D5.1 The 7SHIELD ontology and data representation model

Work Package:	WP5
Lead partner:	CERTH
Author(s):	Kopalidis Nikolaos (CERTH), Antzoulatos Gerasimos (CERTH), Gialampoukidis Ilias (CERTH), Vrochidis Stefanos (CERTH)
Due date:	31/05/2021
Version number:	1.0 **Status:** Final
Dissemination level:	Public

Project Number:	883284 **Project Acronym:** 7SHIELD
Project Title:	Safety and Security Standards of Space Systems, ground Segments and Satellite data assets, via prevention, detection, response and mitigation of physical and cyber threats
Start date:	September 1st, 2020
Duration:	24 months
Call identifier:	H2020-SU-INFRA-2019
Topic:	SU-INFRA01-2018-2019-2020 Prevention, detection, response and mitigation of combined physical and cyber threats to critical infrastructure in Europe
Instrument:	IA

Horizon 2020
European Union Funding
for Research & Innovation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 883284

Revision History

Revision	Date	Who	Description
0.1	09/04/2021	CERTH	First release of the template. ToC and assignments finalisation
0.2	12/04/2021	CERTH	First release of the v0.1
0.3	30/04/2021	CERTH	First round of contributions. Update current version and release of the v0.2
0.4	07/05/2021	CERTH	Second round of contributions. Update current version and release of the v0.3
0.5	10/05/2021	CERTH	Version ready for internal review
0.6	25/05/2021	CERTH	Addressed the internal review comments from DFSL and CS
0.7	27/05/2021	CERTH	Added complete list of authors. Updated table of contents, list of figures, list of tables. Minor formatting changes.
1.0	31/05/2021	CERTH, ENG	Release final version

Quality Control

Role	Date	Who	Approved/Comment
Internal review	23 / 05/ 2021	DFSL	Document accepted, only minor changes suggested
Internal review	25 / 05/ 2021	CS	Document accepted, no changes required

Disclaimer

This document has been produced in the context of the 7SHIELD Project. The 7SHIELD project is part of the European Community's Horizon 2020 Program for research and development and is as such funded by the European Commission. All information in this document is provided 'as is' and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability with respect to this document, which is merely representing the authors' view.

Executive Summary

The present deliverable describes primarily the process carried out in reach of the *T5.1 – Semantic representation and linking for decision-making*, relevant to the development of the 7SHIELD ontological framework, representation and mapping multi-modal content on semantic entities. Additionally, it contains the first methodological approach on the reasoning framework.

Based on the requirements structured by *WP2 – User Requirements and Use Cases Design* and the dependencies incurring from the interaction with the other WPs, the purpose, scope, intended users and uses, and the requirements of the 7SHIELD ontology were identified. These specifications, along with the modelling understanding from relevant study fields, played an important guidance role for building the first version of the 7SHIELD ontology that currently comprises modules for capturing the analysis resulting from detectors and correlators regarding physical and cyber alerts, attacks and threats. Furthermore, it describes the population process of these incoming data to the repository of the ontology and presents some validation examples.

The work presented within this document is a preliminary version of the 7SHIELD ontology. A more enriched version, with a solid reasoning framework and holistic reports for the users will be achieved after the implementation of input sources that are under construction at this point (e.g., mitigation plans).

Table of Contents

List of Figures

List of Tables

Definitions and acronyms

ABox	Assertional Axioms
BFO	Basic Formal Ontology
CA	Consortium Agreement
CCO	Common Core Ontologies
CI	Critical Infrastructure
CIP	Critical Infrastructure Protection
C/P	Cyber/Physical
CQ	Competency Question
DL	Description Logic
DoA	Description of Action
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
EC	European Commission
ENVO	ENVironment Ontology
EU	European Union
FR	First Responder
GA	Grant Agreement
GIS	Geographical Information System
HAR	Human Activity Recognition
KB	Knowledge Base
MEMOn	Modular Environmental Monitoring Ontology
MMF	Missions & Means Framework
MOAC	Management Of A Crisis
OOPS	OntOlogy Pitfall Scanner
ORSD	Ontology Requirements Specification Documents
OWL	Web Ontology Language
PC	Project Coordinator
RDF	Resource Description Framework
SOSA	Sensor Observation Sample Actuator
SC	Scientific Coordinator
SGS	Satellite Ground Station
SHACL	Shapes Constraint Language
SPIN	SPARQL Inferencing Notation
SSN	Semantic Sensor Network
SUMO	Suggested Upper Merged Ontology
TBox	Terminological Axioms
TM	Technical Manager
WP	Work Package
W3C	World Wide Web Consortium

1 Introduction

This deliverable *D5.1 "The 7SHIELD ontology and data representation model"* focuses on describing a first view of the 7SHIELD ontology. The latter, which also be called as "the 7SHIELD Knowledge Base (KB)", is a knowledge representation model for semantically representing concepts relevant to the project.

The goal of the KB framework within *WP5 – Post-Crisis management for response and mitigation of physical and cyber threats* is to research and develop technologies for semantic content and sensor input modelling, integration, reasoning and question answering. To this end, the information made available by *WP4 – Crisis management for detecting physical and cyber threats*, regarding the delivering of the detection mechanism, and from in later stage from *WP3 – Pre-Crisis management for prevention of physical and cyber threats* with pre-crisis and prevention technologies. The models that will be created will constitute for the reasoning mechanisms taking into account the ontology vocabulary and infrastructure for capturing and storing information relevant to the 7SHIELD application domain, such as: (a) Observation and events (e.g. data collection from face recognition/detection, multimodal automated surveillance, drone detection), (b) C/P security (e.g. cyber detection, correlation services output), (c) Mitigation and response plans (e.g. First responder teams, UAV neutralization).

The general architecture of the 7SHIELD is depicted in Figure 1–1. The semantic representation repository is a central component in the system's architecture and hosts the 7SHIELD KB, with the other components of the system interacting with it through the message broker.

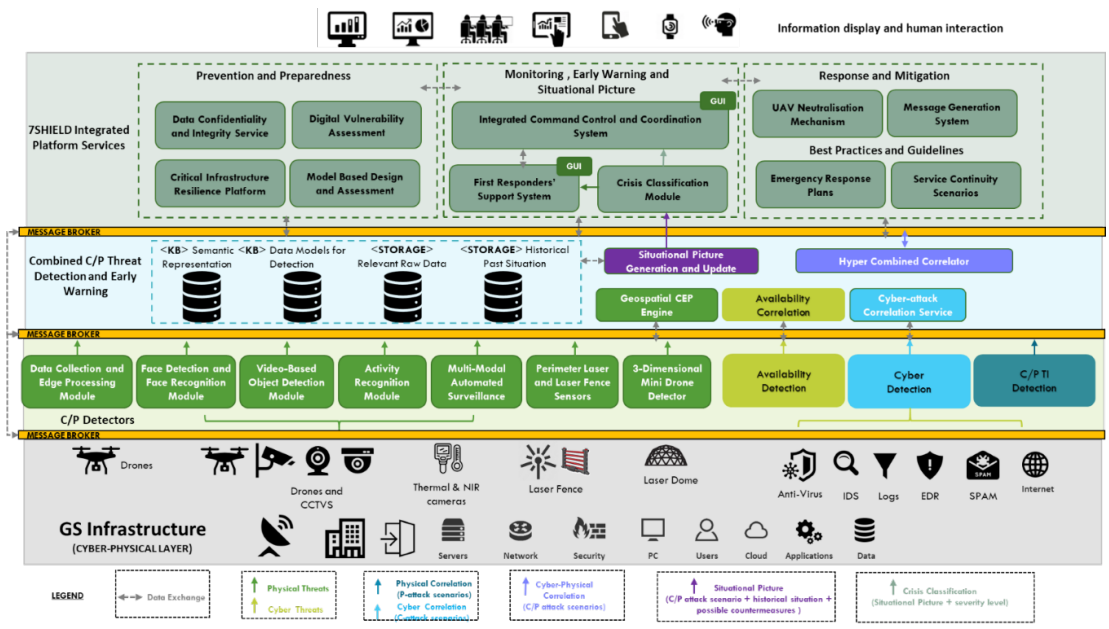

Figure 1–1 - General 7SHIELD architecture

The Figure 1–2 presents the high-level architecture of *T5.1 – Semantic representation and linking for decision-making*. The incoming inputs from the other components are mapped through the population service to the KB which provides a native Resource Description Framework (RDF) storage and querying services. The last entity is referring to the reasoning and context enrichment of the KB.

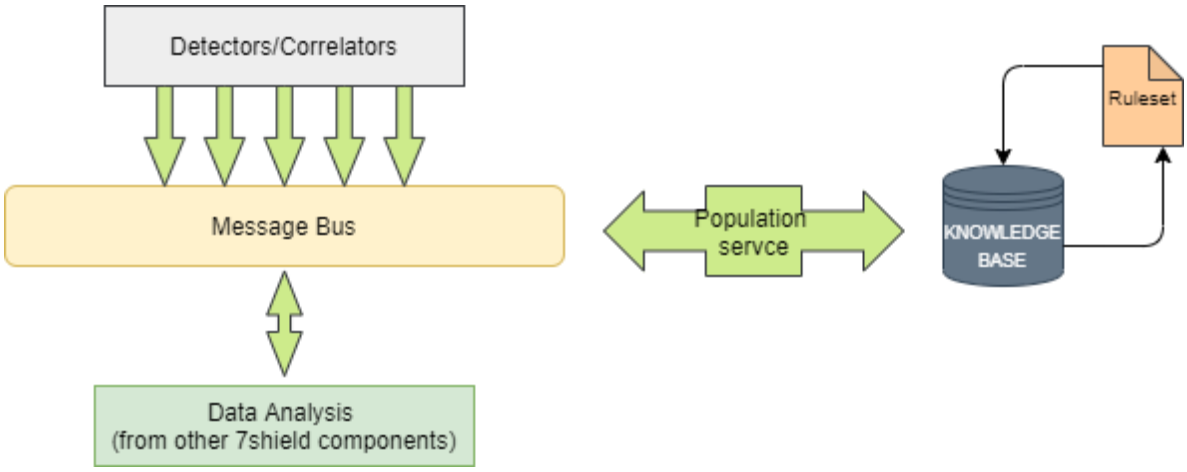

Figure 1–2 - Architecture of T5.1

The present deliverable reports on the work process carried out within Task 5.1 and focusing on the construction of the 7SHIELD ontology. Section 2 reviews the main standards with respect to knowledge representation languages as well as already existing ontologies addressing project-relevant fields. Section 3 presents the requirements the ontology has to meet; as detailed, their specification is largely driven by the requirements set forth by WP2's deliverable D2.2 "Consolidation of Stakeholder Requirements", due month M6 (Feb. 2021), while additional considerations issue from the fore-described possible dependencies with WP3 and WP4. Section 4 reports on the ontology

implementation and presents the current status of the 7SHIELD ontology. Section 5 presents an ontology validation example for an initial approach in the system's functionality. Section 6 contains the semantic reasoning requirements and methodology. Section 7 concludes the document, presenting the conclusions that were drawn and discussing future work and further enrichment of the module.

2 Knowledge Representation Overview

The present section provides an overview on the knowledge representation languages, already existing similar domain ontologies addressing project-relevant fields. More specifically, we present the foundational aspects of Description Logic (DL) languages [1] on which the official W3C recommendation for creating and sharing ontologies in the Web (OWL 2) is grounded, the different OWL 2 species, as well as relevant rule-based languages. Furthermore, a brief review on the representative ontologies that have been proposed in the literature for modelling core aspects relevant to the 7SHIELD application domain that fall into WP5's modelling requirements is presented.

2.1 Ontologies in the Semantic Web

Ontologies have been widely used as an effective way for modelling domain information because they can represent and organise information, context and relationships more accurately. Furthermore, they can be easily expanded by merging and combining parts of existing, relative or not, ontologies into new ones. Ontologies are structures that are mainly used to capture knowledge about some domain of interest. Formally speaking, ontologies are explicit formal specifications of shared conceptualizations [2]. They represent abstract views of the world including the objects, concepts, and other entities that are assumed to exist in some area of interest, their properties and the relationships that hold among them. Their expressivity and level of formalisation depend on the knowledge representation language used.

The Semantic Web-W3C, which is an extension of the current Web aims to establish a common framework for sharing and reusing data across heterogeneous sources, ontologies play a fundamental part. The Semantic Web vision is to make the semantics of web resources explicit by attaching to them metadata that describe meaning in a formal, machine-understandable way. Web Ontology Language (OWL) [3] has emerged as the official W3C recommendation for creating and sharing ontologies on the Web as the result of the previous effort. In the rest of this section, we present the basics of Description Logic (DL) languages, on which OWL semantics are grounded, the different OWL species.

2.1.1 Description Logic

Description Logics is a family of knowledge representation languages that may be used for a representation of knowledge of any application domain. This form of representation is in a structured and formally understandable way. The name Description Logics derives from two features — the first one is the ability to describe a given domain with the help of conceptual descriptions; the second is to provide logic-based semantics in contrast, for example, semantic networks or frames.

It is common for the DLs to include a terminological and an assertional formalism. A set of terminological axioms (TBox) is used to describe names (or labels) for complex descriptions. For example, TBox may contain a description of a concept Father:

```
Human ∩ Parent ∪ Father.
```

On the other hand, a set of assertional axioms (ABox) is used for description of properties of individuals. For example, the expression that describes the relationship between George and his son Alex:

```
hasChild(George, Alex)
```

DLs offer a reliable tool to deduce implicit knowledge from the explicitly defined knowledge with the help of TBox and ABox. The DLs provide well-defined semantics and powerful reasoning tools. For many years, there was a mismatch between the expressivity of DLs and the efficiency of reasoning. In other words, if a user wants to use a DLs, then he needs to establish a trade-off between the expressivity of DLs and the complexity of their inference capability. It means it is needed to restrict DL appropriately.

The cornerstone for OWL design was the expressive DL SHIQ [4]. In OWL language, the developers tried to find a balance between expressiveness and the complexity of reasoning.

2.1.2 Web Ontology Language

The OWL belongs to the Semantic web, which has been created to represent plentiful and complex knowledge about things, groups of things and relations between things. Owl can be described as computational logic-based language. For this purpose, OWL can be easier for machines to automatically process and integrate information available on the Web.

OWL uses RDF's XML syntax (RDF/XML). OWL has adopted several features of RDF/RDFS meaning of classes and properties and those language primitives are beneficial to overall expressiveness. On the other hand, RDF and RDFS have very voluminous modelling concepts such as `rdf:Property` and `rdfs:Class`. Thus, RDF and RDFS may be restricted when a trade-off between expressive power and efficient reasoning has to be established. There are three main kinds of OWL because of the trade-off mentioned above.

Different sub-languages are described in the following list:

- *OWL Full*: this kind of OWL represents the entire OWL language. This kind also offers the possibility to combine OWL primitives and RDF and RDFS. Moreover, the meaning of predefined primitives may be changed. OWL Full provides full compatibility with RDF, i.e., every valid RDF document is also valid OWL Full document. On the other hand, there is a possibility for the ontologies developed in OWL Full to be undecidable.

- **OWL DL**: this kind of OWL, where DL stands for Description Logic, restricts the application of constructors from OWL and RDF. The restrictions include: (1) Vocabulary partitioning; (2) Resources are allowed to be only one of specific type, i.e., a class, a datatype property, an object property, an individual, etc. Strictly speaking, a property cannot be a datatype property and at the same time object property and vice versa. The efficient reasoning is secured because of: (a) explicit typing of resources; (b) no transitive cardinality restrictions; (c) restricted anonymous classes. Furthermore, compatibility with RDF is lost. On the other hand, every valid OWL DL document is a valid RDF document.

- **OWL Lite**: is the last version which represents a restriction of OWL DL. The restrictions are for example excluding enumerated classes, disjointedness of classes, and cardinality (except the values 0 or 1).

2.1.3 Ontology Engineering

Providing well-designed and substantial ontologies which stand the test of largescale applications is a current bottleneck in Semantic Technologies. According to primary intention, the Semantic Web should facilitate a search for suitable ontologies, integrate them with few simple changes and exploit them within a given application. A large number of ontologies are available, but well-designed ones are rarely to be found. Making a good use of upper ontologies for information integration is not limited only to ontology engineering but may be a mean for integration of data sources represented in various formats. The solutions which adopted the aforementioned methodological approach which utilizes an abstract foundational ontology to facilitate domain ontology integration, are plenty, such as ARECIBO, beAWARE, etc.

Upper ontologies can be seen as axiomatic theories about the high-level as well as domain-independent categories in the real world, e.g., physical object, cyber object, threat, observation, etc. DOLCE and SUMO [5] ontologies are considered to be the most capable to play the role of the upper ontology, however our approach is based on the SSN about which we will discuss later in the document. The major advantages of an upper ontology employment are as follows.

- **Conceptual accuracy**: Upper ontologies provide a reference centre for comparison among different ontological methodologies and a framework for integrating existing ontologies.

- **Design patterns**: Ontology design patterns are defined by the upper ontology for properly re-occurring modelling needs.

- **Modelling** background: Upper ontologies can be viewed as instruction guides for building new ontologies while having a methodological background, instead of modelling them from scratch.

2.1.4 Querying and Reasoning

As it was mentioned, DLs, consequently and OWL, trade some expressiveness for more adequate reasoning. The tree-model property is one such example. It conditions the tree-shape structure of models, ensuring decidability, but at the same time it severely restricts the way variables and quantifiers can be used, dictating that a quantified variable must occur in a property predicate along with the free variable. As a result, it is not possible to describe classes whose instances are related to an anonymous individual through different property paths. In order to leverage OWL's limited relational expressiveness and to overcome modelling shortcomings the research body came up with the integration of rules with OWL.

The first step toward this was SPARQL a language recommended by the W3C for extracting and updating information in RDF graphs. It is characterized by expressiveness with the ability to describe complex interactions and relationships between entities in a knowledge graph. The semantics and multiplicity of the SPARQL language have been reviewed in detail theoretically, showing that SPARQL algebra has the same expressive power as relational algebra [6]. Despite the fact that SPARQL is mainly used as query language for RDF, by using the CONSTRUCT graph pattern, it is able to define SPARQL rules that can create new RDF data, combining existing RDF graphs into larger ones. These rules are defined in the interpretation layer in terms of a CONSTRUCT and a WHERE clause: the former defines the graph patterns, i.e. the set of triple patterns that should be added to the underlying RDF graph upon the successful pattern matching of the graphs in the WHERE clause. The SPARQL Inferencing Notation (SPIN) [7] helps with the establishment of an easier expression and execution of SPARQL rules on top of RDF graphs. In SPIN, SPARQL queries can be stored as RDF triples together with any similar domain model, enabling the linkage of RDF resources with the associated SPARQL queries, as well as sharing and reusing them. SPIN supports the definition of SPARQL inference rules that can be used to derive new RDF statements from existing ones through rule application. A newer standard that has been developed as a tool to define structural constraints on RDF charts is Shapes Constraint Language (SHACL). SHACL consists of two parts: (1) kernel that elaborates RDF vocabulary for the definition of shapes and variables and (2) SHACL-SPARQL which is a mechanism for expanding the SPARQL.

2.2 Related Domain Ontologies

The scope of this subsection is to present the state-of-the-art ontologies that can be used for modelling aspects relevant to the 7SHIELD's domain of application. According to the 7SHIELD ontological requirements, which will be reviewed in the following section, we have categorized the relevant ontologies into four domains. First, the ones that can be used to model events and observations. Next there are the crisis management ontologies

(modelling risks and mitigation) followed by the C/P systems (cyber physical threats and vulnerabilities) and finally the ontologies for general purposes; temporal and geospatial.

It should be noted that the purpose of this section is not to provide a complete list of ontologies relevant to the V4Design domain, but to highlight on design concepts and entities that have been proposed or used in systems for annotation and conceptualization.

2.2.1 Observation and Events

The mapping of sensors and their observations, properties and features of interest has been in the centre of many approaches. Towards this, the most well-known are the Semantic Sensor Network (SSN) [8] and Sensor Observation Sample Actuator (SOSA) [9]. They have been applied in various use cases, applications and scenarios including satellite imagery, large-scale scientific monitoring, industrial and household infrastructures, social sensing, citizen science, observation-driven ontology engineering, and the Web of Things.

The first ontology that was studied is Modular Environmental Monitoring Ontology (MEMOn) [10]. MEMOn is based on other ontologies, namely the Basic Formal Ontology (BFO), the ENVironment Ontology (ENVO) [11], the Semantic Sensor Network Ontology (SSN) and the Common Core Ontologies (CCO). In the following Figure 2–1 it is shown that the ontology offers eight main modules covering more aspects than the aforementioned ontologies to represent different emergency incidents.

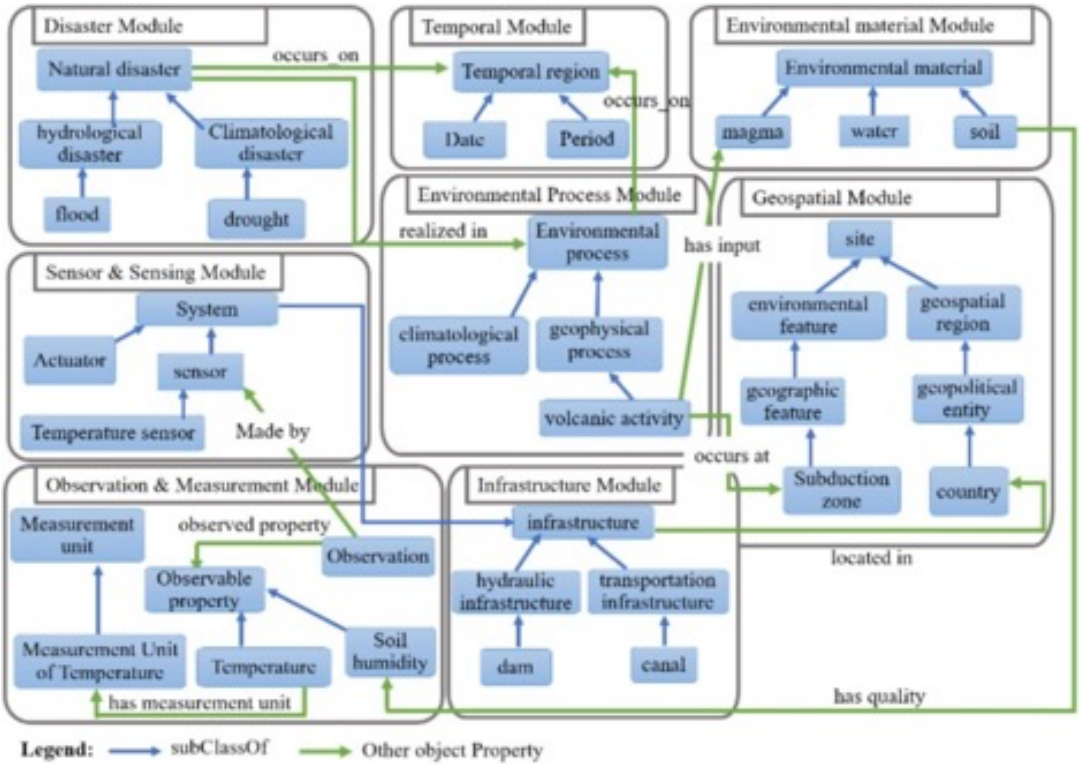

Figure 2–1 - MEMOn ontology overview (Source: [10])

Another notable ontological framework in this domain is Missions & Means Framework (MMF) [12] which is an ontology developed in the context of managing sensor assignment to mission.

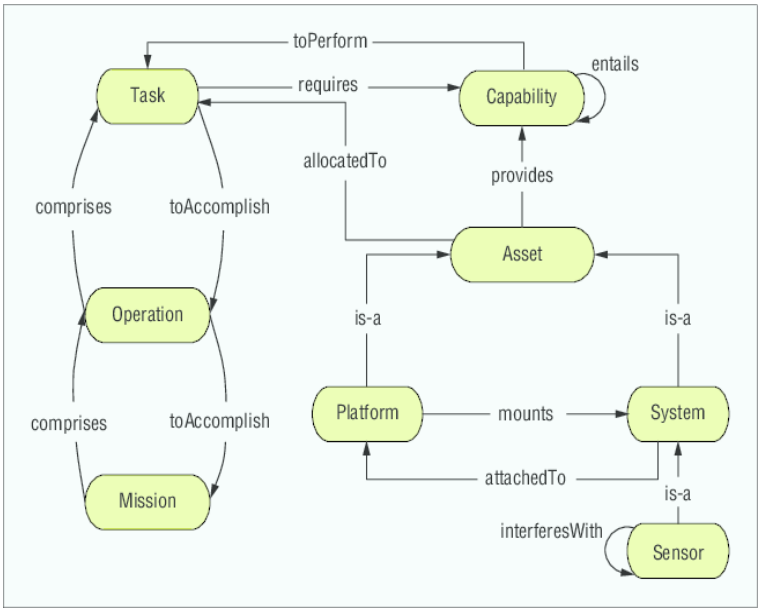

Figure 2–2 - MMF Ontology Overview (Source: [12])

The next ontology was Event Ontology [13], which describes the environment and the events, with their actions, that surround and change the effective state of the character. The following questions are some of the information that is described through this ontology:

1. what the action is, which contains a verb and the complements
2. the description of the place where the action occurs
3. the period of time
4. the persons and animals that execute and/or receive the action.

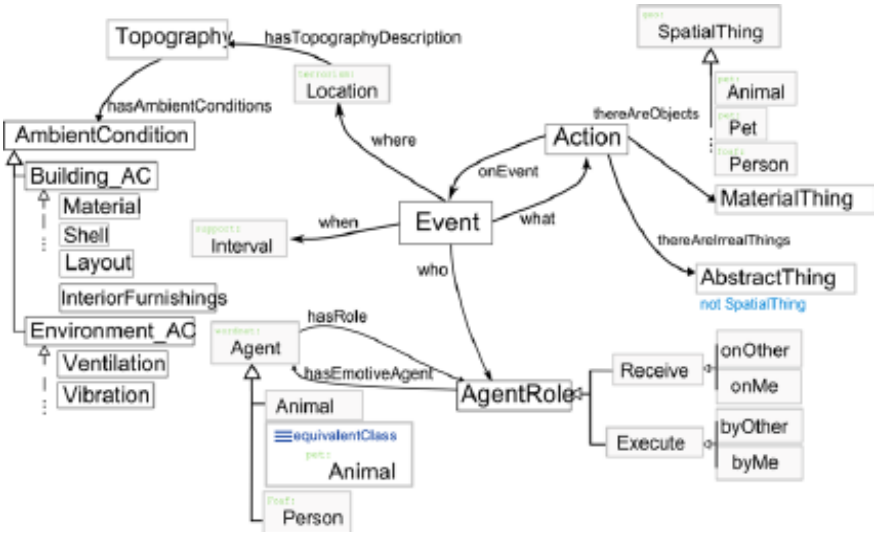

Figure 2–3 - Event Ontology Overview (Source: [13])

2.2.2 Crisis Management

The emergence of Semantic Web technologies [14] has led to the widespread adoption of ontology-based approaches in various domains, including crisis management. A recent thorough review of the state of the art in crisis management ontologies is given in [15] that contains a very detailed comparison between them. Two of the most important approaches with wide field of application in crisis management and response are MOAC [16] and SoKNOS [17].

The approach of building a Knowledge Base for information security that developed an automation of some security implementation and evaluation tasks that can reduce the costs and potentially increase their quality [18]. The ontology is divided into two parts: the concepts representing information security domain knowledge (which actually are core concepts of the domain) and the aspect representing solid information about the considered system, which are essential in assessment and observation of its security level. In the following Figure 2–4, the concepts of vulnerability, threat and assets are visible.

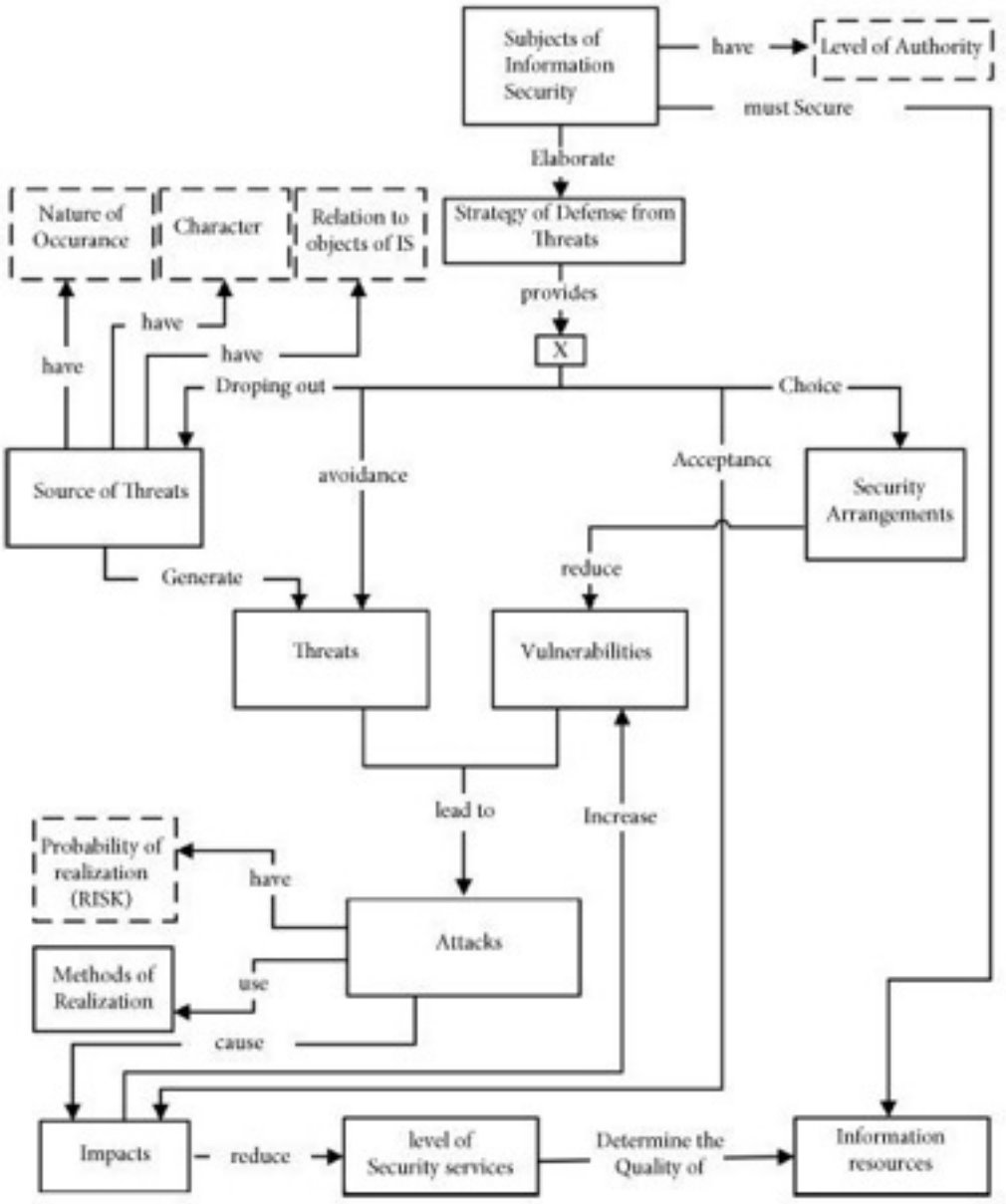

Figure 2–4 - Crisis Ontology Overview (Source: [18])

2.2.3 Cyber and Physical Security

Nowadays we are facing significant cyber-physical threats in integrated systems. These threats are directed to all the levels of Integrated systems. The cyber-physical systems could be distinguished in low-level integration systems and high-level integration systems. A sample pertaining to low-level integration systems could be a camera of a building, which does not have a very complicated system and doesn't possess a variety of sensors. On the contrary a high-level integration system consists of multi-sensors and its function is considered very significant. Such an example is a system for the safety of a nuclear powerplant, which could face physical or cyber threats. In the case that these threats succeed, the impact on people and furthermore the environment would be devastating. In

this regard ontologies could contribute to the field of security. And more specifically in the domain of psychical and cyber threats.

The following methodology, which proposed in [19], approaches a cyber ontology but it is quite easy to expand to cyber and physical domain. The Figure 2–5 depicts a high-level architecture of the developing ontology. Each rounded box represents a major category of concepts. These concepts are feasible to be rearranged along with a level of abstraction continuum from broad and general to domain specific. The bigger surrounding boxes represent separate ontologies that span multiple concept categories.

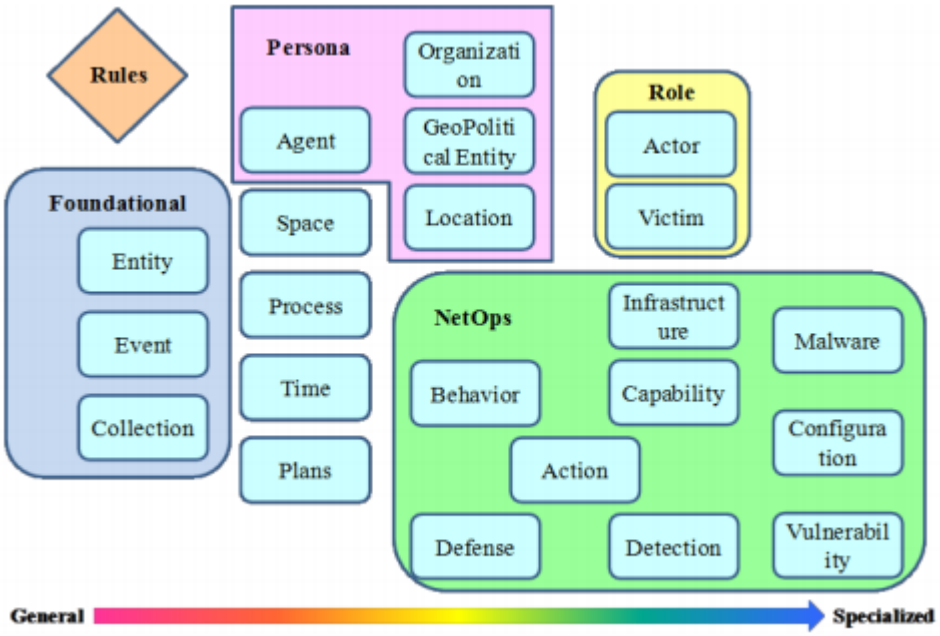

Figure 2–5- Cyber Ontology Overview (Source: [19])

2.2.4 Time and Geospatial Data

In semantic web there are two standard ontologies of temporal concepts, OWL-Time [20] and time-entry [21]. They both provide similar vocabularies for expressing facts about temporal intervals and instants, while time-entry also includes the concept of an event. In addition, the ontologies include classes and relations for expressing time intervals and instants in clock and calendar terms. Both of them include the concept of a time zone, and a separate global time zone recourse in owl is available.

The importance of the geospatial data (e.g., locations, distances, coordinates) and their semantic representation is well known by the research community, because they offer solid methods for retrieving information that are used in several Geographical Information System (GIS) applications. There is a large number of geographical ontologies that are used to express semantically geographical and spatial information. One of the most prominent of them is the GeoSPARQL. GeoSPARQL defines an RDF/OWL vocabulary for representing

the aforementioned information and also elaborates them with the use of a query language with powerful rules and functions, that allow precise semantic reasoning.

3 Modelling and Reasoning Requirements

In this section we describe here the approach followed to collect the modelling and reasoning requirements as well as a description of the results of this approach. Additionally, there is an effort on the association of the technical requirements concern the modelling and reasoning with user requirements.

3.1 Methodology Overview

The methodology followed in order to elucidate modelling and reasoning requirements for the 7SHIELD KB can be visualized with the use of structural blocks of developing actions. An overview of this approach is shown Figure 3–1.

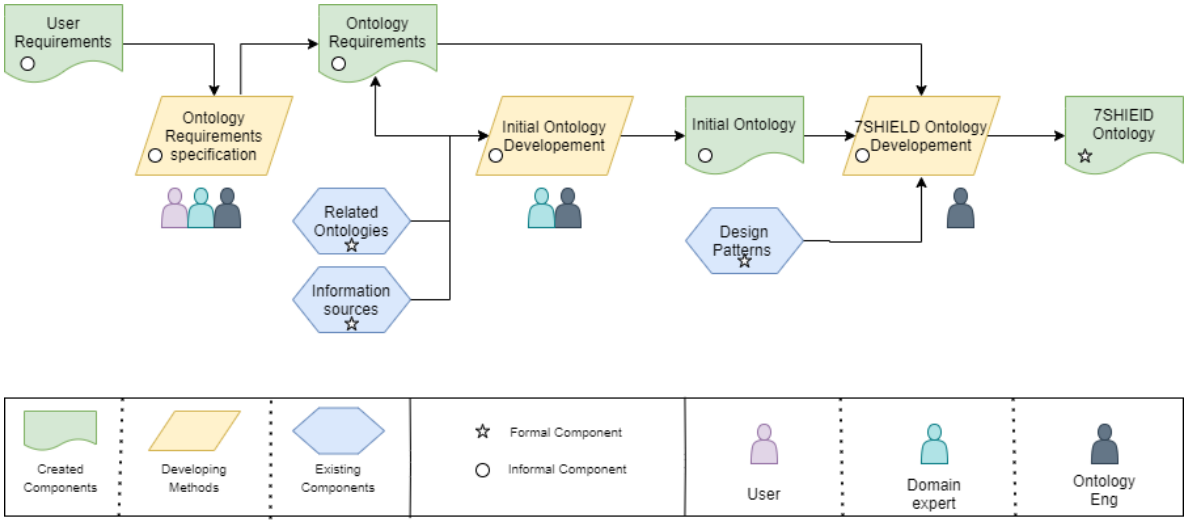

Figure 3–1- General Methodology that followed in T5.1

The process that was followed can be divided into three major developing stages with several possible inputs and outputs.

1. The first stage is focused on ontology requirements specification and the retrieval of ontology requirements specification documents (Ontology Requirements Specification Document, described in the following section). In this stage important is the role of end users that will provide insights regarding the user requirements. Additionally, domain experts will help understand the use cases and find the optimal matching with the ontology requirements. Finally, ontology engineers have a more consulting role in this stage regarding the process execution.
2. The second stage, after the acquisition of ontology requirements, involves the development of an initial ontology making a good use of related ontologies of the same domain, and information from several outputs of the 7SHIELD system, that have filtered with the results of the first stage. The role of the ontology engineers here is major, whilst he translates the domain experts' findings into a machine interpretable format.

3. The third stage contains the enrichment of the initial ontology with the use of more advanced design patterns and further specify the incoming information, with the use of the OWL to finalize the 7SHIELD ontology.

3.2 Related User Requirements

In this section we will revise the user requirements that were presented in the deliverable D2.2: "Consolidation of Stakeholder Requirements". These requirements are the fundamental principles in the development of the 7SHIELD's ontology modelling and reasoning framework. Table 3-1 presents the user requirements relevant to the knowledge base and data representation model, briefly describing their main functionalities and services as well as other possible key results that are involved.

User Req. ID	Type	PUC	Description	Relevance
FR_SCE_01	FR	1	7SHIELD must produce an automated offline report of a physical intrusion incident, after the mitigation of the physical attack	Provide the ability to retrieve the collected metadata which refer to an area of interest from the KB
FR_SCE_42	FR	4	Automated reports that can be used for providing more information to other parties (Security Officer, ESA SCB…)	Provide the ability to retrieve the collected metadata which refer to an area of interest from the KB
FR_SCE_69	FR	3	Record of available data for future reference (assessment of damage, data loss inflicted, mapping of the cyberattack, web analysis etc.) to provide authorities with all available info.	Provide the ability to retrieve the collected metadata which refer to an area of interest from the KB
FR_SCE_91	FR	3	7SHIELD must produce a report on the affected systems and servers after a cyber or physical attack.	Support searching functionality and interface over the KB to find the history of metadata

Table 3-1- User Requirements related to T5.1

3.3 Ontology Requirements Specification

As it is mentioned, the important role in the first stage of the methodology that followed, was the Ontology Requirements Specification Document [22]. This is a template-based report in which we determine which is the domain and scope of the ontology. Furthermore, this document helps us to specify why the ontology is being built, which are the intended

uses, who the end users are, what the ontology should fulfil and the verification, grouping and prioritization of requirements.

3.3.1 ORSD Template

The template of a ORSD contain the following fields in which can be find information regarding the purpose, scope, implementation language, intended end-users, intended uses, requirements and pre-glossary of terms of the ontology which is being built:

- Purpose: The main general goal of the ontology/ main function or role that the ontology should have.
- Scope: The coverage and the degree of details that the ontology should have.
- Implementation Language: The formal language that the ontology should have.
- Intended End-users: The intended end-users expected for the ontology.
- Intended uses: The intended uses expected for the ontology.
- Ontology requirements:
  - Non-functional requirements: The general requirements or aspects that the ontology should fulfil, including optionally priorities for each requirement
  - Functional Requirements: Groups of Competency Questions (CQ): The content specific requirements that the ontology should fulfil in the form of groups of competency questions and their answers, including optional priorities for each group and for each competency questions [23].
- Pre-glossary of Terms:
  - Terms from Competency Questions: The list of items included in the competency questions and their frequencies
  - Terms from Answers: The list of terms included in the answers and their frequencies
  - Objects: The list of objects included in the competency questions and their answers

3.3.2 7SHIELD ORSD

The 7SHIELD ORSD is based on the use cases and requirements laid out in deliverable D2.2 "Consolidation of Stakeholder Requirements". Additional feedback and clarifications have been elicited through iterative cycles of communication with WP3, WP4, and WP6 that extended equally and were qualified to provide supplementary analysed input that ultimately came to further refined and unambiguous requirements. Therefore, the previous process results in the ORSD that reflects the ontology requirements as pertinent to the current status of the 7SHIELD system. It is possible that some revisions and extensions will need to be carried out as the system functionalities evolve. The following Table 3-2 constitutes the 7SHIELD ORSD.

1	*Purpose*
	As the purpose of the 7SHIELD semantic representation framework we can define the structures and the vocabularies that are used, to capture the analysis results coming from other components. The system needs the ontology in order to secure interoperability and reusability between the individual modules and to support, in conjunction with inference rules, personalised interpretation services.
2	*Scope*
	The ontology has to focus just on the following aspects: • Representation of the analysed data from multimodal sensors. • Representation of correlated and aggregated incoming C/P detections. • Representation of C/P threats and the risks that relate to them. • Representation of an event and its mitigation. • Representation of historical data
3	*Implementation language*
	The ontology will be implemented in OWL 2, the officially recommended language by W3C for knowledge representation in the Semantic Web.
4	*Intended End-Users*
	• *PUC1: Physical Attack in Arctic Space Centre* The duty operator who wants to have an overall view about and intrusion of an unauthorised person or a hostile drone, the geospatial data regarding the event and historical data regarding the locations of ground station infrastructures. • *PUC2: Cyber-physical attack in Deimos Ground Segment* The IR operator after a series of problems like unauthorized/malicious access to ground facilities and/or cyber intrusions. The basic system logs and action will be mapped for future reinspection. • *PUC4: Threat detection and mitigation on the ICE Cubes Service* The security officer who wants to retrieve data regarding a cyber-attack (DDoS) that includes logs, mitigation actions in order to create a specific report. These data that have to be retrieved had earlier been stored to the KB by the operator.
5	*Ontology Requirements: Functional Requirements - CQs*
	1. Observations 1.1. What is the severity of the observation [X]? 1.2. What is the confidence of the observation [X]? 1.3. What is the analyser category that made the observation [X]? 1.4. What is the detection/creation time of the observation [X]? 1.5. Which analyser made the observation [X]?

1.6. Which is the GeoLocation of the analyser[X]?
1.7. Which is the UnLocation of the analyser[X]?
1.8. Which is the Location of the analyser[X]?
1.9. Which is the method used by the analyser[X]?
1.10. What is the data used by the analyser[X]?
1.11. In which infrastructure does the observation [X] take place?
1.12. Which is the most/least severe observation?
1.13. Which agents where detecting between time intervals [t_1]-[t_2]?
1.14. Which observations occurred after time [t_1]?
1.15. How many physical vectors were detected between time intervals [t_1]-[t_2]?
1.16. How many physical vectors were detected between time intervals [t_1]-[t_2]?
1.17. What is the Location of the target in the observation [X]?
1.18. What is the IP of the source in the observation [X]?

2. Threats
2.1. What is the category of the threat [X]?
2.2. What is the source/target IP in a cyber threat [X]?
2.3. What is the type of intruding object [X]?
2.4. What is the type of the recognised activity [X]?
2.5. Who is the recognised face[X]?
2.6. How many Incidents[X] were recorded?
2.7. What type of threats are detected between time intervals [t_1]-[t_2]?
2.8. Which is the manifestation of threat [X]?
2.9. Which infrastructure is targeted the most?
2.10. Which is the most/least common threat [X]?
2.11. Which observation led to the threat [X]?

**The following competency questions are not yet implemented but they will in the following version (the syntax is not mandatory to be the same)*

3. Risk Assessment & Mitigation Plan
3.1. What is the location of the FR [X]?
3.2. Who is the leader of the FR [X]?
3.3. What is the current mitigation plan of the FR [X]?
3.4. For which incident is the mitigation plan?
3.5. What is the location of the FlyingHunter?
3.6. What is the Impact on the Critical Infrastructure [X]?
3.7. What is the Likelihood on the Critical Infrastructure [X]?
3.8. What is the Vulnerability on the Critical Infrastructure [X]?
3.9. What is the Condition of the FR [X]?

Table 3-2 – 7SHIELD OSRD

4 7SHIELD Ontology

In this section, we present the content of the first version of the 7SHIELD ontology. The modelling of the classes, properties and individuals has been structured in accompany with the competency questions that we reviewed within the previous section. Furthermore, whenever it was possible on a conceptual level the related standards and ontologies drew guidelines that we followed. The formalization of the ontology is based in the SSN/SOSA ontological framework, and it was used as an upper ontology.

4.1 Reuse of existing sources

For dealing with semantic heterogeneity in complex systems like 7SHIELD, Semantic Web technologies were chosen for building a suitable solution. Semantic Web Technologies represent one of the promising ways to ensure interoperability as discussed in Chapter 2. One of its approaches in providing the suitable outcome is by making good use of similar domain ontologies.

Semantic Sensor Network (SSN) ontology is introduced, and this ontology may be utilized for a description of sensing devices as well as related processes. The SSN ontology is based on the ontology design pattern called Stimulus-Sensor-Observation pattern [24]. The SSO was designed as the cornerstone for heavy-weight ontologies for the Semantic Sensor Web applications. This pattern is also aligned to the Dolce Ultra-Lite ontology, a very common framework that is used as an upper ontology. The architecture of SSN ontology together with the dividing to modules is illustrated in Figure 4–1 .

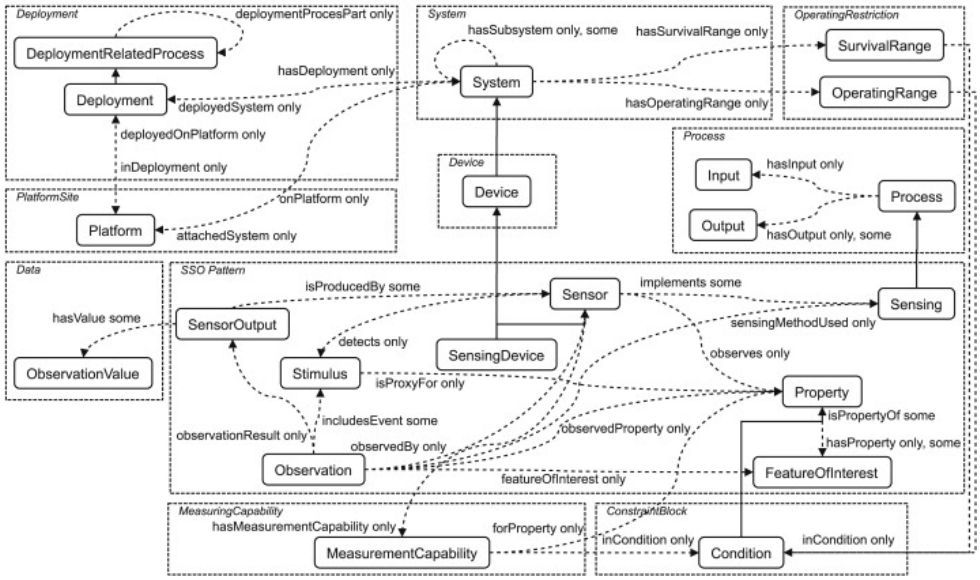

Figure 4–1 - SSN architecture Overview (Source:[24])

SSN ontology is composed of several modules that are fundamental in the sensor representation domain. The module Skeleton represents the core conceptualization as a lightweight ontology with a minimal commitment. This part includes the main concepts such as Sensor, SensorOutput, Observation, SensingDevice, and Sensing. Next, the module Process represents processes together with their inputs and outputs. Besides of the main modules, SSN is also composed of following modules — MeasuringCapability, ConstraintBlock, Device, OperatingRestriction, System, Deployment, PlatformSite, and Data which are not relevant at this point in the 7SHIELD conceptualization.

A major role in the conceptualization played a newer version of the SSN, the Sensor, Observation, Sample, and Actuator (SOSA) which is a lightweight version that incorporates Actuators, and it is not based on the DUL ontology. This allows the representation of the:

- **Sensor**: A sensor is any entity that implements a sensing method and thus observes some property of real-world entities (things, persons, events, etc). Sensors may be physical devices, computational methods, a laboratory setup with a person following a method, or any other thing that can follow a sensing method to observe a property.
- **Observations**: They can be considered as the connection among stimuli, sensors and their outputs. In SSN/SOSA, observations are rather contexts for the interpretation of the incoming stimuli than physical events, in contrast to O&M where observations are interpreted as events.
- **Feature of interest**: A feature is an abstraction of real-world phenomena that are the target of sensing, e.g. a person.
- **Procedure**: Procedure is a description of how a sensor works, e.g. a description of the scientific method behind the sensor. Sensors can be thought of as implementations of sensing methods to derive information about the same type of observed property.

4.2 Conceptualization

In this subsection, the detailed conceptualization of the ontology and its entities is presented. As it is already mentioned the concepts introduced by the SSN ontology are quite important for cyber-physical systems-sensors, observations, sensing devices, their relationships etc. Thus, additional concepts have to be designed in order to cover the multifaceted nature of the knowledge that was previously presented as CQs. The following graphs visualize with simplicity the new concepts that were integrated, starting from a higher level and gradually reaching the lower-level entities of the ontology. The methodology followed was based on the Modelling OWL Ontologies with Graffoo.

The

Figure 4–*2* displays an overview of the core ontology classes. In order to make it simpler, we have omitted data type and inverse properties, as well as extensive class hierarchies. The entities that are a different colour are the ones that have not yet implemented in this version of the ontology population however our concept behind the will be described. The full list of classes and properties is presented in Appendix A.

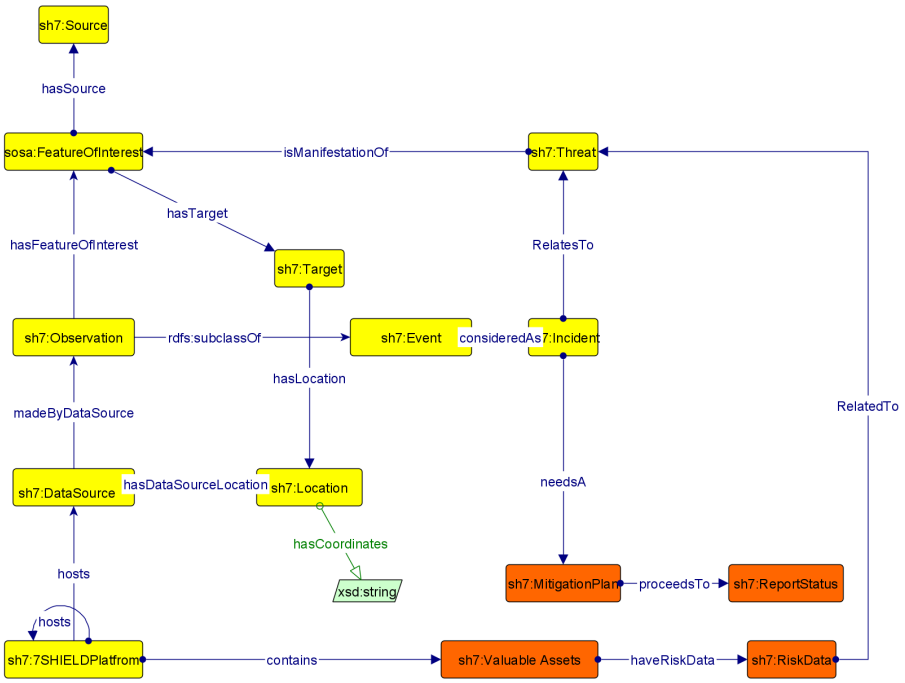

Figure 4–2 - High Level overview of 7SHIELD ontology

The first version of the 7SHIELD semantic model is analysed in detail. Also, the fundamental classes of 7SHIELD ontology are described.

Data Source: This class represents data that have been analysed and a result has been extracted.
Analyzer: This class represents a piece of equipment used to analyse data from different sources and to draw conclusions.
Event: This class represents one of the primaries of the overall data model of the information sharing environment. Event is an abstract entity which has a subclass, the Observation.
Sensor: This class represents the type of the sensor which detects the event. From an instance of the sensor, we receive the information of the IP of the sensor, name and location of the sensor.
Method: This class is used to contain all the methods of the analyser.
7SHIELDPlatform: This class hosts other entities, particularly Sensors, Detectors, Samplers.
Location: This class represents the place or position that something is in or where something happens. The class Threat is further divided into 3 subclasses (PhysicalLocation, GeoLocation, Unlocation). The subclass **Unlocation** is used to characterize geographic coding scheme which is the United Nations Code for Trade and Transport Locations. Furthermore, the subclass PhysicalLocation indicates the location of a physical object.
Threat: This class represents any hostile action on someone or something. The class Threat is further divided into 3 subclasses (AvailabilityOrigin,CyberOrigin, PhysicalOrigin) which define the type of threat. An instance threat leads to an event.

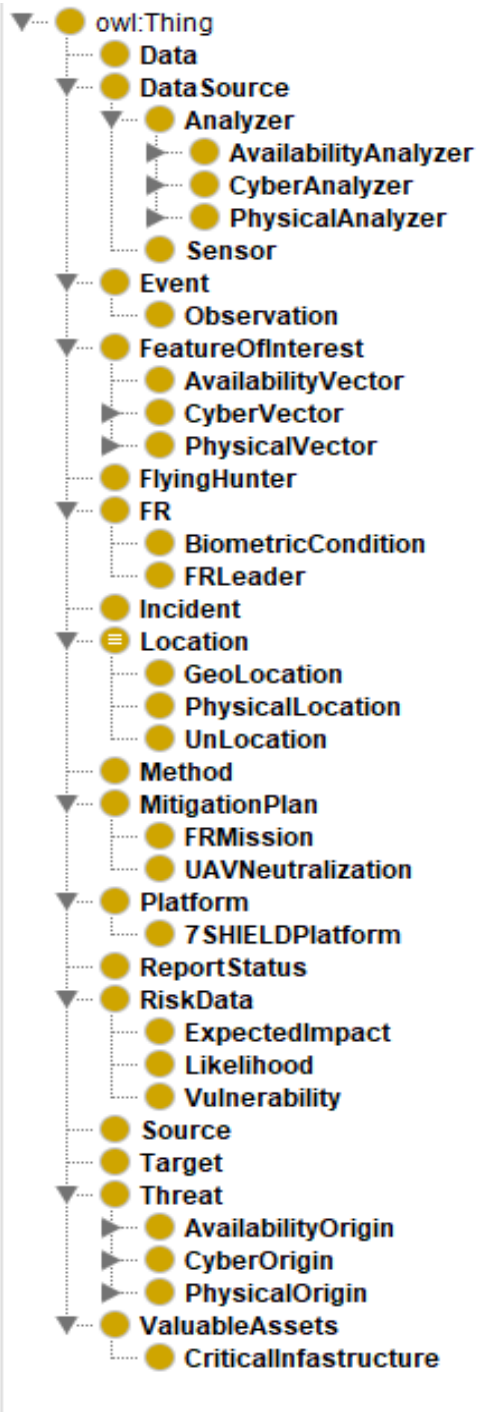

Figure 4–3 - List of classes as they are viewed in Protege

ValuableAssets: This class is used to characterize all the assets which have been assessed as valuable.
Target: This class represents an object of attention or attack.
Source: This class represents information about the source of the cyber threat.
ReportStatus: It has not been developed yet, however its purpose is to make a report when trigger from an event.

Data: This class contains information about data linked to a source, target or vector.
FR: This class has not been integrated yet.

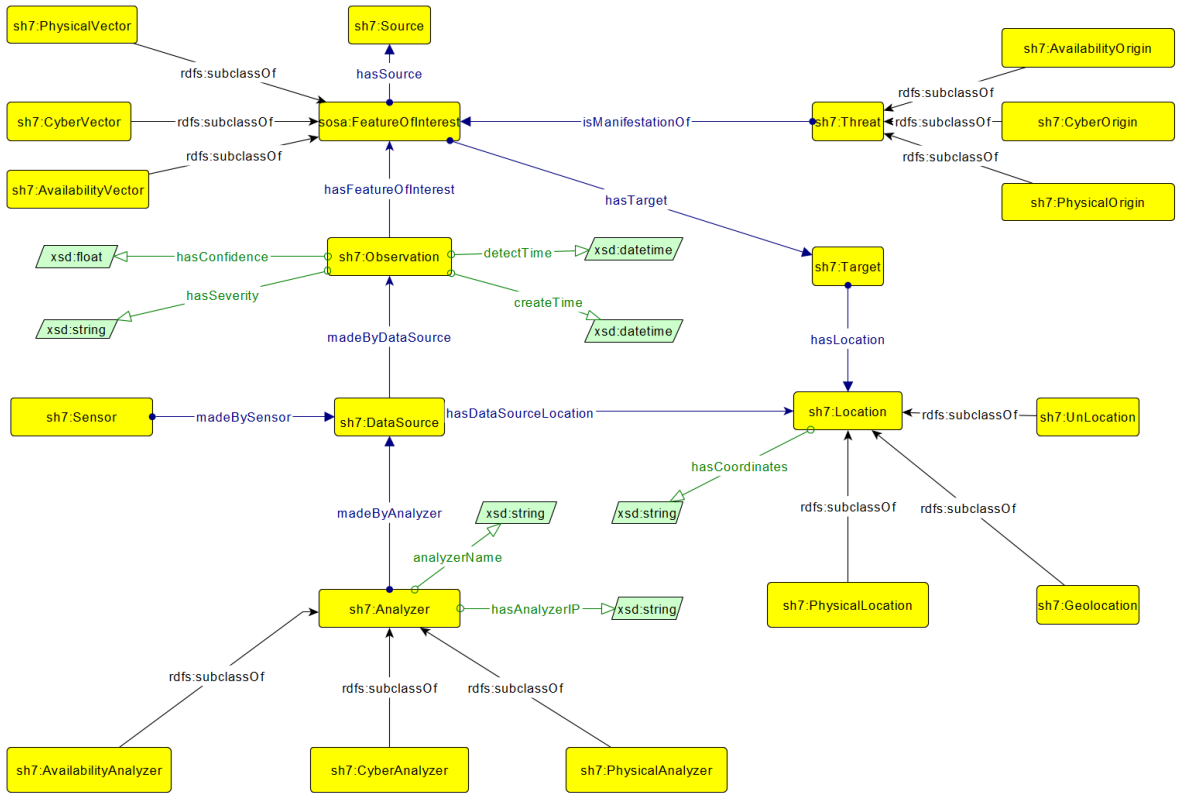

Figure 4–4 - Representation of analysed data in 7SHIELD otology

In the Figure 4–4, a more detailed view of the core classes is presented, with some additions of datatype properties. The confidence and the severity of the observation, the methods and the data of the analyser, the origin of the threats and the categories of the feature of interest are some of the datatypes represented.

Figure 4–5 demonstrates an example of a detected instance, where a man is detected participating in a threat that it is assumed to be a burglary. There is the specific location of the basic components (target, analyser, sensor). It is worthy to mention here that the 7SHIELD ontology contains a complete typology of data, methods, threats, sensors, analysers, etc.

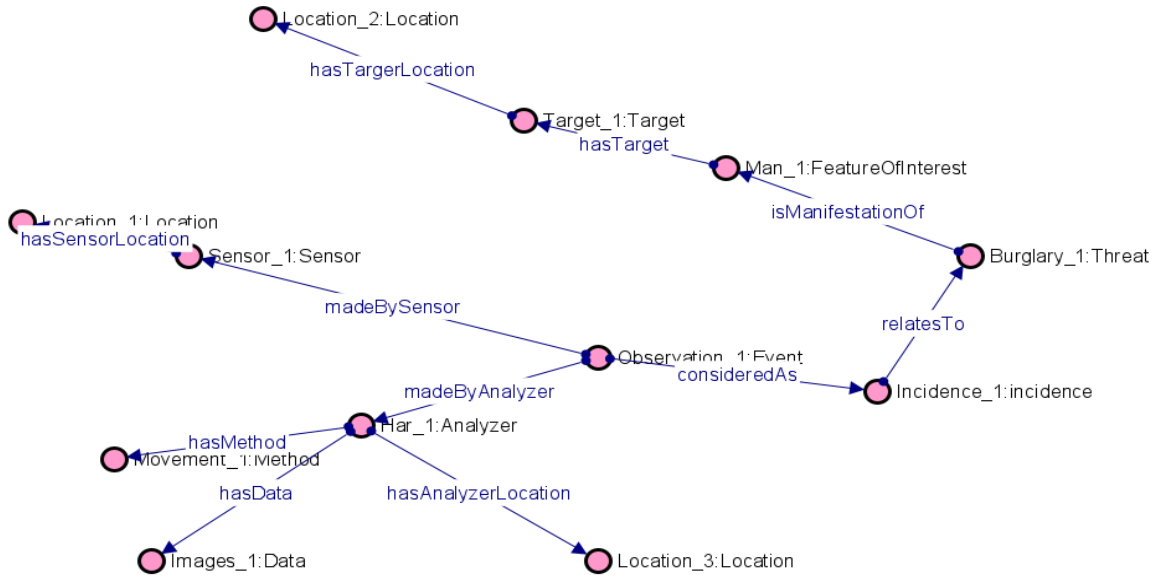

Figure 4–5 - Representation of a specific instance mapping

As it was mentioned before there are two branches of the ontology that have been conceptually designed but due to the lack of incoming data they are not implemented yet (it is quite possible for them to be restructured). In the following figures we depict the concepts behind Risk assessment and the Mitigation plan. In the Figure 4–6, the Critical Infrastructure is considered as one of the Valuable Assets and consequently has some data regarding the risk behind it (Vulnerability, Likelihood, Expected Impact). These risk data are also related to the possible threats.

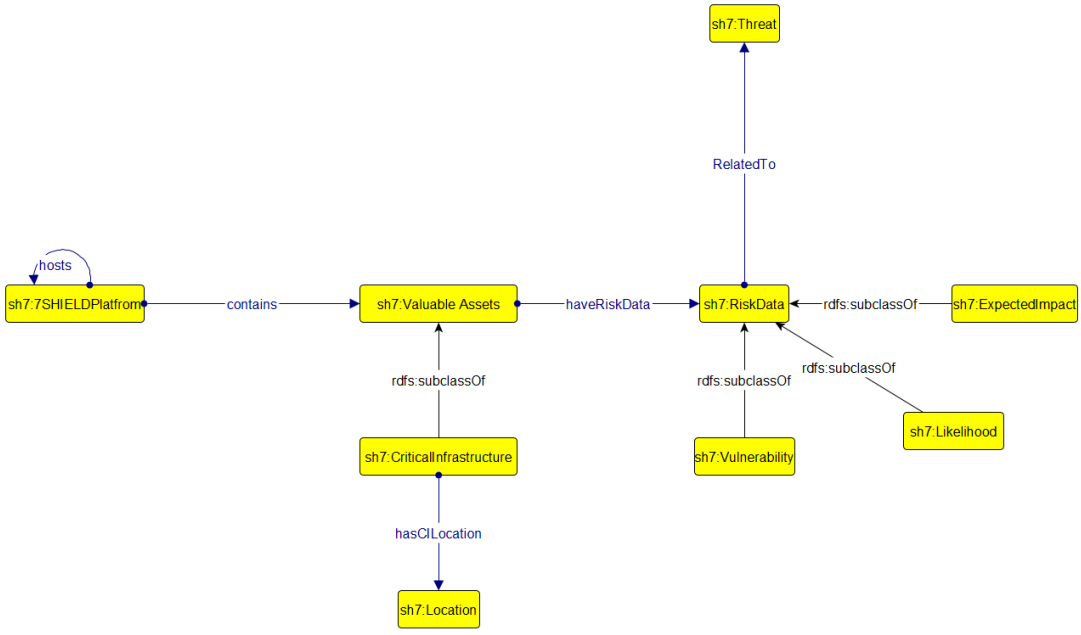

Figure 4–6 - Representation of Risk Analysis concept

As for the Mitigation Plan it is presented in the following Figure 4–7. Regarding the incident, a mitigation plan will be constructed, and it will be executed either by First responders e.g., by Flying Hunter. The data available for each unit regarding the location, the leader, the teams and the condition are an initially to be assumed.

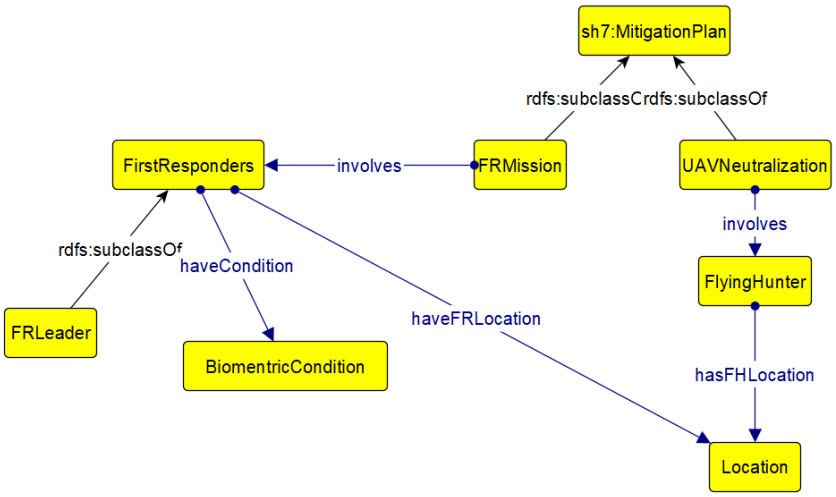

Figure 4–7 - Representation of the Mitigation Plan concept

4.3 Ontology Implementation

As it was described in Section 1 OWL 2 a knowledge representation language widely used within the Semantic Web community for developing ontologies. As a result, the 7SHIELD ontology is expressed in it and furthermore we capitalize on its wide adoption as well as its formal structure and syntax based on DL.

The tool that has been operated for the development and deployment of the ontology that we described in the previous subsections are listed in the Table 4-1.

Protégé-OWL v5.5.0	Is an open-source ontology editor and framework for building intelligent systems
GraphDB	A popular graph database for locally hosting test versions of the ontology and serving queries as a SPARQL endpoint
yEd Graph Editor	yEd is a general-purpose diagramming program that can be used to draw many different types of diagrams via an intuitive user interface with the addition of a Graphical Framework for OWL Ontologies (Graphoo). Graphoo is an open-source tool that can be used to present the classes, properties and restrictions within OWL ontologies, or sub-sections of them, as clear and easy-to-understand diagrams.

SPARQL	The semantic query language for submitting queries to the ontology and running rules on top of the knowledge base.

Table 4-1 - Implementation Tools

4.4 Ontology Evaluation

The ontology evaluation theory is a rising field of research in Ontological Engineering which allows one to cope with the problems of assessing an individual ontology from the angle of specific application aspects. The existing methods for evaluating an ontology adopt approaches either automated or semi-automated that focus on:

- Quantitative aspects: e.g., consistency, expandability, sensitiveness.
- Qualitative aspects: e.g., numbers of classes, properties, individuals.

In the work [25] four basic methodologies for ontology evaluation had been proposed. The main concept for each one of them and an example of their application are:

- Comparing the new ontology to gold standard ontologies of proven quality [26].
- Utilizing new ontology in its intended uses and confirm their functionality [27].
- Evaluating the interconnection of the new ontology and its source data [28].
- Overseeing an evaluation based on pre-defined requirements and standards [29].

None of the approaches, referred or not, have proved particularly successful neither can guarantee a good ontological framework, in yielding substantial content. Although they aim to establish the parameters of ontology evaluation, they lack the concrete criteria to gauge ontology quality. In addition, their focus on precision and recall would be better served were ontologies assessed via more systematic methodologies.

In order to evaluate our work, we used the following methods that cover the aspects of consistency, quality and structure.

4.4.1 Consistency and Quality Evaluation

For the consistency and quality evaluation of the ontology we used OOPS (OntOlogy Pitfall Scanner), an online tool for detecting the most common pitfalls in ontologies [30]. The tool, after analysing the ontology, provides a list with all the pitfalls it detected along with the associated negative consequences, and suggests modifications in order to improve the quality of the ontology. The pitfalls are categorized based on their severity to:

- **Minor**: Which do not cause any critical problems bur correcting them will improve the quality of the ontology.
- **Important**: They are quite important and affect the quality of the ontology.
- **Critical**: They are affecting the ontology's consistency and must be corrected.

We submitted the current early version (v1) of the ontology to OOPS and we have already corrected all the detected pitfalls, which were critical but were made due to accidentally wrong definitions in domain/range values of object properties and the annotation labels to the entities. Also, we noticed a similarity in the names of the categorized threats that caused an important pitfall. The current version of the ontology has no more pitfalls, with the exception of some pitfalls concerning the inverse object properties that will be corrected in the next version.

4.4.2 Structural Evaluation

For the structural part of the evaluation, we used the OntoMetrics tool, an online framework that evaluates the ontology based on predefined metrics, namely basic and schema metrics. The following tables present the results of the aforementioned process. The Table 4-2 contains the basic metrics which show the quantity of the ontology, numbers of triples, classes, object and datatype properties, individuals and DL expressivity.

Basic Metric	Value
Axioms	594
Logical axioms count	251
Class count	91
Total classes count	91
Object property count	36
Total object properties count	36
Data property count	22
Total data properties count	22
Properties count	58
DL expressivity	ALCHI(D)

Table 4-2- Basic Metrics

Initially we will comment about the base metrics, the total count of classes and properties of the 7SHIELD ontology reflects that this version is a lightweight one, which could be easily adopted by various applications, in contrast with other ontological frameworks with vast amounts of confusing interactions. Nonetheless, we have to repeat at this point that there are going to be additions and further enrichment with entities regarding the systems aspects that will be integrated later.

As for the schema metrics we used the methodological framework proposed in OntoQA ([31]) regarding the interpretation of the OntoMetrics results (Table 4-3). The following definitions were adopted:

- **Attribute richness:** the number of attributes that are defined for each class can indicate both the quality of ontology design and the amount of information pertaining to instance data. So, we assume that the more slots that are defined the more knowledge the ontology holds.
- **Inheritance richness:** This measure describes the distribution of information across different levels of the ontology's inheritance tree or the fan-out of parent classes. This is a good indication of how well knowledge is grouped into different categories and subcategories in the ontology.
- **Relationship richness**: this metric reflects the diversity of relations and placement of relations in the ontology. An ontology that contains many relations, other than class-subclass relations, is richer than a taxonomy with only class-subclass relationships.
- **Axiom/Class, Class/Relation, Inverse Relations ratio**: are indications of the ontology's transparency and understandability. Describe the relations between the aforementioned attributes (axioms, class. relation etc).

Schema Metric	Value
Attribute richness	0.241758
Inheritance richness	0.857143
Relationship richness	0.344538
Axiom/class ratio	6.527473
Inverse relations ratio	0.305556
Class/relation ratio	0.764706

Table 4-3- Schema Metrics

5 Semantic Reasoning Framework

An abstract reasoning architecture is depicted in the Figure 5–1. Briefly, we can say that the reasoning framework extends the 7SHIELD's semantic models to predefined rules that formulated based on the available context (e.g. metadata collected from the analysis results, population of the KB). The semantics are used to acquire an early understanding of the available contents and dependencies among the multimodal results in the form of interlinked data. The knowledge graphs that formed are used as an input to the reasoning tool that triggers the necessary reasoning rules to export additional knowledge. For a better understanding the reasoning framework can be seen as a scheme that combines data integration and interpretation.

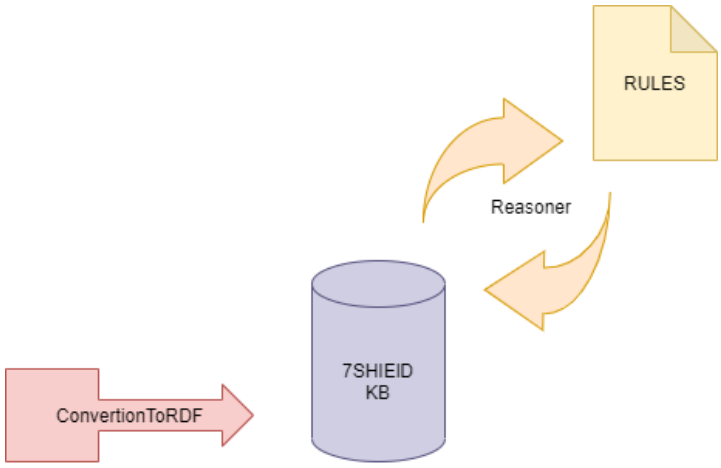

Figure 5–1 - Abstract Reasoning Architecture

Apart from semantically analysing and correlating metadata, the reasoning framework excels at providing more complex searching capabilities to the end users, elaborating the SPARQL mechanics. This module is still in progress, and it will be further refined and presented in upcoming deliverable. The subsections that follow present some tasks that will be handled by the reasoning framework.

The reasoning module is under development, and it will be refined and presented via a later WP5 deliverable. In the following subsections, some basic form of the tasks that can be handled by the reasoning framework, are presented.

5.1 Report Formulation

In this subsection we describe the first iteration of a simple rule that was implemented with SPARQL. As it was described there is the need for the Knowledge Base to export some report, so an assumption for this example that a report instance needs to be created when the severity and the confidence of an observation is above some specific values. These

reports will contain information that are already mapped in the KB but also be deduced during the reasoning.

```
PREFIX ex: <http://example.org/>
PREFIX sh7: <http://www.7shield.eu/ontology/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX sh: <http://www.w3.org/ns/shacl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>


INSERT
{
    sh7:report_status1 rdf:Type owl:NamedIndividual.
    sh7:report_status1 rdf:Type sh7:ReportStatus.
    ?Observation sh7:hasReport sh7:Report_Status1.

}
 where {
?Observation sh7:hasSeverity ?severity .
?Observation sh7:hasConfidence  ?confidence.
FILTER ((?confidence>"0.5"^^xsd:double) && (?severity="Medium") )
}
```

Figure 5–2 - Sample Rule for creating an Instance of report

6 Ontology Validation

In this following section a 7SHIELD annotation model is presented in order to map the outcome of *Task 4.7 – Combined Physical and Cyber Threat Detection and Early Warning*. In this regard we took in account a simulation example which were provided by the technical partners. The aforementioned simulation example was related to generated results which guided us to generate the annotation vocabularies. The following JSON was given as input and accordingly the TURTLE RDF was formed as output.

6.1 Sensor


```
{"Version": "2.0.3",
  "ID": "e5f9bbae-163e-42f9-a2f2-0daaf78fefb1",
  "Status": "Incident",
  "Entity": "ACME",
  "CreateTime": "2021-01-18T23:33:05.21Z",
  "StartTime": "2021-01-18T23:33:04.52Z",
  "Category": ["Availability.Outage"],
  "Confidence": 0.5,
  "Severity": "Medium",
  "Description": "CRITICAL - 172.10.5.31: rta nan, lost 100%",
  "Analyzer": {
    "IP": "172.10.1.1",
    "Name": "Local NMS",
    "Hostname": "nms.example.com",
    "Type": "Availability",
    "Model": "Super NMS v5.2",
    "Category": ["NMS"],
    "Data": ["Protocol"],
    "Method": [ "Threshold"] },
  "Sensor": [{
      "IP": "172.10.5.1",
      "Name": "Security Center",
      "Hostname": "sec-center.example.com",
      "Model": "Security Center v5.7",
      "UnLocation": "GR ATH",
      "Location": "Security Office"} ],
  "Target": [
    { "IP": "172.10.5.31",
      "Hostname": "camera1.example.com",
      "GeoLocation": "+48.75726,+2.299528,+65.1",
      "UnLocation": "GR ATH",
      "Location": "Front Door" } ]  }
```


Figure 6–1 - JSON Example Availability Event

To create an observation, primarily we need to detect the event and afterwards to analyse the event. In the above JSON example, we use as input an availability event. The information we pull from the JSON are IP and Name of the sensor. In the picture below it appears the example for the sensor.

	subject	predicate	object
1	sh7:sensor_2	sh7:hasSensorIp	"172.10.5.1"
2	sh7:sensor_2	sh7:hasSensorName	"Security Center"
3	sh7:sensor_2	rdf:type	sh7:Sensor
4	sh7:sensor_2	rdf:type	owl:NamedIndividual

Figure 6–2 - Mapping the sensor in GraphDB

6.2 Analyser


```
{ "Version": "2.0.3",
     "ID": "e5f9bbae-163e-42f9-a2f2-0daaf78fefb1",
     "Entity": "EXAMPLE",
     "Category": ["Intrusions.Burglary"],
     "Cause": "Malicious",
     "Description": "Physical intrusion detected",
     "Status": "Incident",
     "Severity": "Medium",
     "CreateTime": "2021-01-18T23:33:05.21Z",
     "DetectTime": "2021-01-18T23:33:04.52Z",
     "Confidence": 0.9,
     "Analyzer": {"IP": "172.10.5.31",
       "Name": "Front door camera",
       "Hostname": "camera3.example.com",
       "Type": "Physical",
       "Model": "Netatmo 42B",
       "Category": [ "HAR"],
       "Data": ["Images"],
       "Method": [ "Movement"],
       "GeoLocation": "+48.75726,+2.299528,+65.1",
       "UnLocation": "GR ATH",
       "Location": "Front Door"},
     "Sensor": [
       { "CaptureZone": "{\"type\":\"conic-zone\",\"Azimuth\":34.00,\"El\u00e9vation\":-30.00,\"Range\":600.00,\"HFOV\":200.00,\"VFOV\":20.00}"}],
     "Target": [ {"Location": "Garden" } ],
     "Vector": [{"Category": "Human",
         "Name": "Unknown",
         "Size": "Medium",
         "Observable": ["PTZ Relative position"]}],}
```


Figure 6–3 - JSON Example Physical Event

For the next step we mention the analyser, in the above json example we have a physical event, and it has been analysed with HAR which means Human Activity Recognition. From that JSON, we get the information about the IP of the analyser, the location and name of the analyser, what method and what data it uses. As we see in the picture below, we have gathered all this information.

	subject	predicate	object
1	sh7:HAR_0	sh7:hasAnalyzerIp	"172.10.5.31"
2	sh7:HAR_0	sh7:hasAnalyzerLocation	sh7:location_0
3	sh7:HAR_0	sh7:hasAnalyzerName	"Front door camera"
4	sh7:HAR_0	sh7:usedData	"Images"
5	sh7:HAR_0	sh7:usedMethod	"Movement"
6	sh7:HAR_0	rdf:type	sh7:HAR
7	sh7:HAR_0	rdf:type	owl:NamedIndividual

Figure 6–4 - Mapping the analyser in GraphDB

6.3 Vector


```
{ "Version": "2.0.3",
    "ID": "e5f9bbae-163e-42f9-a2f2-0daaf78fefb1",
    "Entity": "EXAMPLE",
    "Category": ["Intrusions.Burglary"],
    "Cause": "Malicious",
    "Description": "Physical intrusion detected",
    "Status": "Incident",
    "Severity": "Medium",
    "CreateTime": "2021-01-18T23:33:05.21Z",
    "DetectTime": "2021-01-18T23:33:04.52Z",
    "Confidence": 0.9,
    "Analyzer": {"IP": "172.10.5.31",
      "Name": "Front door camera",
      "Hostname": "camera3.example.com",
      "Type": "Physical",
      "Model": "Netatmo 42B",
      "Category": [ "HAR"],
      "Data": ["Images"],
      "Method": [ "Movement"],
      "GeoLocation": "+48.75726,+2.299528,+65.1",
      "UnLocation": "GR ATH",
      "Location": "Front Door"},
    "Sensor": [
      { "CaptureZone": "{\"type\":\"conic-zone\",\"Azimuth\":34.00,\"El\u00e9vation\":-30.00,\"Range\":600.00,\"HFOV\":200.00,\"VFOV\":20.00}"}],
    "Target": [ {"Location": "Garden" } ],
    "Vector": [{"Category": "Human",
        "Name": "Unknown",
        "Size": "Medium",
        "Observable": ["PTZ Relative position"]}],}
```


Figure 6–5 - JSON Example Physical Event

For vector and more specifically for the Vector physical we take consideration of the above JSON example. We get the size, target and vector name information from the JSON. Whenever we understand from the data which has been collected there is a man of medium size intrude in the garden.

	subject	predicate	object
1	sh7:event_0	sh7:consideredAs	sh7:incident_0
2	sh7:event_0	sh7:leadsTo	sh7:Burglary_0
3	sh7:event_0	rdf:type	owl:NamedIndividual
4	sh7:event_0	rdfs:label	sh7:Event

Figure 6–6 - Mapping the vector in GraphDB

6.4 Event


```
{ "Version": "2.0.3",
     "ID": "e5f9bbae-163e-42f9-a2f2-0daaf78fefb1",
     "Entity": "EXAMPLE",
     "Category": ["Intrusions.Burglary"],
     "Cause": "Malicious",
     "Description": "Physical intrusion detected",
     "Status": "Incident",
     "Severity": "Medium",
     "CreateTime": "2021-01-18T23:33:05.21Z",
     "DetectTime": "2021-01-18T23:33:04.52Z",
     "Confidence": 0.9,
     "Analyzer": {"IP": "172.10.5.31",
       "Name": "Front door camera",
       "Hostname": "camera3.example.com",
       "Type": "Physical",
       "Model": "Netatmo 42B",
       "Category": [ "HAR"],
       "Data": ["Images"],
       "Method": [ "Movement"],
       "GeoLocation": "+48.75726,+2.299528,+65.1",
       "UnLocation": "GR ATH",
       "Location": "Front Door"},
     "Sensor": [
       { "CaptureZone": "{\"type\":\"conic-zone\",\"Azimuth\":34.00,\"El\u00e9vation\":-30.00,\"Range\":600.00,\"HFOV\":200.00,\"VFOV\":20.00}"}],
     "Target": [ {"Location": "Garden" } ],
     "Vector": [{"Category": "Human",
         "Name": "Unknown",
         "Size": "Medium",
         "Observable": ["PTZ Relative position"]}],}
```


Figure 6–7 - JSON Example Physical Event

In order to get the event information, we draw specific data from our JSON example. And as we observe the event that leads to burglary. We consider the event as incident_0. As we see in the table below.

	subject	predicate	object
1	sh7:event_0	sh7:consideredAs	sh7:incident_0
2	sh7:event_0	sh7:leadsTo	sh7:Burglary_0
3	sh7:event_0	rdf:type	owl:NamedIndividual
4	sh7:event_0	rdfs:label	sh7:Event

Figure 6–8 - Mapping the event in GraphDB

6.5 Observation


```
{ "Version": "2.0.3",
     "ID": "e5f9bbae-163e-42f9-a2f2-0daaf78fefb1",
     "Entity": "EXAMPLE",
     "Category": ["Intrusions.Burglary"],
     "Cause": "Malicious",
     "Description": "Physical intrusion detected",
     "Status": "Incident",
     "Severity": "Medium",
     "CreateTime": "2021-01-18T23:33:05.21Z",
     "DetectTime": "2021-01-18T23:33:04.52Z",
     "Confidence": 0.9,
     "Analyzer": {"IP": "172.10.5.31",
       "Name": "Front door camera",
       "Hostname": "camera3.example.com",
       "Type": "Physical",
       "Model": "Netatmo 42B",
       "Category": [ "HAR"],
       "Data": ["Images"],
       "Method": [ "Movement"],
       "GeoLocation": "+48.75726,+2.299528,+65.1",
       "UnLocation": "GR ATH",
       "Location": "Front Door"},
     "Sensor": [
       { "CaptureZone": "{\"type\":\"conic-zone\",\"Azimuth\":34.00,\"El\u00e9vation\":-30.00,\"Range\":600.00,\"HFOV\":200.00,\"VFOV\":20.00}"}],
     "Target": [ {"Location": "Garden" } ],
     "Vector": [{"Category": "Human",
         "Name": "Unknown",
         "Size": "Medium",
         "Observable": ["PTZ Relative position"]}],}
```


Figure 6–9 - JSON Example Physical Event

For the observation that has been triggered, we pick from the above json example the information about when the observation took place, when it was detected and what level of confidence and severity was. Also, there is a correlation of the observation with the sensor, analyser and the vector as we see in the Figure 6–10.

	subject	predicate	object
1	sh7:observation_#e5f9bbae-163e-42f9-a2f2-0daaf78fefb1	sh7:createTime	"2021-01-18T23:33:05.21"^^xsd:dateTime
2	sh7:observation_#e5f9bbae-163e-42f9-a2f2-0daaf78fefb1	sh7:detectTime	"2021-01-18T23:33:04.52"^^xsd:dateTime
3	sh7:observation_#e5f9bbae-163e-42f9-a2f2-0daaf78fefb1	sh7:hasConfidence	"0.9"^^xsd:double
4	sh7:observation_#e5f9bbae-163e-42f9-a2f2-0daaf78fefb1	sh7:hasSeverity	"Medium"
5	sh7:observation_#e5f9bbae-163e-42f9-a2f2-0daaf78fefb1	sh7:madeByAnalyzer	sh7:HAR_0
6	sh7:observation_#e5f9bbae-163e-42f9-a2f2-0daaf78fefb1	sh7:madeBySensor	sh7:sensor_0
7	sh7:observation_#e5f9bbae-163e-42f9-a2f2-0daaf78fefb1	rdf:type	owl:NamedIndividual
8	sh7:observation_#e5f9bbae-163e-42f9-a2f2-0daaf78fefb1	rdf:type	sosa:Observation
9	sh7:observation_#e5f9bbae-163e-42f9-a2f2-0daaf78fefb1	rdfs:label	"observation_#e5f9bbae-163e-42f9-a2f2-0daaf78fefb1"
10	sh7:observation_#e5f9bbae-163e-42f9-a2f2-0daaf78fefb1	sosa:hasFeatureOfInterest	sh7:Human_0

Figure 6–10 - Mapping the observation in GraphDB

7 Conclusions and Future Outlook

In this document the requirement specifications and the state-of-the-art analysis relevant to the development of the semantic knowledge structures addressed within "T5.1: The 7SHIELD ontology and data representation model" is provided. The current status of the 7SHILELD ontology towards the first prototype is also described. In addition, it was presented the knowledge base population procedure with incoming analyses results from the detector/correlator components. We also presented a basic structure of the reasoning framework with sample rules for combining, integrating, semantically interpreting and enriching the knowledge captured in the KB.

Next steps for this task that are going to be implemented until M15 (Nov. 2021) include:

1. Extension of the 7SHILED ontology, in order to fully cover the user requirements. The aspects that are not covered yet are the Mitigation Plan; with the missions to FR and to UAV neutralization, the reports of the mitigated act, the Risk assessment part; including the critical infrastructure and risk data. Also there have to be an addition to the reports that the knowledge base is going to provide through the interface.
2. New data requires the update of the population tool so it can be capable for mapping the information and adapt in the case of different structures (eg. the case the the SPGU will be the source of data for the KB)
3. Development of the reasoning framework ruleset, that includes geospatial criteria, threat related knowledge or valuable targets.
4. Integration of advanced reasoning techniques, like fuzzy ontologies or Semantic Complex Event Processing techniques.
5. Mapping the 7SHIELD ontology with other models, at the final stage of the development such kind of mappings with external frameworks will establish interoperability. This process includes the formulation of a document that contains the semantic relationships between our concepts with other vocabularies, some of which presented in Section 2.

8 References

[1] Baader F. et al. (2003). The description logic handbook: Theory, implementation and applications. Cambridge university press.

[2] Studer R. et al. (1998). Knowledge engineering: Principles and methods. Data & Knowledge Engineering. 25, 1–2, 161–197. DOI: https://doi.org/10.1016/S0169-023X(97)00056-6.

[3] Deborah L., McGuinness, F. van H. (2004). Owl web ontology language overview. W3C recommendation 10.2004-03. DOI: https://doi.org/10.1145/1295289.1295290.

[4] Horrocks I., Sattler U., Tobies S. (2000) Practical reasoning for very expressive description logics. Logic Journal of IGPL, volume 8, no. 3: pp. 239–263.

[5] Oberle D., Ankolekar A., Hitzler P. et al. (2007) DOLCE ergo SUMO: On foundational and domain models in the SmartWeb Integrated Ontology (SWIntO). Web Semantics: Science, Services and Agents on the World Wide Web, volume 5, no. 3: pp. 156–174.

[6] Perez J. et al. (2006). Semantics and Complexity of SPARQL. 30–43. DOI: https://doi.org/10.1145/1567274.1567278.

[7] Knublauch H., Hendler J. A., Idehen K. (2011). SPIN - Overview and Motivation W3C: Member Submission 22 February 2011.

[8] Compton M., Barnaghi P., Bermudez L., García Castro R., Corcho O., Cox S., Graybeal J., Hauswirth M., Henson C., Herzog A., Huang V.,Janowicz K., Kelsey D., Phuoc D., Lefort L., Leggieri M., Neuhaus H., Nikolov A., Page K., Taylor K. (2012). The SSN Ontology of the W3C Semantic Sensor Network Incubator Group. Web Semantics: Science, Services and Agents on the World Wide Web. 17. 10.1016/j.websem.2012.05.003.

[9] Janowicz, **K**., Haller, A., Cox.J.D., S., Le Phuoc, D., Lefrançois, **M**. (2019). "SOSA: A lightweight ontology for sensors, observations, samples, and actuators". Journal of Web Semantics, 56, pp. 1-10. DOI: https://doi.org/10.1016/j.websem.2018.06.003

[10]Masmoudi M., Karray H., Ben Abdallah S., Zghal H., Archimède B. (2019). MEMOn: Modular Environmental Monitoring Ontology to link heterogeneous Earth Observed data. Environmental Modelling & Software. 124. 104581. 10.1016/j.envsoft.2019.104581.

[11]Buttigieg P. L., Morrison N., Smith B. et al. (2013). The environment ontology: contextualising biological and biomedical entities. Journal of Biomedical Semantics, 4(43). https://doi.org/10.1186/2041-1480-4-43

[12]Gomez M. et al. (2008), "An Ontology-Centric Approach to Sensor-Mission Assignment," in International Conference on Knowledge Engineering and Knowledge Management, Berlin, Heidelberg.

[13]Raimond Y., Abdallah S. (2007). The event ontology. Online Available at: http://motools.sourceforge.net/event/event.html.

[14] Hendler J. (2009). Web 3.0 Emerging. Computer, 42(1), pp. 111-113. DOI: 10.1109/MC.2009.30.

[15]Liu S., Brewster C., Shaw D. (2013). Ontologies for crisis management: a review of state of the art in ontology design and usability. Proceedings of the 10th International ISCRAM Conference, Baden-Baden, Germany, May 2013, pp. 349–359.

[16]Limbu M. (2012). Management of a Crisis (MOAC) Vocabulary Specification.

[17]Babitski G., Bergweiler S., Grebner O., Oberle D., Paulheim H., Probst F. (2011). SoKNOS – Using Semantic Technologies. Disaster Management Software, pp. 183-197. 10.1007/978-3-642-21064-8_13.

[18]Atymtayeva L., Kozhakhmet K., Bortsova G. (2014). Building a Knowledge Base for Expert System in Information Security. 270. 10.1007/978-3-319-05515-2_7.

[19] Obrst L. et al. (2012). "Developing an Ontology of the Cyber Security Domain." STIDS.

[20]Hobbs J. R., Pan F. (2004). An Ontology of Time for the Semantic Web. CM Transactions on Asian Language Processing (TALIP): Special issue on Temporal Information Processing. Vol. 3, 1, pp. 66-85.

[21]Pan F., Hobbs J. R. (2004). Time in OWL-S. Proceedings of the AAAI Spring Symposium on Semantic Web Services. s.l.: Stanford University. pp. 29-36.

[22]Suárez-Figueroa M. C., Gómez-Pérez A., Villazón-Terrazas B. (2009). "How to Write and Use the Ontology Requirements Specification Document." In Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).

[23]Gruninger M. , Fox M.S. (1995). Methodology for the Design and Evaluation of Ontologies.In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95, Montreal.

[24] Janowicz K., Kelsey W., Le Phuoc D., Lefort L., Leggieri M., Neuhaus H., Nikolov A., Page K., Passant A., Sheth A., Taylor K. (2012). "The SSN Ontology of the W3C Semantic Sensor Network Incubator Group", In Web Semantics: Science, Services and Agents on the World Wide Web.

[25]Brank J., Mladenic D., Grobelnik M. (2006), "Gold standard-based ontology evaluation using instance assignment", paper presented at the 4th International Workshop on Evaluation of Ontologies for the Web (EON) at the 15th International World.

[26]Maedche A., Staab S. (2002), "Measuring similarity between ontologies", paper presented at 13th International Conference, EKAW, Sigüenza, October 1-4, available at: www.researchgate.

[27]Porzel R., Malaka R. (2004), "A task-based approach for ontology evaluation", paper presented at ECAI Workshop on Ontology Learning and Population, Valencia,

[28]Lee B., Kim H. (2013), "Design and evaluation of an individual instance-based ontology retrieval system for archival records of the 'Saemaul movement'", Journal of Korean Society of Archives and Records Management, Vol. 13 No. 3, pp. 67-97.

[29]Park J., Cho W., Rho S. (2008), "Measurement criteria for ontology extraction tools", Journal of Intelligent Information Systems, Vol. 14 No. 4, pp. 69-87.

[30]Poveda-Villalón M., Gomez-Perez A., Suárez-Figueroa M. C. (2014). OOPS! (OntOlogy Pitfall Scanner!): An on-line tool for ontology evaluation. International Journal on Semantic Web and Information Systems. 10. 7-34. 10.4018/ijswis.2014040102.

[31]Tartir S., Arpinar I., Moore M., Sheth A., Aleman-Meza B. (2005). OntoQA: Metric-Based Ontology Quality Analysis.

Appendix A - Detailed Ontology

This appendix lists the ontology classes, object properties and data properties

Classes

Name	Data
Definition	Data analysed for detection
Instance of	owl:class

Name	ReportStatus
Definition	The class ReportStatus is used to contain information that are worthy for report
Instance of	owl:class
Disjoint with	Event DataSource FeatureOfInterest Threat ValuableAssets

Name	ValuableAssets
Definition	The list of assets in the 7SHIELD systems that need to be protected
Instance of	owl:class
Disjoint with	Event DataSource ReportStatus FeatureOfInterest Threat

Name	Threat
Definition	The Threat class presents hostile action on someone or something.

Instance of	owl:class
Disjoint with	Event DataSource ReportStatus FeatureOfInterest ValuableAssets

Name	Target
Definition	The class Target presents an object of attention or attack.
Instance of	owl:class

Name	Source
Definition	The Source of an incoming threat
Instance of	owl:class

Name	RiskData
Definition	Contains the information regarding the risk that each Critical Infrastructure has.
Instance of	owl:class

Name	Platform
Definition	A Platform is an entity that hosts other entities, particularly Sensors, Actuators, Samplers, and other Platforms.
Instance of	owl:class

Name	MitigationPlan
Definition	The plan that will be created to face each respective incident
Instance of	owl:class

Name	Method
Definition	The class Method contains all the methods of analyser
Instance of	owl:class

Name	Incident
Definition	The class Incident is a sub-class of the Event. An incident indicates to a particular happening which is noteworthy
Instance of	owl:class

Name	Location
Definition	Location class presents the place or position that something is in or where something happens:
Instance of	owl:class
Disjoint with	GeoLocation PhysicalLocation Unlocation

Name	FR
Definition	This class represents First responder units
Instance of	owl:class
Disjoint with	FlyingHunter

Name	FlyingHunter
Definition	The drone that is responsible for UAV Neutralization
Instance of	owl:class
Disjoint with	FR

Name	DataSource

Definition	An entity that combines the analysers and the sensors
Instance of	owl:class
Disjoint with	Event ReportStatus FeatureOfInterest Threat ValuableAssets

Name	Event
Definition	The class Event is one of the primaries of the overall data model of the information sharing environment. Event is an abstract entity which has a sub-entity, the Observation
Instance of	owl:class
Disjoint with	DataSource ReportStatus FeatureOfInterest Threat ValuableAssets

Name	FeatureOfInterest
Definition	The entity that is value we want to observe; also, it triggers and observation
Instance of	owl:class
Disjoint with	Event DataSource ReportStatus Threat ValuableAssets

Name	Analyzer
Definition	The class Analyzer presents a piece of equipment used to analyse data from different sources and to draw conclusions
Subclass of	DataSource

Name	AvailabilityAnalyzer
Definition	AvailabilityAnalyzer contains all the analysis which related to availability actions
Subclass of	Analyzer

Name	NMS
Definition	AvailabilityAnalyzer contains all the analysis which related to availability actions
Subclass of	AvailabilityAnalyzer PhysicalAnalyzer

Name	CyberAnalyzer
Definition	The class CyberAnalyzer contains all the analysis which related to cyber actions
Subclass of	Analyzer
Disjoint with	PhysicalAnalyzer

Name	AV
Definition	Represents Detects malware (signature)
Subclass of	CyberAnalyzer

Name	EDR
Definition	Represents Endpoint Detection and Response
Subclass of	CyberAnalyzer

Name	FW
Definition	Represents Firewall
Subclass of	CyberAnalyzer

Name	HIDS

Definition	Represents Host Intrusion Detection System
Subclass of	CyberAnalyzer

Name	LOG
Definition	Represents Log analysis
Subclass of	CyberAnalyzer

Name	NIDS
Definition	Represents Network Intrusion Detection System
Subclass of	CyberAnalyzer

Name	SPAM
Definition	Represents Detect Spam, Phishing, etc.
Subclass of	CyberAnalyzer

Name	WIDS
Definition	Represents Wifi Intrusion Detection System
Subclass of	CyberAnalyzer

Name	PhysicalAnalyzer
Definition	The class PhysicalAnalyzer contains all the analysis which related to physical actions
Subclass of	Analyzer
Disjoint with	CyberAnalyzer

Name	ADS
Definition	Represents Anti Drone System

Subclass of	PhysicalAnalyzer

Name	FRC
Definition	Represents Face Recognition Camera
Subclass of	PhysicalAnalyzer

Name	HAR
Definition	Represents Human Activity Recognition
Subclass of	PhysicalAnalyzer

Name	MWIR
Definition	Represents Middle Wavelength InfraRed
Subclass of	PhysicalAnalyzer

Name	NMS
Definition	Represents NMS
Subclass of	PhysicalAnalyzer

Name	ODC
Definition	Represents Object Detection Camera
Subclass of	PhysicalAnalyzer

Name	VAD
Definition	Represents Voice Activity Detection
Subclass of	PhysicalAnalyzer

Name	Sensor

Definition	A Sensor is an instrument that observes a property or phenomenon with the goal of producing an estimate of the value of a parameter.
Subclass of	DataSource

Name	Observation
Definition	The act made by a Datasource (sensor, analyzer) in order
Subclass of	Event

Name	AvailabilityVector
Definition	The class AvailabilityVector is used to characterize all the availability vector entities
Subclass of	FeatureOfInterest

Name	CyberVector
Definition	The class CyberVector is used to characterize all the cyber vector entities
Subclass of	FeatureOfInterest

Name	PhysicalVector
Definition	The class PhysicalVector is used to characterize all the physical vector entities
Subclass of	FeatureOfInterest

Name	Artifact
Definition	Represents an Artifact
Subclass of	CyberVector

Name	Autonomous System
Definition	Represents an autonomous System
Subclass of	CyberVector

Name	Directory
Definition	Represents a DIrectory
Subclass of	CyberVector

Name	Domain Name
Definition	Represents Domain Name
Subclass of	CyberVector

Name	Email Addr
Definition	Represents an email address
Subclass of	CyberVector

Name	Email Message
Definition	Represents an email message
Subclass of	CyberVector

Name	Drone
Definition	represent a drone
Subclass of	PhysicalVector

Name	Face
Definition	Represents a face to be recognised
Subclass of	PhysicalVector

Name	High Temperature
Definition	Represents the parameter of high temperature

Subclass of	PhysicalVector

Name	Human
Definition	Represents a detected Human
Subclass of	PhysicalVector

Name	Running Man
Definition	Represents a recognised activity of a man running
Subclass of	PhysicalVector

Name	Man
Definition	Represents a recognised Man
Subclass of	Human

Name	Woman
Definition	Represents a recognised Woman
Subclass of	Human

Name	BiometricCondition
Definition	Represents the biometric condition of the FR members. Data will be available through the sensors they wear
Subclass of	FR

Name	FRLeader
Definition	The information regarding the Leader of each First responder team
Subclass of	FR

Name	GeoLocation

Definition	The location in coordinates
Subclass of	Location

Name	PhysicalLocation
Definition	Location of a physical object
Subclass of	Location

Name	Unlocation
Definition	The class Unlocation is used to characterize geographic coding scheme which is the United Nations Code for Trade and Transport Locations
Subclass of	Location

Name	FRMission
Definition	The mission that sends to FR in order to mitigate and incident
Subclass of	MitigationPlan

Name	UAVNeutralization
Definition	The mission given to the Flying hunter in order to retrieve a enemy drone
Subclass of	MitigationPlan

Name	7SHIELDPlatform
Definition	The class 7SHIELDPlatform hosts other entities, particularly Sensors, Detectors, Samplers
Subclass of	Platform

Name	ExpectedImpact
Definition	The impact that will be if the specific asset is attacked

Subclass of	RiskData

Name	Likelihood
Definition	The possibility of a specific asset to be targeted
Subclass of	RiskData

Name	Vulnerability
Definition	The vulnerable parts of an asset
Subclass of	RiskData

Name	CriticalInfastructure
Definition	The infrastructures that are of high value and possible targets of attacks
Subclass of	ValuableAssets

Name	AvailabilityOrigin
Definition	the threat tha origins in the availability of the system
Subclass of	Threat

Name	PhysicalOrigin
Definition	PhysicalOrigin class presents any circumstance or event with the potential to harm facilities
Subclass of	Threat
Disjoint with	CyberOrigin

Name	CyberOrigin
Definition	CyberOrigin class presents any circumstance or event with the potential to harm an information system through unauthorized access, destruction, disclosure,

	modification of data, and/or denial of service
Subclass of	Threat
Disjoint with	PhysicalOrigin

Name	Availability
Definition	Represents Availability
Subclass of	AvailabilityOrigin

Name	Outage
Definition	Represents Outage
Subclass of	Availability

Name	Attempt
Definition	Represents all the kind of cyberAttempt
Subclass of	CyberOrigin

Name	Login
Definition	Represents Login attempts
Subclass of	Attempt

Name	Intrusion
Definition	Represents all the cyber-Intrusions
Subclass of	CyberOrigin
Disjoint with	Intrusions

Name	SysCompromise

Definition	Represents SysCompromise intrusion
Subclass of	Intrusion

Name	UserCompromise
Definition	Represents UserCompromise intrusion
Subclass of	Intrusion

Name	Malicious
Definition	Represents all Malicious incoming threats
Subclass of	CyberOrigin

Name	Distribution
Definition	Represents Malicious Distribution
Subclass of	Malicious

Name	System
Definition	Represents System problems
Subclass of	Malicious

Name	Recon
Definition	Represents
Subclass of	CyberOrigin

Name	Scanning
Definition	Represents Scanning Recon
Subclass of	Recon

Name	Scam
Definition	Represents Scams
Subclass of	CyberOrigin

Name	Intrusions
Definition	Represents all the physical intrusion types
Subclass of	PhysicalOrigin
Disjoint with	Intrusion

Name	Burglary
Definition	Represents Burglary Intrusion
Subclass of	Intrusion

Name	Meteorological
Definition	Represents Meteorological hazards
Subclass of	PhysicalOrigin

Name	Storm
Definition	Represents a Storm hazard
Subclass of	Meteorological

Name	Sabotage
Definition	Represents the possible Sabotages
Subclass of	PhysicalOrigin

Name	*Other*

Definition	*Represents classes that will be defined during the course of the implementation*
Subclass of	*PhysicalOrigin*

Name	*Undetermined*
Definition	*Represents classes that will be defined during the course of the implementation*
Subclass of	*Other*

Object properties

Name	hasDataSourceLocation
Instance of	owl:ObjectProperty
Domain	DataSource
Range	Location

Name	hasFHLocation
Instance of	owl:ObjectProperty
Domain	FlyingHunter
Range	Location

Name	hasSensorLocation
Instance of	owl:ObjectProperty
Domain	Sensor
Range	Location

Name	hasTargetLocation
Instance of	owl:ObjectProperty
Domain	Target

Range	Location

Name	haveCondition
Instance of	owl:ObjectProperty
Domain	FR
Range	BiometricCondition

Name	involves
Instance of	owl:ObjectProperty
Domain	MitigationPlan
Range	FR
Inverse of	involvedBy

Name	usedData
Instance of	owl:ObjectProperty
Domain	Analyzer
Range	Data
Inverse of	dataUsedBy

Name	usedMethod
Instance of	owl:ObjectProperty
Domain	Analyzer
Range	Method
Inverse of	methodUsedBy

Name	hasFeatureOfInterest

Instance of	owl:ObjectProperty
Domain	Observation
Range	FeatureOfInterest

Name	hosts
Instance of	owl:ObjectProperty
Domain	Platform
Range	DataSource
Inverse of	hostedBy

Name	analyzerMades
Instance of	owl:ObjectProperty
Domain	Analyzer
Range	Observation
Inverse of	madeByAnalyzer

Name	areContained
Instance of	owl:ObjectProperty
Domain	ValuableAssets
Range	7SHIELDPlatform
Inverse of	contains

Name	consideredAs
Instance of	owl:ObjectProperty
Domain	Event
Range	Incident

Name	contains
Instance of	owl:ObjectProperty
Domain	7SHIELDPlatform
Range	ValuableAssets

Name	dataUsedBy
Instance of	owl:ObjectProperty
Domain	Data
Range	Analyzer

Name	hasAgentLocation
Instance of	owl:ObjectProperty
Domain	Analyzer
Range	Location

Name	hasReport
Instance of	owl:ObjectProperty
Domain	Event
Range	ReportStatus
Inverse of	isReportedBy

Name	hasRisks
Instance of	owl:ObjectProperty
Domain	CriticalInfastructure
Range	RiskData

Name	hasSource
Instance of	owl:ObjectProperty
Domain	FeatureOfInterest
Range	Source

Name	hasTarget
Instance of	owl:ObjectProperty
Domain	FeatureOfInterest
Range	Target

Name	hostedBy
Instance of	owl:ObjectProperty
Domain	DataSource
Range	Platform

Name	involvedBy
Instance of	owl:ObjectProperty
Domain	FR
Range	MitigationPlan

Name	isManifestationOf
Instance of	owl:ObjectProperty
Domain	Threat
Range	FeatureOfInterest
Inverse of	manifests

Name	isNeededFor
Instance of	owl:ObjectProperty
Domain	MitigationPlan
Range	Incident
Inverse of	needsA

Name	isReportedBy
Instance of	owl:ObjectProperty
Domain	ReportStatus
Range	Event

Name	leadsTo
Instance of	owl:ObjectProperty
Domain	Event
Range	Threat
Inverse of	ledBy

Name	ledBy
Instance of	owl:ObjectProperty
Domain	Threat
Range	Event
Inverse of	ledBy

Name	madeByAnalyzer
Instance of	owl:ObjectProperty
Domain	Observation

Range	Analyzer

Name	madeByDataSource
Instance of	owl:ObjectProperty
Domain	DataSource
Range	Event

Name	madeBySensor
Instance of	owl:ObjectProperty
Domain	Observation
Range	Sensor
Inverse of	sensorMades

Name	manifests
Instance of	owl:ObjectProperty
Domain	FeatureOfInterest
Range	Threat

Name	methodUsedBy
Instance of	owl:ObjectProperty
Domain	Method
Range	Analyzer

Name	needsA
Instance of	owl:ObjectProperty
Domain	Incident

Range	MitigationPlan

Name	relatedTo
Instance of	owl:ObjectProperty
Domain	Observation
Range	Incident

Data properties

Name	eventDescription
Definition	The description of an event/mitigation
InstanceOf	owl:DatatypeProperty
Domain	ReportStatus
Range	xsd:string

Name	hasAnalyzerHostname
Definition	Host name of the analyser that makes the detection
InstanceOf	owl:DatatypeProperty
Domain	Analyzer
Range	xsd:string

Name	hasAnalyzerIp
Definition	The Ip of the analyser that makes the detection
InstanceOf	owl:DatatypeProperty
Domain	Analyzer
Range	xsd:string

Name	hasData
Definition	The type of the dataset that used by analyser
InstanceOf	owl:DatatypeProperty

Domain	Data
Range	xsd:string

Name	hasEventId
Definition	The id of the event of one observation
InstanceOf	owl:DatatypeProperty
Domain	Event
Range	xsd:string

Name	hasGeolocation
Definition	The coordinates of one specific location
InstanceOf	owl:DatatypeProperty
Domain	GeoLocation
Range	xsd:string

Name	hasSensorIp
Definition	The Ip of the sensor that makes the observation
InstanceOf	owl:DatatypeProperty
Domain	Sensor
Range	xsd:string

Name	hasLocation
Definition	The name of the area in a specific location
InstanceOf	owl:DatatypeProperty
Domain	Location
Range	xsd:string

Name	hasType
Definition	The type of the analyser, consequently the type of observation
InstanceOf	owl:DatatypeProperty

Domain	Observation
Range	{"Availability","Cyber""Physical"}

Name	hasUnlocation
Definition	The genera location of the event
InstanceOf	owl:DatatypeProperty
Domain	Unlocation
Range	xsd:string

Name	hasVectorName
Definition	The name of the entity that triggers an observation
InstanceOf	owl:DatatypeProperty
Domain	PhysicalVector
Range	xsd:string

Name	hasSensorName
Definition	The name of the sensor that made the observation
InstanceOf	owl:DatatypeProperty
Domain	Sensor
Range	xsd:string

Name	createTime
Definition	The timestamp when the observation data was created
InstanceOf	owl:DatatypeProperty
Domain	Observation
Range	xsd:dateTime

Name	detectTime
Definition	The time when the detection occurred
InstanceOf	owl:DatatypeProperty

Domain	Observation
Range	xsd:dateTime

Name	hasSize
Definition	The size of the physical vector that was detected
InstanceOf	owl:DatatypeProperty
Domain	PhysicalVector
Range	{ "Huge","Large","Medium","Small"}

Name	hasSeverity
Definition	The severity of the observation that was calculated by other systems components
InstanceOf	owl:DatatypeProperty
Domain	Observation
Range	{"High","Info","Low","Medium"}

Name	hasSourceIp
Definition	The ip of the source of the incoming threat
InstanceOf	owl:DatatypeProperty
Domain	Source
Range	xsd:string

Name	hasTargetIp
Definition	The Ip of the targeted asset of the incoming threat
InstanceOf	owl:DatatypeProperty
Domain	Target
Range	xsd:string

Horizon 2020
European Union Funding
for Research & Innovation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 883284