be gathered.

References

[1] Intersoft Consulting. "General Data Protection Regulation (GDPR) – Official Legal Text". (May 25, 2018),
[Online]. Available: https://gdpr-info.eu/ (visited on 02/05/2022).

[2] Intersoft Consulting. "What is GDPR, the EU's new data protection law?" (Nov. 7, 2018),
[Online]. Available: https://gdpr.eu/what-is-gdpr/ (visited on 02/05/2022).

[3] S. Dodd. "Bill Text - SB-1121 California Consumer Privacy Act of 2018." (Aug. 27, 2018), [Online]. Available: https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB1121 (visited on 02/05/2022).

[4] China Briefing. "The PRC Personal Information Protection Law (Final): A Full Translation". (Aug. 24, 2021),
[Online]. Available: https://www.china-briefing.com/news/the-prc-personal-information-protection-law-final-a-full-translation/ (visited on 02/05/2022).

[5] Thales Group. "Beyond GDPR: Data protection around the world". (May 21, 2021), [Online]. Available:
https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/magazine/beyond-gdpr-data-protection-around-world (visited on 02/05/2022).

[6] D. Tudor. “What Is an Attack Surface in Cybersecurity?” (Mar. 31, 2021), [Online]. Available: https://heimdalsecurity.com/blog/what-is-an-attack-surface-in-cybersecurity/ (visited on 02/06/2022).

[7] CyberArk. “Least Privilege”. (Jan. 29, 2020), [Online]. Available: https://www.cyberark.com/what-is/least-privilege/ (visited on 02/06/2022).

[8] A. Volter. “Security: The Need-to-know principle”. (Feb. 3, 2021), [Online]. Available: https://techcommunity.microsoft.com/t5/azure-sql-blog/security-the-need-to-know-principle/ba-p/2112393 (visited on 02/06/2022).

[9] Proofpoint, “2021 Human Factor Report”, Aug. 4, 2021. [Online]. Available: https://www.proofpoint.com/us/blog/email-and-cloud-threats/new-human-factor-report-uncovers-cybersecurity-year-changed-world (visited on 02/06/2022).

[10] Fortune Business Insights,
“Cyber Security Market Size, Share, Growth | Trends Analysis 2028”,
Mar. 2022.
[Online]. Available: https://www.fortunebusinessinsights.com/industry-reports/cyber-security-market-101165 (visited on 02/06/2022).

[11] N. F. Noy and D. L. McGuinness. “Ontology Development 101: A Guide to Creating Your First Ontology”. (Mar. 2001), [Online]. Available: https://protege.stanford.edu/publications/ontology_development/ontology101.pdf (visited on 02/06/2022).

[12] ISO/IEC. “ISO/IEC 27001:2013(en), Information technology — Security techniques — Information security management systems — Requirements”. (2013), [Online]. Available:
https://www.iso.org/obp/ui/#iso:std:iso-iec:27001:ed-2:v1:en (visited on 02/06/2022).

[13] OWASP, *OWASP Application Security Verification Standard 4.0.3*, Oct. 2021. [Online]. Available: https://github.com/OWASP/ASVS/tree/v4.0.3/4.0 (visited on 02/06/2022).

[14] D. Simmons. "17 Countries with GDPR-like Data Privacy Laws". (Jan. 13, 2022), [Online]. Available: https://insights.comforte.com/countries-with-gdpr-like-data-privacy-laws (visited on 02/06/2022).

[15] Defense Information Systems Agency, *Security Technical Implementation Guides (STIGs) – DoD Cyber Exchange*. [Online]. Available: https://public.cyber.mil/stigs/ (visited on 02/08/2022).

[16] MITRE, *MITRE ATTACK*. [Online]. Available: https://attack.mitre.org/ (visited on 02/21/2022).

[17] NDNB, *What is SOC 2? Introduction and Overview*. [Online]. Available: https://socreports.com/audit-overview/what-is-soc-2 (visited on 02/20/2022).

[18] HITRUST Alliance. "HITRUST CSF | Information Risk Management". (Dec. 30, 2021), [Online]. Available: https://hitrustalliance.net/product-tool/hitrust-csf/ (visited on 02/20/2022).

[19] N. Keller. "NIST Cybersecurity Framework". (Nov. 12, 2013), [Online]. Available: https://www.nist.gov/cyberframework (visited on 02/20/2022).

[20] GDPR EU. "GDPR Compliance Guide – User-Friendly Explanation". (Jul. 29, 2020), [Online]. Available: https://www.gdpreu.org/compliance/ (visited on 02/08/2022).

[21] CDC. “Health Insurance Portability and Accountability Act of 1996 (HIPAA)”. (Feb. 21, 2019), [Online]. Available: https://www.cdc.gov/phlp/publications/topic/hipaa.html (visited on 02/21/2022).

[22] L. Zagzebski, “What is Knowledge?”, in *The Blackwell Guide to Epistemology*, J. Greco and E. Sosa, Eds., Oxford, UK: John Wiley & Sons, Ltd, 2017, ch. 3, pp. 92–116.

[23] M. Davies, “Knowledge – Explicit, implicit and tacit: Philosophical aspects”, in *International Encyclopedia of the Social & Behavioral Sciences (Second Edition)*, N. J. Smelser and P. B. Baltes, Eds., Oxford, UK: Elsevier, 2001, pp. 8126–8132.

[24] B. Smith, “Ontology”, in *The furniture of the world*, G. Hurtado and O. Nudler, Eds., Leiden, Netherlands: Brill, Jan. 1, 2012, ch. 5, pp. 47–68.

[25] R. Studer, V. Benjamins, and D. Fensel, “Knowledge engineering: Principles and methods”, *Data & Knowledge Engineering*, vol. 25, no. 1, pp. 161–197, Mar. 1998.

[26] T. Chungoora, *Practical Knowledge Modelling: Ontology Development 101*. [Online]. Available: https://www.udemy.com/course/practical-knowledge-modelling/ (visited on 02/13/2022).

[27] S. Kanza and J. Graham Frey, “Semantic Technologies in Drug Discovery”, in *Systems Medicine*, O. Wolkenhauer, Ed., Oxford, UK: Academic Press, Aug. 2021, ch. 13, pp. 129–144.

[28] M. Horridge. “A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3”. (Mar. 2011),

[Online]. Available: http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf (visited on 02/21/2022).

[29] A. Herzog, N. Shahmehri, and C. Duma, "An Ontology of Information Security", *International Journal of Information Security and Privacy*, vol. 1, no. 4, pp. 1–23, Oct. 2007.

[30] S. Fenz and A. Ekelhart, "Formalizing information security knowledge", in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security - ASIACCS '09*, Sydney, Australia, 2009, pp. 183–194.

[31] S. Fenz, A. Ekelhart, and T. Neubauer, "Information Security Risk Management: In Which Security Solutions Is It Worth Investing?", *Communications of the Association for Information Systems*, vol. 28, pp. 329–256, 2011.

[32] S. Fenz, J. Heurix, T. Neubauer, and F. Pechstein, "Current challenges in information security risk management", *Information Management & Computer Security*, vol. 22, no. 5, pp. 410–430, Jan. 1, 2014.

[33] S. Fenz, S. Plieschnegger, and H. Hobel, "Mapping information security standard ISO 27002 to an ontological structure", *Information & Computer Security*, vol. 24, no. 5, pp. 452–473, Jan. 1, 2016.

[34] S. Fenz and T. Neubauer, "Ontology-based information security compliance determination and control selection on the example of ISO 27002", *Information & Computer Security*, vol. 26, no. 5, pp. 551–567, Jan. 1, 2018.

[35] K. P. Joshi, Y. Yesha, and T. Finin, "An Ontology for a HIPAA compliant cloud service", in *4th International IBM Cloud Academy Conference ICACON 2016*, Edmonton, Canada, Jun. 3, 2016, p. 2.

[36] C. Bartolini, R. Muthuri, and C. Santos,
"Using Ontologies to Model Data Protection Requirements in Workflows",
in *New Frontiers in Artificial Intelligence*,
Cham, Switzerland: Springer International Publishing, 2017, pp. 233–248.

[37] H. J. Pandit, C. Debruyne Lewis, D. O'Sullivan, and D. Lewis. "GConsent - A Consent Ontology based on the GDPR". (Nov. 25, 2018), [Online].
Available: https://openscience.adaptcentre.ie/ontologies/gconsent/main.html (visited on 01/27/2022).

[38] Stanford University, *Protégé 5 Documentation*. [Online]. Available:
http://protegeproject.github.io/protege/ (visited on 02/13/2022).

[39] University of Oxford, *HermiT Reasoner: Home*. [Online]. Available:
http://www.hermit-reasoner.com/ (visited on 02/13/2022).

[40] W3C. "SPARQL 1.1 Query Language". (Mar. 21, 2013), [Online]. Available:
https://www.w3.org/TR/sparql11-query/ (visited on 02/13/2022).

[41] yWorks, *yFiles Diagram Tool*.
[Online]. Available: https://www.yworks.com/yfiles (visited on 03/26/2022).

[42] Sotnikov, Ilia. "Data Classification: What It Is and How to Implement It". (Sep. 2, 2022), [Online]. Available:
https://blog.netwrix.com/2020/09/02/data-classification/ (visited on 03/29/2022).

[43] A. G. Salguero, C. Delgado, and F. Araque, "Easing the Definition of N–Ary Relations for Supporting Spatio–Temporal Models in OWL",
in *Computer Aided Systems Theory - EUROCAST 2009*,
Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 271–278.

[44] J. Potoniec, D. Wiśniewski, A. Ławrynowicz, and C. Keet, "Dataset of ontology competency questions to SPARQL-OWL queries translations", *Data in Brief*, vol. 29, article ID 105098, p. 13, Jan. 7, 2020.

[45] E. S. Bolotnikova, T. A. Gavrilova, and V. A. Gorovoy, "To a method of evaluating ontologies", *Journal of Computer and Systems Sciences International*, vol. 50, no. 3, pp. 448–461, Jun. 2011.

Appendix A Table of the terms recovered in the initial structuring phase

Below is the table of the terms recovered in the initial structuring phase.

Table A.1: Table of terms discovered in the initial structuring phase

Term	Prospective Entity	Action
SecurityMeasure	Class	Accept
Asset	Class	Reject
Data	Class	Reject
Profits	Class	Reject
Technology	Class	Reject
People	Class	Reject
SecurityImplementationPlace	Class	Reject
OrganisationalSecurityMeasure	Class	Accept
TechnicalSecurityMeasure	Class	Accept
SecurityMeasureSource	Class	Accept

Continued on next page

Table A.1 – continued from previous page

Term	Prospective Entity	Action
hasSecurityMeasureSource	Relation	Accept
Standard	Class	Reject
SecurityGuide	Class	Reject
RegulationDocument	Class	Reject
SecurityMeasureSource	Class	Accept
hasCategory	Relation	Accept
requiresImplementationOf	Relation	Accept
suggestsImplementationOf	Relation	Accept
ConditionalSecurityMeasure	Class	Accept
Condition	Class	Accept
hasOperand	Relation	Accept
underCondition	Relation	Accept
Architecture	Class	Accept
Company	Class	Accept
hasArchitecture	Relation	Accept
ArchitectureComponent	Class	Accept
implementedOnArchitectureComponent	Relation	Reject
hasArchitectureComponent	Relation	Accept
DataSet	Class	Accept
processesDataSet	Relation	Accept
PrivacyMechanism	Class	Accept
hasPrivacyMechanism	Relation	Accept
Node	Class	Accept
Channel	Class	Accept

Continued on next page

Table A.1 – continued from previous page

Term	Prospective Entity	Action
hasChannel	Relation	Accept
hasNoe	Relation	Accept
Regulation	Class	Accept
ApplicableRegulationText	Class	Accept
hasApplicableRegulationText	Relation	Accept
mentionedByApplicableRegulationText	Relation	Accept
DataType	Class	Accept
hasDataType	Relation	Accept
DataSetCondition	Class	Accept
underDataSetCondition	Relation	Accept
appliesToDataType	Relation	Accept
madeIrrelevantByPrivacyMechanism	Relation	Accept