FFO: AN ONTOLOGICAL DECISION SUPPORT SYSTEM FOR FOREST FIRE ALERT AND MANAGEMENT SERVICES

A THESIS

Submitted in fulfillment of the requirements for the award of the degree
of

Master of Science (Research)

by

Shibani Das

DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY INDORE

JUNE 2023

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **FFO: An Ontological Decision Support System for Forest Fire Alert and Management Services**in the fulfillment of the requirements for the award of the degree of **MASTER OF SCIENCE (RESEARCH)** and submitted in the **DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from August2021 to June 2023 under the supervision of Prof. Abhishek Srivastava, Professor, IIT Indore.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Shibani Das 05/06/23

Signature of the student with date

SHIBANI DAS

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Abhishek Srivastava 05/06/23

Signature of the Supervisor of MS (Research) thesis (with date)

PROF. ABHISHEK SRIVASTAVA

SHIBANI DAS has successfully given her MS (Research) Oral Examination held on **May 24, 2024**.

18/06/24

Signature of Chairperson (OEB) with date

Abhishek Srivastava 06/06/24

Signature(s) of Thesis Supervisor(s) with date

Ayan Mondal 12/06/24 (ACTING)

Signature of Convener, DPGC with date

Ayan Mondal 12/06/24 (ACTING)

Signature(s) of head of Discipline with date

ACKNOWLEDGEMENTS

Embarking on a thesis journey requires dedication, resilience, and the invaluable support of many individuals, as it is an arduous and demanding undertaking. As I reflect on my journey, I express my gratitude to all the people who have significantly made this journey smooth and helped me in turning it into an accomplishment.

Firstly, I would like to express my deepest gratitude to my thesis supervisor, Prof. Abhishek Srivastava, for his guidance and constant support at each and every step of my research journey. His invaluable feedback, constructive criticism, and proper expert guidance have truly helped me in paving the right path through and toward the completion of my work. His motivation and encouragement have always guided me to think and work in the proper way and not get discouraged even if I have failed to get results as expected at times.

Secondly, I would like to thank the members of my thesis committee, Dr. Neminath Hubballi, Prof. Nirmala Menon, Dr. Somnath Dey, Dr. Aniruddha Singh Kushwaha, Dr. Bodhisatwa Mazumdar, and Dr. Ayan Mondal for their valuable evaluation and critical feedback. I would like to extend my gratitude towards the present CSE Department Head, Dr. Ranveer Singh and the Chairperson of the Oral Examination Board, Dr. Amod C Umarikar for their support and guidance through my final thesis work and defense. Their insightful suggestions have helped me immensely to look into the bigger picture and improve my work step by step toward practical feasibility. Their expertise in their respective research areas and guidance have helped me to come up with a holistic model and produce a quality thesis.

I would like to thank the Ubiquitous Laboratory members, Ankit Sir, Rupendra Sir, Shekhar Sir, and Drishti for extending their help and support whenever I needed it during my journey. I have learned a lot from my laboratory members, mainly from their research experiences that they used to share with me. Their invaluable suggestions have helped me to a great extent in shaping my research journey. I am grateful to have mentors like Ankit Sir and Rupendra Sir, who have consistently provided guidance

and support in every little aspect I sought assistance. Their knowledge and dedication have been a constant source of inspiration to me.

I am also grateful to all the faculties and staff of the Computer Science Department at IIT Indore for providing me with a stimulating and supportive environment to pursue my research work. In this regard, I would like to specially mention Shailendra Verma Sir for his constant help and support for which I have never faced any difficulties during my experiments. I am grateful for all the resources, infrastructure, and opportunities that have been provided to me by the department which has made my research journey smooth.

I extend my gratitude to my family and friends for believing in my potential and for their love, care, encouragement, and support throughout my entire journey. I am sincerely thankful to my parents for without them, I could not have come this far. My friends have been one of the greatest support systems who believed in me and that kept me motivated even in the most challenging times. Special mentions include Bharath, Ashok Daa, Ananya, Arpita, Snehalata, Tanisha, Nisha, Achint, Vaishnavi, Shubhi, Deepak, Mithun, and Priyanshu who are from the campus. Those who are outside the campus include Surasmita, Antara, and Sukanya.

Lastly, I would like to thank all the individuals who have contributed to my research journey directly or indirectly. Thesis journey was not a cakewalk but with the support and guidance of these people, it turned out to be a rewarding experience.

ABSTRACT

Forest fires, also known as wildfires, present a significant danger to property, human lives, and the natural environment. In order to minimize the impact caused by these emergencies, it is crucial to detect and mitigate them at an early stage. Unfortunately, there are often shortcomings in the detection mechanisms, leading to delayed responses and increased destruction. These detection anomalies can stem from sensor defects or a lack of information interoperability among the sensors deployed in forest areas. To address these challenges, this paper proposes a lightweight ontological framework. The main objective of this framework is to improve forest fire detection and management. One of the key issues in achieving effective interoperability is the heterogeneity in technologies employed and the diverse data generated by different sensors. In response to this problem, the Forest Fire Detection and Management Ontology, FFO, is introduced as a standardized model for sharing and reusing knowledge and data across various sensor systems. To validate the proposed ontology, semantic reasoning, and query processing techniques are utilized. Real-time data collected from experiments conducted in a forest setting are stored as RDF (Resource Description Framework) triples, aligning with the design principles of the ontology. By applying reasoning and querying processes to this data, the effectiveness of FFO in early wildfire detection and subsequent process management is demonstrated. The outcomes of the queries and inferences resulting from the reasoning process provide evidence that FFO is a viable solution for timely wildfire detection. Moreover, it facilitates efficient management processes once a fire has been detected. By leveraging the standardized ontology, stakeholders involved in forest fire prevention and response can enhance their capabilities in terms of detection, decision-making, and resource allocation, thereby reducing the severity of wildfire impacts on both human and natural systems.

LIST OF PUBLICATION

1. **S. Das**, A. Srivastava. FFO*: Forest Fire Ontology and Reasoning System for Enhanced Alert and Management Services*, International Journal of Web & Semantic Technology (Published).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

Here are the terms arranged in alphabetical order:

1. BUI — Build-up Index
2. CO2 — Carbon Dioxide
3. CQ — Competency Question
4. DC — Drought Code
5. DL — Description Logics
6. DMC — Deep Moist Convection
7. DUL — DOLCE-Ultra Lite
8. ER — Emergency Responders
9. FFMC — Fine Fuel Moisture Code
10. FFO — Forest Fire Ontology
11. FWI — Fire Weather Index
12. GML — Geography Markup Language
13. GPS — Global Positioning System
14. ISI — Initial Speed Index
15. JSON — JavaScript Object Notation
16. JSON-LD — JavaScript Object Notation for Linking Data
17. OWL — Web Ontology Language
18. RDF — Resource Description Framework
19. SOSA — Sensor, Observation, Sample, and Actuator
20. SPARQL — SPARQL Protocol and RDF Query Language
21. SRS — Spatial Reference System
22. SSN — Semantic Sensor Network
23. SWRL — Semantic Web Rule Language
24. URI — Uniform Resource Identifier
25. W3C — World Wide Web
26. WKT — Well-Known Text
27. XML — Extensible Markup Language
28. XSD — XML Schema Definition

Chapter 1

Introduction

1.1 Overview

Forest fires pose a significant danger to human life, property, and the environment, with millions of hectares of forests lost each year due to fires. In 2021 alone, more than 56,000 wildfires ravaged over 4.7 million acres of land, marking the highest number of fires in the Amazon rainforest in a decade. The timely detection and communication of emergency events are crucial as they enable swift response and intervention, which can greatly reduce the extent of destruction and minimize the impact on wildlife and natural habitats. When emergency events such as wildfires, floods, or oil spills are detected early, authorities can take immediate action to contain and control the situation, preventing it from spreading or escalating further. Early detection also allows for the initiation of rescue efforts, ensuring the safety, and well-being of animals that may be directly affected by the emergency. By communicating emergency events in a timely manner to the public through various channels, individuals can be made aware of the situation and take necessary precautions to ensure their safety.

One of the main challenges in detecting forest fires is the potential for delays in identification and communication. Emergency Responders (ERs) use a variety of heterogenous information obtained from various systems and distinct technologies, which causes communication issues and uncertainty [1]. Addressing these challenges requires efficient and interoperable systems that can access and share information about the environment in real-time.

To overcome these difficulties, researchers have developed different methods for forest fire detection, ranging from traditional approaches like human observers to advanced technologies such as remote sensing and artificial intelligence. However, the key lies in quickly accessing and sharing the environmental information collected from multiple observation sources with a central monitoring system.

Semantic Web technologies provide a solution to the interoperability issues faced in forest fire detection and mitigation efforts. These technologies leverage ontologies and information retrieval using SPARQL (SPARQL Protocol and RDF Query Language) [2] to enable seamless data sharing and knowledge reuse across different systems and organizations. Ontologies can facilitate the sharing and reuse of knowledge across different systems and organizations, which can help to improve the overall effectiveness of forest fire detection and mitigation efforts. Ontologies help make the systems interoperable by standardizing the forest fire-related data used in the different devices that gather information. Ontologies store information using the RDF (Resource Description Framework) triple format (*subject-predicate-object*). RDF is the data model for Semantic Web. Through the SPARQL Protocol and RDF Query Language (SPARQL), we are able to query this RDF data. Ontologies also facilitate Semantic Reasoning through which new knowledge can be derived or new relationships can be inferred based on the existing domain knowledge or data and some given rules. Rules provide a set of logical statements that express relationships and constraints within a domain.

Therefore, ontologies play a very important role in the realm of natural disasters like forest fires, where not only a robust alert system is required, but also timely communication of this information is very crucial. We discuss the basic concepts involved in the work below.

1.2 Semantic Web

The Semantic Web is an extension of the World Wide Web that aims to make web content more meaningful and machine-readable. It is an initiative to enhance how information is organized, linked, and interpreted on the internet, enabling computers to understand and process the content more intelligently and automated manner. At its core, the Semantic Web promotes the use of structured data and metadata that provide additional context and meaning to web resources. This is achieved through the use of standardized technologies such as Resource Description Framework (RDF), which provides a framework for describing resources and their relationships using *subject-predicate-object* triples. RDF forms the basis for representing knowledge in a machine-readable format. There are standards that are employed in Semantic Web. These are as follows:

1. RDF Data Model
2. SPARQL Query Language
3. RDF Schema and OWL standards for storing vocabularies and ontologies.
4. Every resource is identified using a URI (Uniform Resource Identifier)

RDF is the data model for Semantic Web which is the basic triple data structure. SPARQL is a query language used to query the RDF data. RDF Schema is an extension of RDF. It is a data-typing model of RDF. OWL (Web Ontology Language), recommended by World Wide Web (W3C) Consortium is a Semantic Web language that provides a standardized way to represent and reason about knowledge in a formal and machine-readable format. It is designed to express rich and complex ontologies, which capture the concepts, relationships, and constraints within a particular domain of interest.

While the Semantic Web was initially developed to enhance web resources, its core principles of standardized data representation, ontology-based knowledge modelling, interoperability, and semantic reasoning can be applied beyond the web context. In the case of forest fire detection, these principles can be adapted to enable better data integration, knowledge sharing, and decision support among different systems and stakeholders involved in mitigating and responding to forest fire incidents. Data integration from diverse data sources used in forest fire detection can be done by transforming the data into RDF triples and aligning them with ontologies, information from various sources like ground-based observations can be linked together, providing a holistic view of the forest fire situation. Ontologies facilitate knowledge sharing by capturing the domain-specific knowledge related to forest fire detection. Ontologies can be used for reasoning as well as decision support in forest fire detection.

1.3 Resource Description Framework

The Resource Description Framework (RDF) is a World Wide Web (W3C) standard framework for representing and exchanging structured information. It provides a flexible and extensible model for describing resources, their attributes, and the relationships between them.

RDF is based on a graph data model, where information is represented as a collection of interconnected triples. Each triple consists of three components: subject, predicate, and object. The subject represents the resource being described, the predicate represents the attribute or relationship of the resource, and the object represents the value or target of the attribute or relationship. The components of an RDF triple are typically represented using Uniform Resource Identifiers (URIs), which uniquely identify resources, and literals, which represent values such as strings or numbers. URIs can be used to refer to web resources or to create unique identifiers for concepts, properties, or individuals in a knowledge representation system.

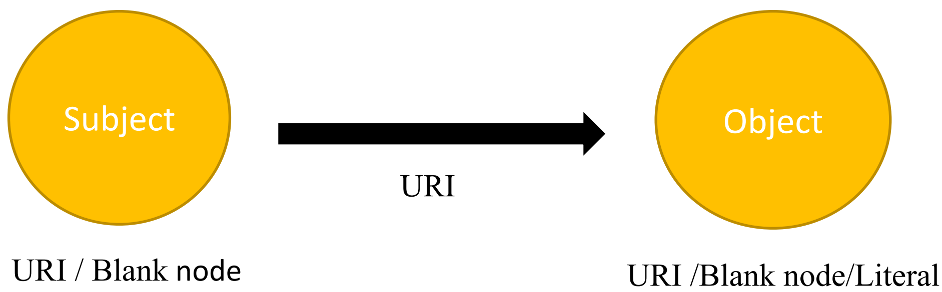

Fig. 1.1 The RDF triple structure

In the above figure, we can see that subject and object are represented as nodes and the edge represents the property or relationship between the subject and the object. A triple is like a graph with two nodes and an edge. Therefore, this representation is often called an RDF Graph.

For example, we can have

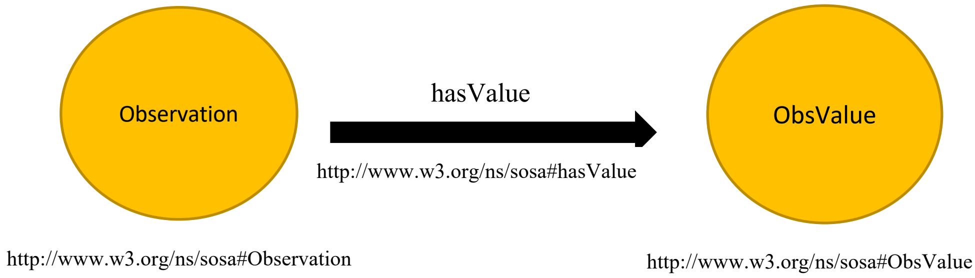

Fig. 1.2 An RDF triple example showing subject (URI), predicate (URI) and an object (URI).

RDF provides a way to describe resources and their properties in a machine-readable format. It enables the creation of semantic relationships and the integration of data from multiple sources. By using RDF, data can be linked and connected across different domains and applications, enabling interoperability and data integration.

1.4 Resource Description Framework Schema

Resource Description Framework Schema (RDF Schema) is a particular language and specification in the Semantic Web technology stack that provides a framework for defining the structure and vocabulary of RDF data. RDF Schema builds upon the basic capabilities of RDF by introducing a vocabulary for defining schemas and providing a foundation for representing knowledge in a more structured manner. RDF Schema allows the definition of classes (*rdfs:Class)*, which represent categories or sets of resources with similar characteristics. For example, *foaf:Person* is a *rdfs:Class*. *Person* is a resource in the foaf ontology [3]. RDF Schema also defines properties or relationships between resources as well as subclasses and subproperties. It also allows the specification of domain and range constraints for properties. *rdfs:domain* is the class of the subject in an RDF triple. *rdfs:range* declares the class or datatype of the object in an RDF triple. For example, if we have *Observation1-hasValue-45* based on the example given in Fig. 1.2, then the *rdfs:domain* is class Observation and *rdfs:range* is the datatype integer because Observation1 is a type of class

Observation and 45 belongs to the datatype integer. RDF Schema provides a framework for defining the structure and vocabulary of RDF data.

1.5 Ontologies

The term "ontology" was defined by T. R. Gruber in 1992 as “explicit specification of a conceptualization” [4].Ontologies are formal representations of concepts and relationships within a particular domain of knowledge. They give a common understanding of concepts, their meanings, and the connections between them. Ontologies have grown in significance in a variety of disciplines as they allow machines to comprehend the meaning of data and support knowledge sharing and reuse.

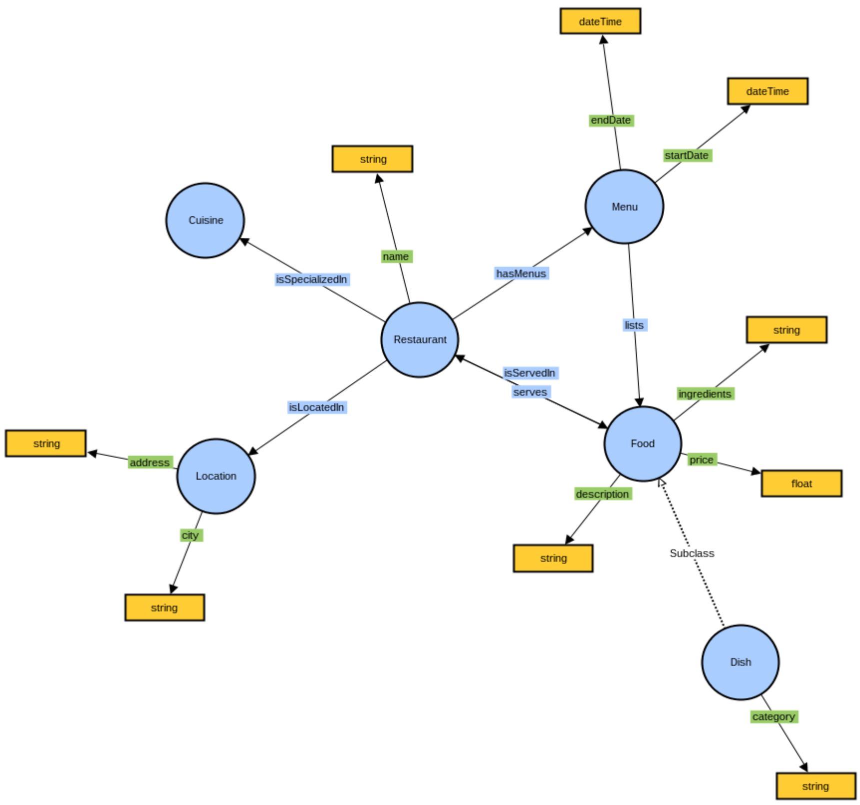

Fig. 1.3 An example of an ontology for data related to restaurant.

Ontologies include –

- Concepts or Classes: Concepts or classes represent categories or types of entities within a specific domain.

- Properties: Classes can have attributes or properties associated with them. Properties capture the relationships or associations between classes and can have domain and range constraints to specify the types of resources involved.
- Constraints on properties or attributes: Attributes can have constraints or restrictions associated with them. These constraints define rules or conditions that must be satisfied by the attribute values. Constraints ensure data integrity and enforce consistency within the ontology.
- Individuals: In ontologies, individuals are specific instances or objects within a particular class or concept. They represent concrete entities that exist within the domain being modeled. Individuals provide a way to represent and describe real-world entities, allowing for more specific and concrete knowledge representation. For example, *Dog* can be a class and our pet, *Tom*, can be the individual of the class *Dog*.

Ontologies organize concepts into taxonomies or hierarchies, where concepts are arranged in a hierarchical structure based on their generalization and specialization relationships. By hierarchies, it means, classes, subclasses, subclasses of subclasses, etc. This hierarchical structure provides a means of classification and categorization within the domain, thereby storing the data in the ontology in a more structured way. Ontologies can also include axioms or logical constraints that define rules or restrictions on the concepts, relationships, and properties. Axioms ensure consistency and logical coherence within the ontology. By defining rules, we can perform semantic reasoning based on the existing knowledge (classes and properties) and evaluate an ontology by assessing the inferences deduced by the reasoner. Ontologies also facilitate semantic querying through which we can evaluate them.

Ontologies are typically represented using formal languages, such as RDF or OWL (Web Ontology Language). This is standard data representation to which data from diverse sources are mapped. Therefore, this data integration helps systems with varying data models to communicate and exchange information effectively, facilitating interoperability between them.

The conceptualization of data, standard knowledge representation, semantic reasoning, and query processing are the reasons why ontologies are widely used in the realm of natural disasters like forest fires. Moreover, due to processes like semantic reasoning and query processing, the decision support system has been enhanced in case of such emergencies.

1.6 SPARQL

SPARQL Protocol and RDF Query Language or simply SPARQL is a W3C-recommended query language used for querying and manipulating data stored in RDF format. SPARQL is specially designed for querying RDF data. RDF represents information in a graph structure as discussed in section 1.3. SPARQL allows the expression of complex queries that traverse and retrieve data from RDF Graphs.

SPARQL has a concise syntax for specifying queries. It uses a combination of patterns, variables, and triple patterns to match and retrieve data from the RDF graph. The queries are expressed in a declarative manner, describing what data is desired rather than how to retrieve it.

For example,

```
select ?B ?A
where{
?A hasSon ?B.
}
```

This SPARQL example displays B and A such that B is the son of A. The term *select* allows users to retrieve specific data elements from the RDF graph. The *where* clause is used for patterns and conditions that must be specified in the RDF graph in order for the query to return results.

SPARQL is used in conjunction with ontologies, RDF vocabularies, and other semantic web technologies. It allows for the retrieval and integration of data from multiple sources, enabling powerful knowledge discovery and information retrieval applications.

1.7 Rules and Inferences

Rules and inferences refer to the logical reasoning and deduction processes used to derive new knowledge or conclusions based on the existing knowledge represented in the ontology. Rules and inferences play a crucial role in ontologies by enabling automated reasoning and inference capabilities.

Rules in the ontology are logical statements or axioms that define relationships, constraints, or patterns within the ontology. They specify logical conditions and actions, allowing for the deduction of new knowledge based on existing knowledge. Existing knowledge refers to the designed classes and properties in an ontology for the specific domain. Inferences are the logical deductions or conclusions deduced by a reasoner based on the rules defined and the existing knowledge in the ontology.

Reasoning in ontologies is the process of using rules to draw logical conclusions and make inferences based on the available knowledge in ontology. Rules and inferences are used to ensure the consistency of the ontology by checking for contradictions or conflicts between the asserted knowledge and the logical rules. Inconsistencies can be detected and resolved by identifying conflicting statements and removing or modifying the conflicting information.

In the forest fire detection and management context, rules and inferences can help in building decision support systems like performing management steps subsequent to fire detection. Ontologies can assist in making informed decisions based on the available knowledge and the logical deductions derived from it. We will discuss more about this in Chapter 6.

1.8 Web Ontology Language

Web Ontology Language (OWL) is a computational knowledge representation language used for creating and sharing ontologies. It is recommended by World Wide Web (W3C). It is a standardized language within the Semantic Web technology stack and is based on Description

Logics (DL), a family of logic-based knowledge representation formalisms.

OWL provides a rich set of constructs and features for modeling and representing knowledge in a structured and machine-readable format. It allows for defining classes, properties, individuals, and relationships between them, enabling the representation of complex conceptual hierarchies, constraints, and logical axioms.

Key features of OWL include:

- **Classes and instances**: OWL allows the definition of classes, which represent categories or concepts, and individuals, which represent specific instances or objects belonging to those classes. Classes can be organized into hierarchies through subclass relationships.
- **Properties**: OWL supports the definition of properties to describe relationships between classes and individuals. Properties can be of different types such as object properties (representing relationships between individuals) or data properties (representing relationships between individuals and data values).
- **Inference and Reasoning:** OWL provides a basis for logical reasoning and inference over ontologies. Reasoning engines can infer implicit knowledge based on the explicitly defined axioms and constraints in the ontology. This enables automatic classification, consistency checking, and deduction of new knowledge.
- **Axioms and Constraints:** OWL allows the specification of logical axioms and constraints to define the semantics and behaviour of the ontology. Axioms can express relationships like subclass, equivalence, and disjointness, while constraints can impose restrictions on properties and individuals.
- **Semantic Annotations**: OWL supports the annotation of ontology elements with additional metadata and documentation using standard annotation properties. This helps in providing human-

readable labels, descriptions, and comments to enhance the understanding and documentation of the ontology.

OWL ontologies are typically represented using RDF and serialized in OWL file formats such as OWL/XML, RDF/XML, Turtle, or JSON-LD (JSON for Linking Data). OWL ontologies can be processed by various tools and reasoners that support OWL semantics, enabling applications such as data integration, knowledge sharing, semantic search, and intelligent reasoning systems.

Chapter 2

Review of Past Work and Problem Formulation

Ontologies have received significant attention and research focus in the field of natural disasters. Researchers have extensively studied the development and application of ontologies in various domains related to natural disasters, such as hydrology or wildfire management. These ontologies aim to define and represent the concepts, relationships, and properties within these specific domains, enabling a standardized and structured representation of knowledge.

One aspect of ontology development in the realm of natural disasters involves creating domain-specific ontologies. These ontologies are designed to capture the key concepts, entities, and relationships relevant to a particular domain. For example, for floods, an ontology may define concepts such as rivers, rain, floods, water flow, etc., and their relationships, allowing for a comprehensive representation of hydrological processes. Similarly, in wildfire detection and management, an ontology may define concepts like fire risk, sensors, fire alert, mitigation, etc. In addition to domain-specific ontologies, generalized ontologies have also been developed. These ontologies aim to provide a more generic representation of concepts and relationships that can be applied across multiple domains. Reasoning rules are incorporated into these ontologies to enable automated inference and deduction of new knowledge from the existing data.

The development and application of ontologies, along with reasoning rules, have proven valuable in the field of natural disasters. They provide a standardized and structured representation of knowledge, enabling interoperability and knowledge sharing across different systems and organizations. These ontologies and rules help in capturing and integrating domain-specific knowledge, facilitating a holistic understanding of natural disasters and supporting various tasks such as risk assessment, emergency response planning, and mitigation strategies during natural disasters.

2.1 Ontologies Related to the Forest Fire Domain

There are quite a few ontological approaches related to the forest fire domain. Masa et al. [5] present the ONTO-SAFE framework which attempts to enhance forest fire detection accuracy and also provides a forest fire Decision Support System to sustain in the wildfire hazard context. Based on the SSN vocabulary [6], SoKNOS ontology [7] and also beAWARE ontology [8], ONTO-SAFE is a lightweight framework for gathering and connecting heterogeneous data received from a variety of resources such as environmental, sensors, social media, input from first responders. It uses SHACL-compliant rules [9] as the reasoning scheme. Chandra et al. [10] developed rules for calculating fire weather indices like FFMC, DMC, DC ISI, BUI and FWI [11], which are used to measure fire danger. Using SWRL (Semantic Web Rule Language) [12], the rules have been designed to forecast the severity of fire with respect to the prevailing weather conditions. SWRL is a rule language to define rules in OWL (Web Ontology Language). Kalabokidis et al. [13] presented OntoFire, in which they attempted to extract meaningful information in geo-portal environments by employing hyperlinks rather than browsing with keyword-based queries and for that reason, they had to maintain a meta-data catalogue. OntoFire offers ontology-based and spatially based navigation algorithms, which take advantage of the semantic and spatial relationships between the resources. In our ontology, we extended the base SSN ontology to our domain by including the GeoSPARQL ontology for location, and created the Settlement Ontology and the class for different temperature and gas sensors. For reasoning, we used the W3C recommendation. The ontologies SoKNOS and beAWARE are more general and less geared at forest fires. Our methodology is to get the information from sensors like smoke sensors and carbon dioxide level sensors and then propose rules for forest fire detection at a location and management after detection. OntoFire ontology was not preferred because we will have to maintain a metadata catalogue about the wildfire resources in the area of our interest, which is a complex and a tedious work.

2.2 Generic Ontologies for Natural Disaster Management

Wang et al. [14] proposed a hydrological sensor web ontology based on W3C SSN ontology by including the W3C Time Ontology [15] and OGC GeoSPARQL [16] and established the rules for the reasoning status of the hydrological event (flood) by determining the water and precipitation levels. The BeAWARE [8] ontology is a lightweight knowledge representation of concepts relevant to the management of climate-related crises. The ontology incorporates diverse data generated from a crisis representation, data from sensor analyses, and social media inputs. The ONTOEMERGE (2010-2013), an ontology developed by UFRJ (Universidade Federal do Rio de Janeiro) along with the University of Valencia, has the purpose of supporting variability solutions for emergency plans [17]. It contains some generic potential concepts like climatic conditions, incidents, emergencies, organisation, resources, and events, among others. The EmergencyFire ontology [18] enables standardization and sharing of response protocols for fire in buildings. It facilitates a) sharing and integration of information, b) providing interoperability between people and systems, c) reducing occurrences of false compliances, and d) improve response time in emergencies. BeAWARE, ONTOEMERGE and EmergencyFire ontologies are too generic and contain generalised concepts related to natural disasters. Our proposed ontology is more lightweight and specific to forest fire detection and management. It incorporates logical rules to notify nearby fire stations in case of fire detection, as well as the ability to locate hospitals within a specified distance through semantic querying, among other management features. This ensures that our ontological model not only detects forest fire but also facilitates efficient management afterward.

2.3 Work related to Competency Questions

Competency Questions (CQs) play a vital role in ontology evaluation. The efficiency of an ontology depends on the answerability of the ontology to the CQs. The QuestionChecker module, which considers CQs expressed as interrogative phrases that work over classes and their relations, is presented by Bezerra et al. along with a discussion of its significance [19]. References [20], [21], [22] have also suggested utilizing CQs.

Chapter 3

Proposed Ontology

The occurrence of forest fires entails two crucial aspects: timely forest alerts and effective communication of information pertaining to the location and impact. Unfortunately, there are anomalies caused in detection or communication due to defects in sensors or lack of proper information interoperability among the sensors deployed in forests. The heterogeneity in data and technologies causes these difficulties in interoperability among the sensor nodes. With the application of ontologies in communication, interoperability issues are minimised to a great extent, thereby improving the communication among the sensors. Ontologies are designed or extended from existing ontologies as per the requirements in a specific domain. Ontologies aid in the conversion of data into a fundamental RDF graph, which is a technology-agnostic standard format. In our case, we need an ontology that can be applied to temperature, humidity and gas sensors in Wireless Sensor Nodes. These nodes are responsible for monitoring the forest and continuously storing the corresponding environmental data.

We propose a novel FFO, a lightweight ontology that can be used to interpret sensor data and is intended to standardize the concepts and relationships among the concepts involved in forest fire detection and also provide efficient steps for managing the wildfire. Our primary research questions include displaying readings in a specific time period, identifying sensor locations, the location of the detected fire, giving information about the population of a particular settlement near the fire location, finding hospitals and fire stations nearby, and more. Overall, the objective is to eradicate the information interoperability issues, to respond to the detected anomalies in sensors' readings promptly while fire is detected, and to properly manage the wildfire if detected.

The proposed FFO ontology is designed by extending the standard W3C SSN ontology [6] and OGC GeoSPARQL [16]. The main ontological components involved are the Sensor Ontology (extension of SSN ontology) in which concepts or classes related to sensors and their observations are defined, the Settlement ontology defining the concepts related to settlements, the GeoSPARQL ontology having the concepts related to the location of a point or an area of interest like location of the point where the fire is detected or location of a hospital nearby, etc.

Table. 3.1 Prefixes and Namespaces used in FFO

PREFIX	NAMESPACE URI	DESCRIPTION
sosa	http://www.w3.org/ns/sosa/	The Sosa Ontology, also known as the Sensor,Observation, Sample, and Actuator Ontology, forms the basis of SSN ontology.
ssn	http://www.w3.org/ns/ssn/	The Semantic Sensor Network (SSN) ontology is an ontology for describing actuators and sensors, as well as their observations, related processes, interesting topics for research, samples utilized in that research, and observed attributes.
geosparql	http://www.opengis.net/ont/geosparql#	The Open Geospatial Consortium (OGC) has developed GeoSPARQL, a standard, for representing and querying geospatial linked data for the Semantic Web.
geof	http://www.opengis.net/def/function/geosparql/	A collection of GeoSPARQL-compatible, domain specific spatial filter functions for use in SPARQL queries.
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	An information representation system for the Web is called Resource Description Framework (RDF). In RDF graphs, which are collections of *subject-predicate-object*triples, IRIs, blank nodes, and datatyped literals can all be used as elements. They are used to give descriptions of resources.
rdfs	http://www.w3.org/2000/01/rdf-schema#	For RDF data, RDF Schema offers a vocabulary for data modelling. An expansion of the fundamental RDF vocabulary is RDF Schema.
xsd	http://www.w3.org/2001/XMLSchema#	The structure of an XML document is described by an XML Schema. XML Schema Definition (XSD) is another name for the XML Schema language.
swrlb	http://www.w3.org/2003/11/swrlb	The logic operation formulae for boolean operations, string operations, mathematical computations, etc. are included in built-ins, which are modular SWRL components.
	http://www.semanticweb.org/hp/ontologies/2023/1/ffo	The proposed ontology. No prefix is used for the ontology.

The namespaces and the prefixes used in the ontology are listed in Table 3.1. The proposed ontology was constructed with five main objectives:

1. to define the main concepts and properties (relationships) between the concepts in the Forest Fire Detection and Management domain.
2. to link the ontologies involved like the sensor ontology, the settlement ontology and GeoSPARQL.
3. to overall monitor efficiently and enable fast response by improving the semantic interoperability among the sensor nodes.
4. to infer new knowledge from the existing data stored and enhance the reasoning process by establishing inference rules and query processing.
5. to perform some emergency management tasks like alerting the authorities of fire stations nearby about the detected fire, finding hospitals nearby, etc.

3.1 Semantic Sensor Network Ontology

The Semantic Sensor Network or SSN ontology, developed by the Semantic Sensor Networks Incubator Group under the W3C [23], provides a semantic framework to represent and analyze sensor networks and their surroundings. It allows developers to model different sensor systems and devices, capturing the stimuli generated by sensors in response to environmental changes. By observing the properties of these sensors, meaningful results can be derived.

We have used the SSN ontology in our model and extended to the forest fire detection and management domain. Using OWL constructs, SSN models the concepts, relationships, attributes, data types, and constraints for forest fire detection and management. The class *Observation* (figure 3.1) establishes the link between a sensor and its output, allowing for the identification of significant environmental features. Additionally, the SSN incorporates sensing methods to describe real-time events, such as sensor positioning or usage. It aligns its foundational concepts and relationships (features, observations, characteristics, systems, and sensors) with the DOLCE-Ultra Lite (DUL) ontology [24].

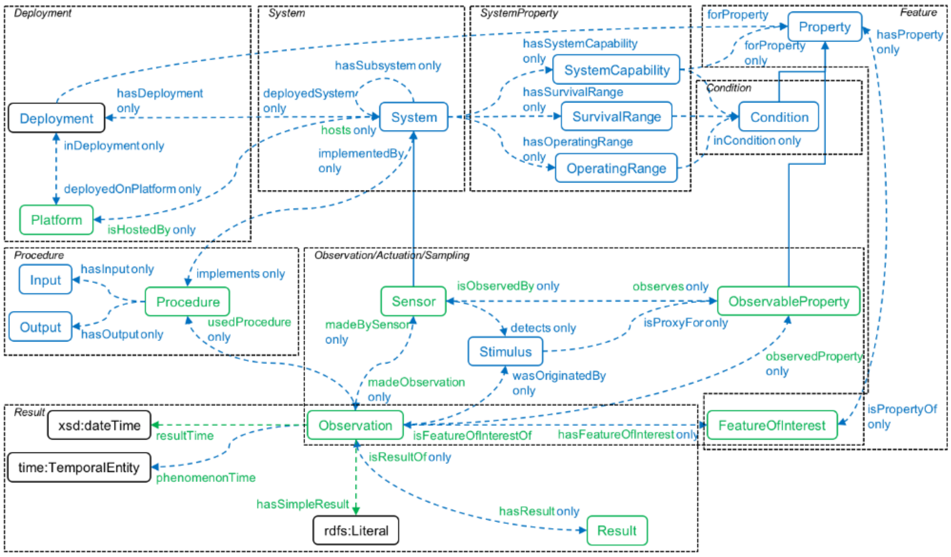

Fig. 3.1 SSN classes and properties (observation perspective) [23].

There are three perspectives of the SSN classes and properties viz. the observation perspective, the actuation perspective and the sampling perspective. We consider the observation perspective (as shown in figure 3.1) because we are working with sensors deployed in the forest and their observations. The SSN is a domain-independent model that has to be extended with specialized concepts and instances [14].

In our work, we extended the SSN ontology by introducing new classes for various gas sensors and temperature-humidity sensors required for forest-fire detection, classes to indicate the fire risk levels, as well as by adding new properties to link the classes. Additionally, we instantiated these classes based on the specifications of our experiments conducted in the IIT Indore forest. The extended parts are shown in figure 3.3.

3.2 GeoSPARQL Vocabulary

GeoSPARQL, developed by Open Geospatial Consortium [25], is a standardized query language and ontology for representing and querying geospatial data in the Semantic Web context [16]. It extends the SPARQL query language, which is used for querying RDF (Resource Description Framework) data, to include spatial and geometric concepts.

The GeoSPARQL ontology defines a set of spatial concepts and relationships that allow for the representation of geospatial data. It includes classes for representing spatial objects such as points, lines, and polygons, as well as properties to describe their spatial relationships, such as containment, intersection, and distance.

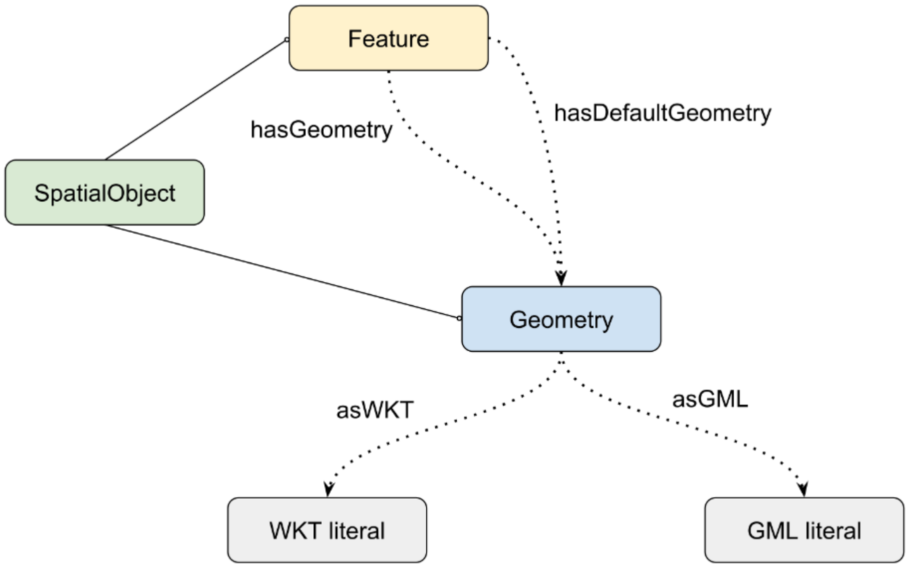

Fig. 3.2 GeoSPARQL classes Feature and Geometry, as well as some of their properties [26].

As per the GeoSPARQL standard, any resource possessing a geographical location or occupying a geographical area is classified as a *Feature*. The coordinates of a location are represented by the *Geometry* class, which offers two serialization options: *WKT literal* or *GML literal*. The Geometry class further comprises two subclasses: *Point* and *Polygon*. A specific point of location is represented using the *Point* subclass, while an extended space or area is represented using the *Polygon* subclass. The relationship between the *Feature* and *Geometry* classes is established through the *hasGeometry* property.

Figure 3.2 shows some of the GeoSPARQL classes and the properties. *Feature, Geometry, SpatialObject, WKT literal* and *GML literal* are the classes. *hasGeometry, hasDefaultGeomtry, asWKT* and *asGML* are the properties. *hasGeometry* connects the classes *Feature* and *Geometry*. For instance, in the proposed FFO ontology, we have *Hospital* as a *Feature* and *Hospital* has a pair of location coordinates as *Geometry*. In this way, other properties connect the different classes in the GeoSPARQL vocabulary.

The FFO ontology is an extension of the GeoSPARQL vocabulary. It introduces additional subclasses under the class Feature, namely *Deployment, Forest, FireStation, Hospital,* and *Settlement*. These subclasses were created to enable the inclusion of fire location information when detected. The fire is identified through the sensors deployed within a deployment, necessitating the inclusion of deployment locations. Furthermore, the locations of hospitals and nearby fire stations are also needed. This information is utilized to calculate distances between various entities, such as determining the nearest hospital or fire station following the detection of a fire.

3.3 The Framework of the Proposed Ontology

Simplicity was the key principle while constructing our ontology and we wanted the ontology to be forest fire detection and management specific. Therefore, we covered all of the aspects for the forest fire detection and management system with the minimum number of classes and properties. FFO is designed using the Protégé software [27]. The core classes and properties are shown in Figure 3.3. As we can see in the figure, the properties are defined to connect a subject to an object in an RDF triple structure. There are two types of properties - 1) Object properties and 2) Data Properties. The object properties are those in which the object is a class whereas data properties are the properties in which the object is a literal or value. In figure 3.3, we have shown the main object properties.

In figure 3.3, the classes from the SSN ontology are outlined in blue colour, those from the GeoSPARQL ontology are outlined in green colour and the classes from our proposed ontology are outlined in orange colour. *prefix*:*name* notation is also used to denote the source ontology and class or property name. The proposed ontology, FFO, has no prefix. Therefore, classes and properties from FFO are denoted as *:name*.

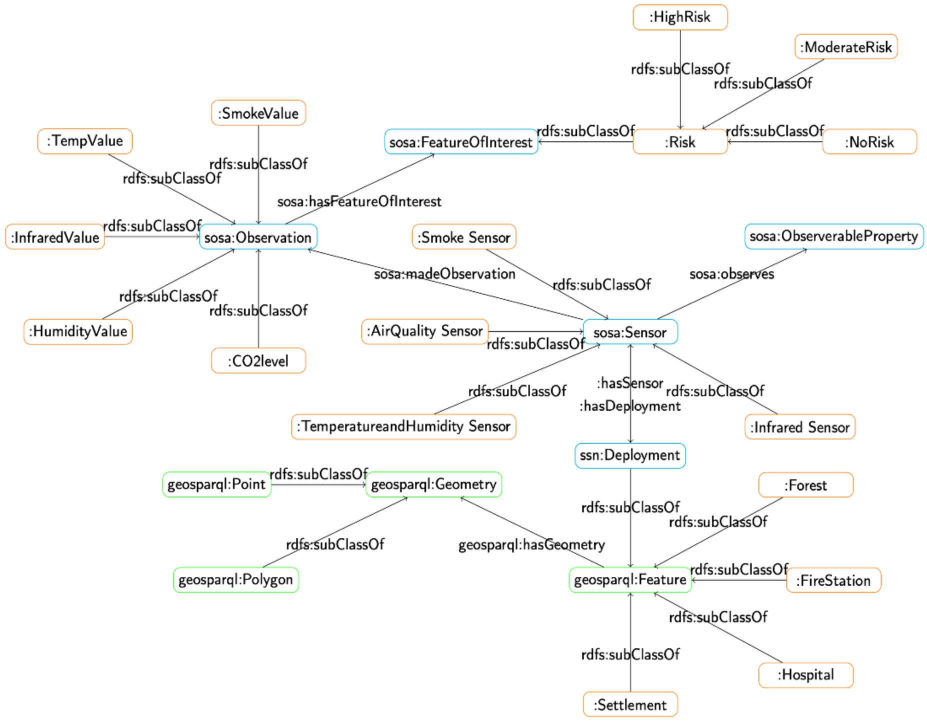

Fig. 3.3 The core classes and properties in FFO based on SSN and GeoSPARQL.

Below we discuss the main classes and properties in the proposed ontology. *prefix*:*class_name* or *prefix*:*property_name* notation is used to introduce them. No prefix is used for our ontology.

3.3.1 Classes

- *sosa:Sensor*: The class, *Sensor*, is taken from SOSA ontology (Table 1). It represents all the sensors that we deployed and has four subclasses in our ontology: 1) *TemperatureandHumiditySensor* (to measure temperature and humidity), 2) *SmokeSensor* (to detect smoke), 3) *AirQualitySensor* (to get the carbon dioxide level), and 4) *InfraredSensor* (to detect movement). DHT11 has been used as the temperature and humidity sensor. For the smoke sensor, we have used the MQ2 gas sensor. MQ135 is used as the air quality gas sensor to measure the level of carbon dioxide (CO2) in the atmosphere. Finally, the IR sensor as an infrared sensor. Altogether, there are 20 sensors (five of each category) under the class *Sensor*. These 20 sensors are the individuals or instances of the class.

- *sosa:Observation*: Class *Observation* has five sub-classes in our ontology - *TempValue*, *HumidityValue*, *SmokeValue*, *InfraredValue*, and *CO2level*.
- *geosparql:Feature*: Every entity is a *Feature* if it has a geographical location or area. We have created many subclasses in Feature, viz. *Deployment*, *Forest*, *FireStation*, *Hospital*, and *Settlement*.
- *ssn:Deployment*: There are five deployments that have been deployed in the forest of IIT Indore. Each deployment has a set of four sensors DHT11, MQ2, MQ135 and IR sensor. The location of the detected fire can be traced from the location of a Deployment whose sensors detect the fire. The location has been recorded by GPS (Global Positioning System) sensor.
- *ssn:Deployment*: There are five deployments that have been deployed in the forest of IIT Indore. Each deployment has a set of four sensors DHT11, MQ2, MQ135 and IR sensor. The location of the detected fire can be traced from the location of a Deployment whose sensors detect the fire. The location has been recorded by GPS (Global Positioning System) sensor.
- *geosparql:Geometry:* In [25], the OGC GeoSPARQL class, *Geometry,* is described as a coherent collection of direct positions in space. A spatial reference system (SRS) is used to hold the positions. It has two subclasses: *Point* for one single location of interest, and *Polygon* for an area of interest. These define the coordinates of a location.

The class hierarchy is shown in figure 3.4.

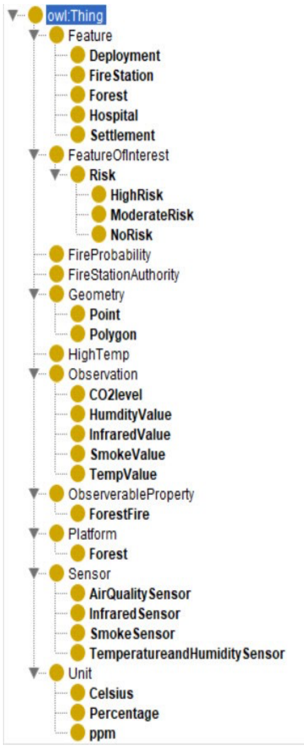

Fig. 3.4 Class hierarchy of FFO as displayed in Protégé.

3.3.2 Properties

In this section, the main properties (object properties and data properties) are discussed and also, $Subject \rightarrow Object$ notation is used to denote the domain and range of each property. The object properties are shown in subfigure (a) and the data properties of the proposed ontology are shown in subfigure (b) of figure 3.5.

- *ssn:deployedOnPlatform*: It is an object property showing the relation between *Deployment* and*Platform*. In our case, *Forest* is a *Platform*. For example, *Deployment-deployedOnPlatform-Forest*.

  Deployment → Platform

- *:hasCO2Level*: An object property created by us to show the relation between *AirQualitySensor* and *CO2level*.

  AirQualitySensor → CO2level

- :*hasDeployment:* showing on which deployment a sensor is placed.

Sensor → *Deployment*

- *:hasSensor*: Inverse property of the property *hasDeployment*. It shows which sensor a deployment has.

  Deployment → *Sensor*

- *geosparql:hasGeometry*: The object property, *hasGeometry* from GeoSPARQL defines the spatial representation of a *Feature* (class from GeoSPARQL). It forms the link between GeoSPARQL's *Feature* and *Geometry*, which is basically the coordinates of a location.For example, *Deployment-hasGeometry-LatitudeLongitude*.

  Feature → *Geometry*

- *:hasLocation*: We created this object property to directly connect a *Feature* with a location (geosparql: Point) or an area (geosparql:Polygon).

  Feature → *Point*

  Feature → *Polygon*

- *sosa:madeObservation*: Showing the relation between *Sensor* and *Observation*.

  Sensor → *Observation*

- *geosparql:asWKT*: This a data property from GeoSPARQL which links class *Geometry* with the datatype, wktLiteral, from GeoSPARQL. The datatype geosparql:wktLiteral is used to contain the Well- Known Text (WKT) serialization of a Geometry (24).

  Geometry → *wktLiteral*

- *sosa:hasSimpleResult*: It links *HumidityValue* or *TempValue* or *CO2level* or *SmokeValue* to xsd:float datatype from XSD (table 1) to get the output value of each observation from a sensor.

$HumidityValue \rightarrow float$

$TempValue \rightarrow float$

$SmokeValue \rightarrow float$

$CO2level \rightarrow float$

- *:hasTimestamp*: It links *HumidityValue* or *TempValue* or *CO2level* or *SmokeValue* to xsd:dateTime to get the timestamp of observation.

$TempValue \rightarrow dateTime$

$HumidityValue \rightarrow dateTime$

$SmokeValue \rightarrow dateTime$

$InfraredValue \rightarrow dateTime$

$CO2level \rightarrow dateTime$

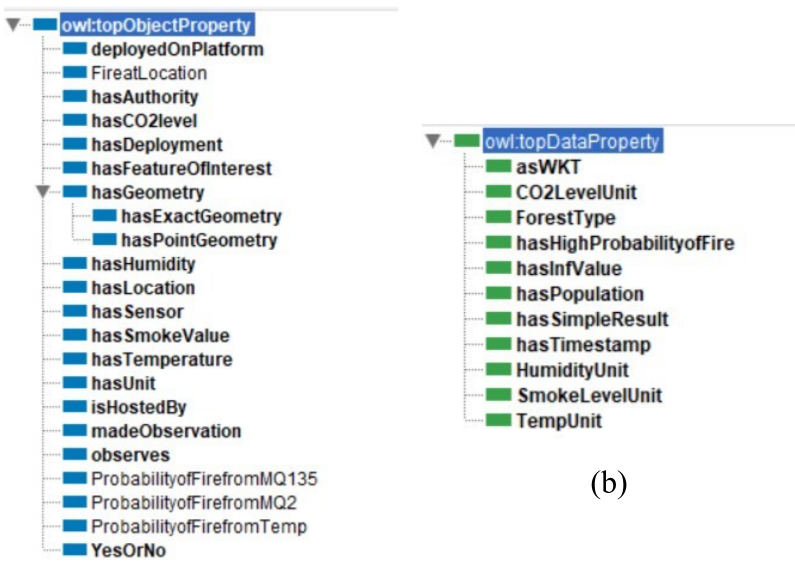

Fig. 3.5 Properties in the proposed ontology (as displayed in Protégé): (a) Object properties, (b) Data properties.

- *:hasPopulation*: It links *Settlement* and xsd:integer. It defines the population of a settlement. This is important to know to get an idea of the impact that will be caused by the fire.

Settlement → *integer*

3.3.3 Individuals

Individuals are the concrete entities or instances that exist within a domain and are represented within the ontology. A class may or may not have individual(s). In our ontology, *Deployment1*, *Deployment2*, *Deployment3*, *Deployment4*, and *Deployment5* are the five instances of the class Deployment as we have five systems deployed at five locations in the IIT Indore forest. In each deployment, there are four sensors. For example, *Deployment1* has *DHT_1*, *MQ135_1*, *MQ2_1* and *IR_1* sensors. *DHT_1*, *DHT_2*, *DHT_3*, *DHT_4,* and *DHT_5* as the instances of the DHT sensors. Every deployment has a location that is of class *Point*. For example, *D1PointGeom* is the location of *Deployment1*. Some of the individuals are shown in figure 3.6.

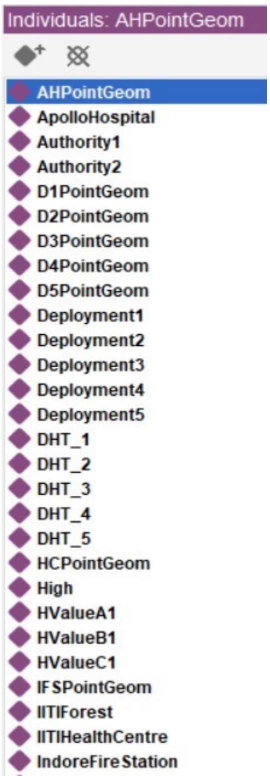

Fig. 3.6 An illustrative depiction of select individuals of the proposed ontology.

Chapter 4

Competency Questions

Competency questions (CQs) play a crucial role in ontology development as they help to identify the requirements and scope of an ontology [19]. Competency questions are essentially queries or questions that users or applications may ask about a particular domain or subject matter. They provide a way to elicit and specify the knowledge that should be captured in the ontology. Competency questions serve several purposes. These purposes are as follows:

- **Scope Determination**: Competency questions help define the boundaries and scope of an ontology by identifying the types of information and relationships that need to be represented.

- **Requirement analysis**: By formulating CQs, the information needs and requirements of users or applications can be identified, ensuring that the ontology captures the relevant knowledge.

- **Evaluation and testing**: CQs can be used to evaluate the effectiveness of an ontology by checking if the questions can be answered using the ontology's concepts, relationships, and axioms (rules).

- **Communication**: CQs provide a means to communicate and discuss the goals and objectives of an ontology with stakeholders, domain experts, and developers.

When formulating competency questions, it is important to consider various aspects, such as the target audience, the intended use of the ontology, and the specific domain or subject area. Competency questions should be clear, and precise, and cover a wide range of relevant aspects within the domain.

There are eight CQs on the basis of which our FFO ontology is evaluated. These are as follows.

1. **Show the values of sensors from time t1 to time t2.**

   The purpose of this competency question (CQ) is to display the readings of all sensors or specific sensors within a defined time frame. This information is necessary for users who wish to verify whether a fire was detected during a specific period. The sensors continuously collect data from the surrounding environment. However, when dealing with a substantial volume of data, it becomes challenging to identify instances where a value exceeds a predefined threshold. To simplify this task, we can examine the values within a specified time period, thus facilitating the process.

2. **Find the location of the sensor which recorded values greater than the threshold value.**

   The purpose of this CQ is to determine the specific location of a sensor that has recorded readings surpassing a predefined threshold. This addresses the need to identify the exact sensor responsible for detecting values that exceed a particular threshold. By finding the location of such a sensor, users or applications can gain insight into the specific area or region where the threshold was surpassed. Surpassing a threshold value indicate (mostly) a hazardous or critical situation, which is a forest fire in our case. By finding the location of the sensor, emergency response teams or safety personnel can quickly pinpoint the affected area and take the appropriate measures to mitigate risks or provide assistance.

3. **What are the hospitals that are nearby with respect to location?**

This CQ is for identifying the hospitals that are in close proximity to a specific location. When fire spreads, it causes a great impact on people or natural habitats along with the environment. In such cases of a medical emergency or urgent healthcare needs, knowing the hospitals that are closest to a particular location (location of the probable fire) can help in making informed decisions about where to seek medical assistance promptly. It enables individuals or emergency services to quickly identify and reach the nearest healthcare facility.

4. **Which are the nearest fire stations?**

The purpose of this CQ is to identify the fire stations that are located closest to a specific point or area (the point at which fire has been detected, to be appropriate). By determining the nearest fire stations, emergency response planners can assess the availability and proximity of fire stations to different areas or communities. This information is valuable for designing efficient emergency response strategies and ensuring the timely deployment of firefighting resources to address fire incidents effectively. Identifying the nearest fire stations enables fire departments and emergency management agencies to optimize resource utilization by strategically assigning fire stations based on proximity, ensuring efficient coverage and response capabilities.

5. **State whether there is a settlement located near a detected fire location. If yes, what is the distance of the settlement from the fire location and what is its population?**

This CQ is to determine if a settlement is present in close proximity to a detected fire location. If such a settlement exists, the competency question aims to provide information about the distance between the settlement and the fire location, as well as details regarding the population of the settlement. Its objective is to understand the potential impact of a fire incident on nearby settlements and gather relevant information for emergency response and planning. Determining if a settlement is located near a detected fire location helps in evaluating the potential risks associated with the fire incident. It enables emergency response teams and authorities to understand the immediate impact on the settlement and allocate appropriate resources accordingly. Furthermore, obtaining information about the population of the settlement near the fire location allows for an assessment of potential vulnerability and the need for additional support or assistance. It assists in determining the resources required to address the needs of the settlement's population during and after the fire incident.

6. **If there is a probability of fire, state which sensor sensed the probable fire and what is the risk level?**

This is to know which sensor sensed or detected a probable fire event and to provide information about the associated risk level according to the readings of the sensor. Knowing the sensor which detected fire enables monitoring and maintenance of the sensors. It allows for focused attention on specific sensors, ensuring their proper functioning, calibration, and maintenance to enhance the reliability and accuracy of fire detection systems. Providing information about the risk level associated with the probable fire event helps in risk assessment and prioritizing response efforts. By understanding the risk level, emergency response teams can allocate appropriate resources and prioritize their actions based on the severity and potential impact of the fire event. The risk level is determined by the predefined rules. According to the rules we have designed for reasoning, there are three risk levels – High, Medium and No risk, based on the threshold values of the sensor which detected the probable fire.

7. **Whether there is a high risk of fire, a medium risk, or no risk at all?**

The objective of this competency question is to provide a quick and concise assessment of the fire risk level. The competency question helps in identifying whether a high, moderate or no risk of fire exists. This classification allows for a preliminary understanding of the severity and potential consequences of a fire event or fire-prone condition. The determined risk level assists in resource allocation and response planning. It provides valuable information for allocating firefighting resources, personnel, and equipment based on the assessed risk level. Response plans can be tailored accordingly to address the identified risk level effectively.

8. **Notify the fire station authorities about the location where there is a high probability of fire.**

The objective of this competency question is to facilitate timely communication and alert the fire station authorities to potential fire-prone areas or situations. By notifying the fire station authorities about locations with a high probability of fire, this CQ enables early response and intervention. It ensures that firefighting resources and personnel can be dispatched to the identified locations promptly, minimizing response time and enhancing the chances of containing and extinguishing fires before they escalate. As a future measure, authorities can implement targeted fire prevention strategies, conduct inspections, enforce safety regulations, and educate residents or occupants in the identified areas, reducing the likelihood of fires and promoting fire-safe practices.

Chapter 5

Real-time Data Collection

5.1 Sensors used in the experiments

We carried out experiments in the forests of the Indian Institute of Technology Indore to collect real-time data. This data is stored using FFO ontology and the proposed ontology is evaluated based on this data. We have selected four types of sensors for the experiment process. DHT11 temperature-humidity sensor, MQ2 gas sensor, MQ135 gas sensor and IR sensor. A system showing these sensors is shown in figure 5.1. The selection of appropriate sensors plays a critical role in accurately detecting and monitoring forest fires. The following sensors were chosen for their specific capabilities and suitability for the task:

- DHT11 Sensor:

  The DHT11 sensor was employed to measure temperature and humidity levels in the forest environment. Temperature fluctuations and high humidity can provide important contextual information for assessing fire risks and potential fire propagation. The DHT11 sensor allowed for real-time monitoring of these environmental factors.

- MQ2 Gas Sensor:

  The MQ2 sensor was utilized for detecting smoke and flammable gases, which are indicative of fire presence. This sensor responds to the presence of combustible gases such as methane, propane, and butane, as well as smoke particles. By monitoring the concentration of these gases, the MQ2 sensor provided valuable insights into the presence and intensity of a potential fire.

- MQ135 Gas Sensor:

The MQ135 sensor was employed to measure air quality and detect the presence of hazardous gases. Forest fires can release various pollutants into the air, including carbon dioxide (CO_2). The MQ135 sensor allowed for the monitoring of the gases, enabling the assessment of air quality in the forest.

- IR Sensor:

  The IR (Infrared) sensor detects movement if any. Therefore, it was used to detect if there is any human activity or an animal at the location of the probable fire.

During the experimentation phase, these sensors were strategically placed in different locations within the forest area to capture environmental data and detect potential fire events. The collected sensor readings were then processed and analyzed to identify patterns, anomalies, and potential fire risks.

It is important to note that the choice of sensors may vary depending on specific requirements, environmental conditions, and available resources. The mentioned sensors were selected based on their capabilities and suitability for forest fire detection in the context of this particular experiment.

5.2 Deployment of Sensors

A Deployment has the four above-mentioned sensors and there are five such deployments deployed at five locations in the forest of IIT Indore. Therefore, there are altogether 20 sensors deployed. For example, Deployment1 has sensors DHT_1, MQ2_1, MQ135_1 and IR_1. The five deployments are Deployment1, Deployment2, Deployment3, Deployment4 and Deployment5. A deployment is shown in figure 5.1.

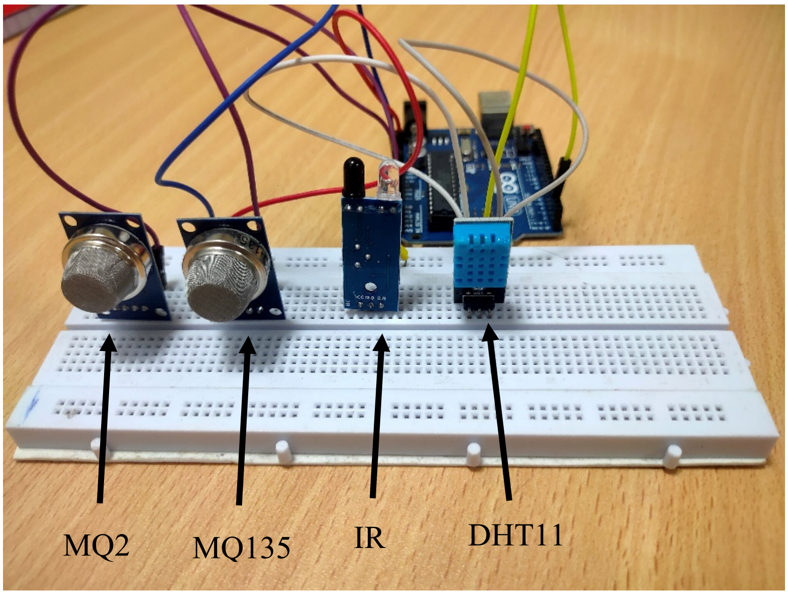

Fig. 5.1 A deployment showing the four sensors.

5.3 Experimental Data

We have collected real-time data by experimenting in the IIT Indore forest. We placed one deployment in each of the five locations and initiated a controlled fire in the forest area near Deployment 3 and collected the readings for different sensors. Figure 5.2 shows the recorded readings and figure 5.3 shows the placement of deployment 3 in the forest and the initiated fire.

MQ2 smoke sensor exhibited 0.15 ppm (parts per million) when no smoke is detected and a value range from 1200 ppm to 400,000 ppm when smoke is detected. MQ135 (CO2 level) gas sensor recorded values in a range of 0.3 - 138 ppm. However, the DHT11 temperature and humidity sensor showed a gradual increase in temperature and humidity. The temperature range was recorded to be from 43.1 °C to 59.2 °C in the presence of fire whereas the normal temperature at the time of the experiment was 38 °C.

We have defined certain thresholds for the DHT11 temperature value, MQ2 smoke level, and MQ135 ppm value for the “Fire Alert” based on the findings of our experiment.

Time	Date	Temperature	Humidity	LPG (ppm)	CO (ppm)	Smoke (ppm)	CO2 (ppm)	Obstacle
14:25:00	6/4/2023	nan	nan	0.02	0.05	0.15	nan	0
14:25:03	6/4/2023	43.9	15	0.02	0.05	0.15	0.3	0
14:25:05	6/4/2023	43.7	15	0.02	0.05	0.15	1.79	0
14:25:07	6/4/2023	43.7	15	0.02	0.05	0.15	6.51	0
14:25:09	6/4/2023	43.8	15	0.02	0.05	0.15	13.92	0
14:25:11	6/4/2023	43.9	15	3875.44	409769.5	28855.07	23.68	0
14:25:14	6/4/2023	43.9	15	6226.54	785008.06	44262.81	27.77	0
14:25:17	6/4/2023	43.9	15	6749.25	802429.75	44262.81	26.71	0
14:25:20	6/4/2023	43.9	15	6537.94	785008.06	44262.81	24.66	0
14:25:22	6/4/2023	43.7	15	6028.84	640666.12	37176.72	21.84	0
14:25:25	6/4/2023	43.9	15	4533.26	496380.62	29388.66	18.35	0
14:25:28	6/4/2023	43.8	15	3610.07	345407	22146.59	16.04	0
14:25:31	6/4/2023	43.8	15	3545	336830.62	23470.85	16.04	0
14:25:34	6/4/2023	43.8	15	3610.07	345407	23470.85	15.31	0
14:25:36	6/4/2023	43.8	15	3298.44	320399.59	23019.25	14.61	0
14:25:39	6/4/2023	43.7	15	3610.07	345407	23470.85	15.33	0
14:25:42	6/4/2023	43.8	15	3610.07	345407	23470.85	15.31	0
14:25:45	6/4/2023	43.9	15	3610.07	345407	23019.2	14.59	0
14:25:48	6/4/2023	44	16	3238	282454.4	19322.78	13.16	0
14:25:50	6/4/2023	44.1	16	2741.02	236045.78	17489.21	12.52	0
14:25:53	6/4/2023	44.2	16	2741.02	236045.78	17489.21	11.89	0
14:25:56	6/4/2023	44.3	16	2741.02	229882.84	17135.29	11.88	0
14:25:59	6/4/2023	44.4	16	2492.73	207010.12	15802.32	11.28	0
14:26:02	6/4/2023	44.6	16	2492.73	207010.12	15802.32	11.26	0
14:26:04	6/4/2023	44.7	16	2492.73	196177.81	15476.92	10.68	0
14:26:07	6/4/2023	44.8	16	2175.3	171481.28	12831	10.12	0
14:26:10	6/4/2023	44.8	16	2051.11	158098.06	12831	9.59	0
14:26:13	6/4/2023	45.1	16	1855.6	137653.39	11528.96	9.56	0

Fig. 5.2 The date and time and the values from the experiment recorded by Deployment 3. The sudden increase in the values implies detected fire.

Fig. 5.3 Deployment 3 in the forest of IIT Indore, taking readings in the presence of fire.

In the case of temperature,

- If the value is greater than 45 °C, then the risk level for DHT11 is HIGH.

- If the value is less than or equal to 45 °C and greater than 30 °C, then the risk level is MODERATE.
- If the value is less than or equal to 30 °C, then there is NO RISK.

In the case of smoke level,

- If the smoke level is more than or equal to 30,000 ppm (parts per million), then the risk level for MQ2 gas sensor is HIGH.
- If the smoke level is less than 30,000 ppm and greater than 1200 ppm, then that is a MODERATE RISK.
- If it has less than or equal to 0.15 ppm, then the risk status is NO RISK.

In the case of carbon dioxide levels,

- If it has greater than or equal to 20 ppm, then it's HIGH RISK for MQ135 sensor.
- If there is less than 20 ppm and greater than 5 ppm, then its a MODERATE RISK.
- If it has less than or equal to 5 ppm, then it is NO RISK.

It should be noted that the observed values from the experiment, as presented in figure 5.2, are not universal and may vary from sensor to sensor. The value range of gas sensors depends on several factors. These include the type of sensor used, the specific target gas being detected, the calibration process employed, the sensitivity of the sensor, potential interference from other gases, and the operating conditions in which the sensor is deployed. Different gas sensor technologies have varying characteristics, resulting in different value ranges. Calibration is essential to establish a correlation between the sensor's output and the actual gas concentration. Factors such as sensor sensitivity and environmental conditions can also influence the value range. Considering these factors is crucial for the accurate interpretation of gas sensor readings and obtaining meaningful gas concentration measurements.

Chapter 6

Results and Discussion/Analysis

A series of semantic querying and reasoning was developed to assess the proposed FFO ontology. For reasoning, we devised 14 SWRL (Semantic Web Rule Language) rules using Protégé software, and for semantic querying, we used the RDF query language, SPARQL [2] which we have implemented in GraphDB. GraphDB [26] is a semantic graph database management system that specializes in storing, querying, and managing large-scale semantic data. It is developed by Ontotext, a company focused on semantic technology solutions. GraphDB leverages the Resource Description Framework (RDF) data model and supports the SPARQL query language for working with semantic data.

The SWRL rules for reasoning and queries for query-result processes were designed on the basis of competency questions discussed in Chapter 4. We have used the Pellet Reasoner [28] for semantic reasoning. Pellet Reasoner is an open-source software library that provides powerful reasoning capabilities for ontologies represented in the Web Ontology Language (OWL). It is developed as part of the Protégé project and is widely used in various applications and research projects. The reasoner deduces new inferences based on the existing data or knowledge and the rules defined.

6.1 Query Processing

Query processing is a fundamental component that enhances the value and utility of an ontology. It enables users to retrieve specific information from the ontology based on their needs, facilitating efficient access to knowledge. Through queries, users can explore the ontology, uncovering relationships and gaining insights into the data. Query processing also leverages reasoning capabilities to infer new knowledge and expand the scope of information that can be queried. Additionally, queries can be used to validate and test the ontology's correctness and completeness, identifying any inconsistencies or errors. Ultimately, query processing empowers users to make informed decisions, build intelligent applications, and integrate ontology data into various systems, maximizing the ontology's impact and utility.

Queries are processed in GraphDB. SPARQL is used as the query language. We test the functionality of an ontology using query processing to check its capability to answer as expected to the competency questions. Some of the queries are discussed below.

- **Query 1**: Query to show the temperature values of DHT sensors from all the deployments within the specified time period. This query corresponds to competency question no 1. Figure 6.1 shows the query and the results.

```
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX ssn:<http://www.w3.org/ns/ssn#>
PREFIX : <http://www.semanticweb.org/hp/ontologies/2023/1/ffo#>
PREFIX sosa: <http://www.w3.org/ns/sosa#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select ?datetime ?temperature ?sensor ?deployment
where {
        ?sensor :hasDeployment ?deployment .
        ?sensor :hasTemperature ?t .
        ?t :hasTimestamp ?datetime .
        ?t sosa:hasSimpleResult ?temperature .
        Filter(?datetime >= "2023-04-06T14:27:00"^^xsd:dateTime && ?datetime <= "2023-
04-06T14:28:00"^^xsd:dateTime) .
}
```

(a)

	datetime	temperature	sensor	deployment
1	"2023-04-06T14:27:00"^^xsd:dateTime	"41.8"^^xsd:float	:DHT_2	:Deployment2
2	"2023-04-06T14:27:02"^^xsd:dateTime	"40.0"^^xsd:float	:DHT_1	:Deployment1
3	"2023-04-06T14:27:03"^^xsd:dateTime	"39.0"^^xsd:float	:DHT_2	:Deployment2
4	"2023-04-06T14:27:03"^^xsd:dateTime	"47.6"^^xsd:float	:DHT_3	:Deployment3
5	"2023-04-06T14:27:09"^^xsd:dateTime	"39.6"^^xsd:float	:DHT_4	:Deployment4

(b)

Fig. 6.1 Frame view of GraphDB showing (a) SPARQL query to display the temperature values in DHT sensors of all the deployments in the specified time period (Query 1), (b) showing initial five results to the query.

- **Query 2**: Query to display all the smoke values which are greater than 1200 ppm (the threshold for the smoke sensor), the smoke sensors which detected those values, their corresponding deployments, and the time at which those values were recorded. This is related to competency question 2.This query is given below and figure 6.2 shows the result of query 2.

```
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX geof:<http://www.opengis.net/def/function/geosparql/>
PREFIX ssn:<http://www.w3.org/ns/ssn#>
PREFIX :
<http://www.semanticweb.org/hp/ontologies/2023/1/ffo#>
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX sosa: <http://www.w3.org/ns/sosa#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>


select ?datetime ?deployment ?MQ2 ?smoke value
where {
?MQ2 :hasDeployment ?deployment .
?MQ2 :hasSmokeValue ?v .
?v sosa:hasSimpleResult ?smoke value .
?v :hasTimestamp ?datetime .
Filter( ?smoke value >= 1200) .}
```

	datetime	deployment	MQ2	smoke_value
1	"2023-04-06T14:26:38"^^xsd:dateTime	:Deployment3	:MQ2_3	"2371.4"^^xsd:float
2	"2023-04-06T14:26:09"^^xsd:dateTime	:Deployment3	:MQ2_3	"2655.9"^^xsd:float
3	"2023-04-06T14:26:32"^^xsd:dateTime	:Deployment3	:MQ2_3	"3048.34"^^xsd:float
4	"2023-04-06T14:26:41"^^xsd:dateTime	:Deployment3	:MQ2_3	"3393.82"^^xsd:float
5	"2023-04-06T14:26:11"^^xsd:dateTime	:Deployment3	:MQ2_3	"4406.0"^^xsd:float

Fig. 6.2 Figure showing initial five results of query 2.

- **Query 3**: Query to display all the hospitals that are within the range of 20 km from the MQ2 sensor which detected smoke levels greater than 1200 ppm. This is related to competency question 3. Figure 6.3 shows the result of query 3.

```
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX geof:<http://www.opengis.net/def/function/geosparql/>
PREFIX ssn:<http://www.w3.org/ns/ssn#>
PREFIX :
<http://www.semanticweb.org/hp/ontologies/2023/1/ffol#>
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX sosa: <http://www.w3.org/ns/sosa#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select ?deployment ?sensor ?smoke value ?Hospital ?distance
where
{
?deployment a ssn:Deployment .
?deployment :hasSensor ?sensor .
?sensor :hasSmokeValue ?tvalue.
?tvalue sosa:hasSimpleResult ?smoke value .
:Deployment3 :hasLocation ?d1 .
?d1 geosparql:asWKT ?l1 .
?Hospital a :Hospital .
?Hospital :hasLocation ?h .
?h geosparql:asWKT ?l2 .
BIND(geof:distance(?l1,?l2) as ?distance).
FILTER(?smoke value >1200 && ?distance <= 20000 ).
}
```

	deployment	sensor	smoke_value	Hospital	distance
1	:Deployment3	:MQ2_3	"2371.4"^^xsd:float	:IITIHealthCentre	"432.5770665987057"^^xsd:double
2	:Deployment3	:MQ2_3	"2371.4"^^xsd:float	:MhowRailwayHospital	"16568.84776703363"^^xsd:double

Fig. 6.3 Result of query 3.

- **Query 4**:Query to display all the fire stations that are within the range of 50 km from the DHT sensor which recorded temperatures greater than 45 °C. Thisis related to competency question 4. Figure 6.4shows the result to query 4.

```
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX geof:<http://www.opengis.net/def/function/geosparql/>
PREFIX ssn:<http://www.w3.org/ns/ssn#>
PREFIX :
<http://www.semanticweb.org/hp/ontologies/2023/1/ffo#>
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX sosa: <http://www.w3.org/ns/sosa#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select ?deployment ?sensor ?temperature ?FireStation ?distance
where
{
?deployment a ssn:Deployment .
?deployment :hasSensor ?sensor .
?sensor :hasTemperature ?tvalue.
?tvalue sosa:hasSimpleResult ?temperature .
:Deployment3 :hasLocation ?d1 .
?d1 geosparql:asWKT ?l1 .
?FireStation a :FireStation .
?FireStation :hasLocation ?h .
?h geosparql:asWKT ?l2 .
BIND(geof:distance(?l1,?l2) as ?distance) .
FILTER(?temperature >45 && ?distance <= 50000).
}
```

	deployment	sensor	temperature	FireStation	distance
1	:Deployment3	:DHT_3	"50.5"^^xsd:float	:IndoreFireStation	"21822.099123090455"^^xsd:double
2	:Deployment3	:DHT_3	"50.5"^^xsd:float	:MhowFireStation	"17222.380695005133"^^xsd:double

Fig. 6.4 Result of query 4.

- **Query 5**: Query to display the settlements that are within the range of 50 km from the MQ2 smoke sensor which recorded smoke level values greater than 1200 ppm and also display the population of the settlements. This is related to competency question 5. Figure 6.5shows the result to query 5.

```
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX geof:<http://www.opengis.net/def/function/geosparql/>
PREFIX ssn:<http://www.w3.org/ns/ssn#>
PREFIX :
<http://www.semanticweb.org/hp/ontologies/2023/1/ffol#>
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX sosa: <http://www.w3.org/ns/sosa#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select ?deployment ?sensor ?smoke value ?settlement ?population
?distance where {
?deployment a ssn:Deployment .
?deployment :hasSensor ?sensor .
?sensor :hasSmokeValue ?tvalue.
?tvalue sosa:hasSimpleResult ?smoke value .
:Deployment3 :hasLocation ?d1 .
?d1 geosparql:asWKT ?l1 .
?settlement a :Settlement .
?settlement :hasLocation ?h .
?settlement :hasPopulation ?population .
?h geosparql:asWKT ?l2 .
BIND(geof:distance(?l1,?l2) as ?distance) .
FILTER(?smoke value >1200 && ?distance <= 50000 ) .
}
```

	deployment	sensor	smoke_value	settlement	population	distance
1	:Deployment3	:MQ2_3	"2371.4"^^xsd:float	:Settlement1	"9856"^^xsd:integer	"1911.368351081626"^^xsd:double
2	:Deployment3	:MQ2_3	"2371.4"^^xsd:float	:Settlement2	"11000"^^xsd:decimal	"348.52852346706015"^^xsd:double

Fig. 6.5 Result of query 5.

The results were as we had anticipated.

6.2 Rule-based Reasoning

Based on the competency questions mentioned above, we set up rules with the purpose of making inferences and acquiring new knowledge based on the existing classes and relationships, to check whether our ontology can answer the competency questions. This is also a process of evaluating the ontology in addition to query processing.

6.2.1 Rules for Reasoning

To create the rules, we used the SWRL language [12]. The rules were implemented in Protégé software. Protégé has a Plugin for SWRL, named *SWRLTab*. We have created 14 Rules for reasoning. These are as follows.

1. *swrlb:greaterThanOrEqual* (*?t*, 45) ∧*hasDeployment*(*?s, ?d*) ∧*hasLocation*(*?d, ?p*) ∧*geosparql:asWKT*(*?p, ?loc*) ∧*hasTemperature*(*?s, ?temp*) ∧*sosa:hasSimpleResult*(*?temp, ?t*)→ *ProbabilityofFirefromTemp(?p, High)*

   - If the temperature value measured by a DHT sensor exceeds 45 °C, the Deployment's location associated with that specific sensor is categorized as High Risk.

2. *swrlb:lessThanOrEqual* (*?t*, 45) ∧*swrlb:greaterThan* (*?t*, 30) ∧*hasDeployment*(*?s, ?d*) ∧*hasLocation*(*?d, ?p*) ∧*geosparql:asWKT*(*?p, ?loc*) ∧*hasTemperature*(*?s, ?temp*) ∧*sosa:hasSimpleResult*(*?temp, ?t*)→ *ProbabilityofFirefromTemp(?p, Moderate)*

   - If the temperature value measured by a DHT sensor is less than or equal to 45 °C but greater than 30 °C, the Deployment's location associated with that specific sensor is categorized as Moderate Risk.

3. *swrlb*:*lessThanOrEqual* (?*t,* 30) ∧*hasDeployment*(?*s,* ?*d*) ∧*hasLocation*(?*d,* ?*p*) ∧*geosparql*:*asWKT*(?*p,* ?*loc*) ∧*hasTemperature*(?*s,* ?*temp*) ∧*sosa*:*hasSimpleResult*(?*temp,* ?*t*)→ *ProbabilityofFirefromTemp(?p,No)*

   - If the temperature value measured by a DHT sensor is less than or equal to 30 °C, the Deployment's location associated with that specific sensor is categorized as No Risk.

4. *hasSmokeValue(?s, ?sv)*∧ *hasDeployment(?s, ?d)* ∧ *sosa*: *hasSimpleResult(?sv, ?ss)* ∧ *hasLocation(?d, ?p)* ∧ *geosparql: asWKT (?p, ?loc)* ∧ *swrlb:greaterThanOrEqual(?ss, 30000)* → *ProbabilityofFirefromMQ2(?p, High)*

   - If the smoke level measured by an MQ2 sensor exceeds or is equal to 30000 ppm, the Deployment's location associated with that specific sensor is categorized as High Risk.

5. *hasSmokeValue(?s, ?sv)*∧ *hasDeployment(?s, ?d)* ∧ *sosa*: *hasSimpleResult(?sv, ?ss)* ∧ *hasLocation(?d, ?p)* ∧ *geosparql: asWKT (?p, ?loc)* ∧ *swrlb:lessThan (?ss, 30000)*∧ *swrlb:greaterThan(?ss, 1200)* → *ProbabilityofFirefromMQ2(?p, Moderate)*

   - If the smoke level measured by an MQ2 sensor is less than 30000 ppm but greater than 1200 ppm, the Deployment's location associated with that specific sensor is categorized as Moderate Risk.

6. *hasSmokeValue(?s, ?sv)*∧ *hasDeployment(?s, ?d)* ∧ *sosa*: *hasSimpleResult(?sv, ?ss)* ∧ *hasLocation(?d, ?p)* ∧ *geosparql: asWKT (?p, ?loc)* ∧ *swrlb:lessThanOrEqual(?ss, 0.15)* → *ProbabilityofFirefromMQ2(?p, No)*

   - If the smoke level measured by an MQ2 sensor is less or equal to 0.15 ppm, the Deployment's location associated with that specific sensor is categorized as No Risk.

7. *swrlb: greaterThanOrEqual(?v, 20)∧ hasDeployment(?s, ?d) ∧ sosa: hasSimpleResult(?c, ?v) ∧ hasLocation(?d, ?p) ∧ geosparql: asWKT(?p, ?loc) ∧ hasCO2level(?s, ?c) → ProbabilityofFirefromMQ135(?p, High)*

   - If the carbon dioxide(CO2) level measured by an MQ135 gas sensor exceeds or is equal to 20 ppm, the Deployment's location associated with that specific sensor is categorized as High Risk.

8. *swrlb: lessThan(?v, 20)∧greaterThan(?v, 5)∧ hasDeployment(?s, ?d) ∧ sosa: hasSimpleResult(?c, ?v) ∧ hasLocation(?d, ?p) ∧ geosparql: asWKT(?p, ?loc) ∧ hasCO2level(?s, ?c) → ProbabilityofFirefromMQ135(?p, Moderate)*

   - If the CO2 level measured by an MQ135 sensor is less than 20 ppm but greater than 5 ppm, the Deployment's location associated with that specific sensor is categorized as Moderate Risk.

9. *swrlb: lessThanOrEqual(?v, 5)∧ hasDeployment(?s, ?d) ∧ sosa: hasSimpleResult(?c, ?v) ∧ hasLocation(?d, ?p) ∧ geosparql: asWKT(?p, ?loc) ∧ hasCO2level(?s, ?c) → ProbabilityofFirefromMQ135(?p, No)*

   - If the CO2 level measured by an MQ135 sensor is less or equal to 5 ppm, the Deployment's location associated with that specific sensor is categorized as No Risk.

10. *ProbabilityofFirefromTemp(?p, High)∧ ProbabilityofFirefromMQ2(?p, High)∧ ProbabilityofFirefrmoMQ135(?p, High)→ HighRisk(?p)*

    - If a location is categorized as High Risk by the DHT11 sensor, MQ2, and MQ135 sensors, then that location is on High Alert.

11. *ProbabilityofFirefromTemp(?p, Moderate)∧ ProbabilityofFirefromMQ2(?p, Moderate)∧ ProbabilityofFirefrmoMQ135(?p, Moderate)→ ModerateRisk(?p)*

    - If a location is categorized as Moderate Risk by the DHT11 sensor, MQ2, and MQ135 sensors, then that location is on Moderate Alert.

12. *ProbabilityofFirefromTemp(?p,No)∧ ProbabilityofFirefromMQ2(?p, No)∧ ProbabilityofFirefrmoMQ135(?p, No)→ NoRisk(?p)*

    - If a location is categorized as No Risk by the DHT11 sensor, MQ2, and MQ135 sensors, then that location is on No Alert.

13. *HighRisk(?p) ∧ hasAuthority(?f, ?a)→ FireatLocation(?a, ?p)*

    - Notify the authorities of all the fire stations about the location which is on High Alert.

14. *ModerateRisk(?p) ∧ hasAuthority(?f, ?a)→ FireatLocation(?a, ?p)*

    - Notify the authorities of all the fire stations about the location which is on Moderate Alert.

We have defined some threshold values in the rules for the reasoner to draw inferences from the sensors' readings. These threshold values are defined by analyzing the nature of readings (figure 5.2) recorded by the sensors deployed in the presence of fire.

6.2.2 Semantic Reasoning

Now that we have designed the rules, the reasoner is allowed to deduce the inferences or new knowledge based on the existing knowledge (classes and relationships) and the predefined rules. The reasoner used in our case is the Pellet reasoner [28]. There were three distinct inferences in our ontology when the reasoner was started. These are discussed below.

- **Reasoning1:** DHT_3 of Deployment3 recorded 57.4 °C which is greater than 45 °C. According to Rule 1 of the rules discussed above, if the temperature is greater than 45 °C, then the object property called *ProbabilityofFirefromTemp* connects the "location of the Deployment which has the DHT11 sensor" to *High*. In our case, it connects *D3PointGeom,* which is the location of Deployment 3 to the Risk level, *High*. This is Inference 3 in figure 6.6. Similarly, inferences 1 and 2 are drawn from Rules 7 and 4 mentioned in section 6.2.1. This is corresponding to competency question 6.

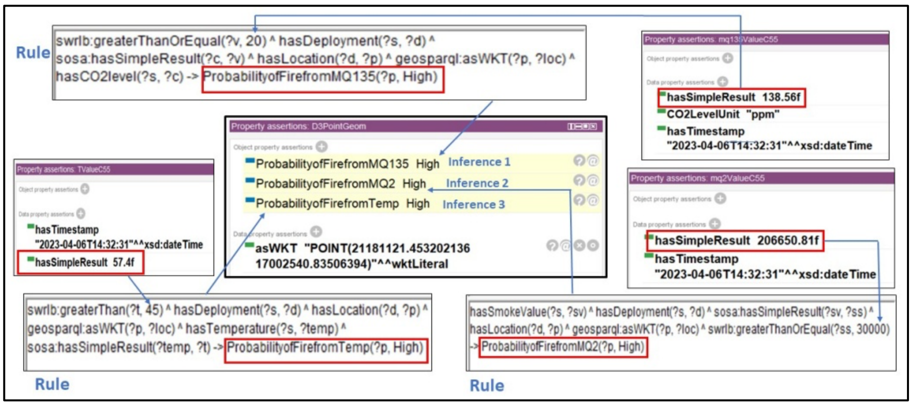

Fig. 6.6 Protégé frame view showing inferences deduced by the reasoner with respect to the experiment data values of Deployment 3 and the rules 1,4,7 given in section 6.2.1.

- **Reasoning 2:** If the probability of Fire from DHT11, MQ2, and MQ135 sensors are high, then the location of the corresponding deployment is marked as *HighRisk*. Rule 10 states this deduction. Figure 6.7 shows how the reasoner marks *D3PointGeom* as *HighRisk* as it deduced the inferences discussed in the Reasoning 1 section. This inference is corresponding to competency question 7.

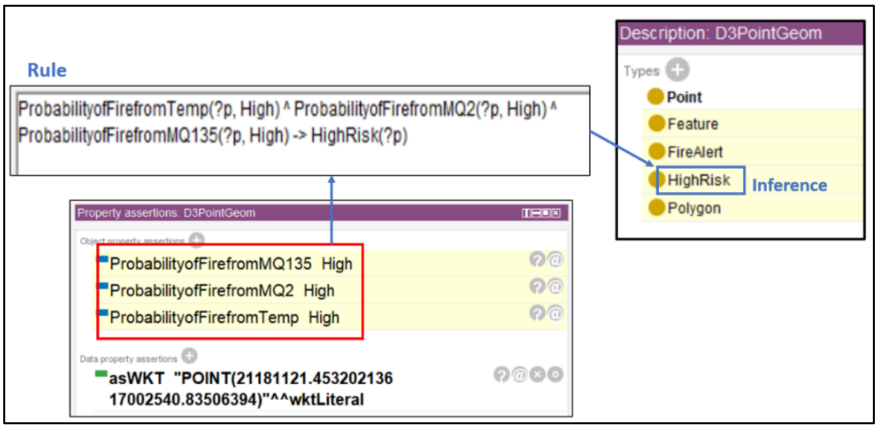

Fig. 6.7 Reasoning 2 which infers that D3PointGeom is at *HighRisk*.

- **Reasoning 3:** If a location is marked as *HighRisk,* then the authorities of all the fire stations should be notified about that location. Rule 13 is designed for this deduction. Figure 6.8 shows the two fire stations in our ontology which are *IndoreFireStation* and *MhowFireStation*, and *Authority1* and *Authority2* are their authorities respectively. The object properties *FireatLocation* connects the following.

Authority1 → D3PointGeom

Authority2 → D3PointGeom

These are inferences 1 and 2 respectively in figure 6.8. This inference is corresponding to competency question 8.

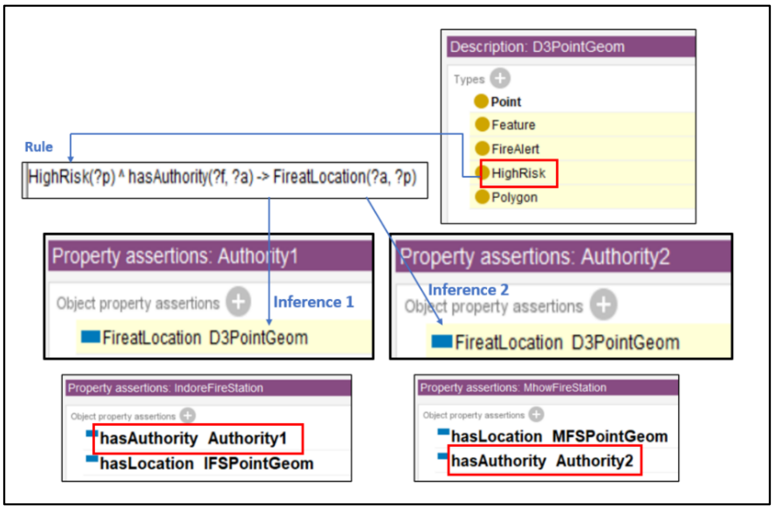

Fig. 6.8 Protégé frame view showing the authorities of the corresponding fire stations and these authorities are being notified about D3PointGeom,the location of Deployment 3 through inferences 1 and 2.

Therefore, we have fulfilled the requirements of all the eight competency questions discussed in chapter 4 with the help of Query Processing and Semantic Reasoning, thereby evaluating the practical feasibility of our FFO ontology.

Chapter 7

Conclusions and Scope for Future Work

We have presented an ontology-based model for Forest Fire Detection and Management that addresses various aspects and objectives in the domain. The purpose of our model encompasses multiple dimensions.

Firstly, our model aims to effectively represent the main concepts and properties of the Forest Fire domain. By capturing the essential elements of forest fire detection and management, we provide a comprehensive ontology that serves as a knowledge representation framework. This allows us to organize and structure the domain-specific information, ensuring its availability and accessibility for further analysis and decision-making.

Additionally, we have instantiated or created individuals of the concepts designed according to the requirements of our experiment. We recognize the importance of tailoring the ontology to specific use cases and application scenarios. By customizing the concepts and properties, we ensure that the ontology aligns with the specific needs and objectives of our experiment. This customization enhances the ontology's relevance and applicability in real-world forest fire detection and management scenarios.

A key objective of our model is to standardize the data created by sensors deployed in the forest. The deployment of sensors plays a vital role in collecting real-time data related to temperature, humidity, gas levels, and other relevant parameters. By incorporating these sensor-generated data into the ontology, we promote efficiency in data sharing and reusing among the sensors. This interoperability allows for seamless integration and collaboration between different sensors, enabling a comprehensive understanding of the forest fire dynamics.

Information interoperability is a critical aspect addressed by our ontology-based model. By representing the data in a standardized format and leveraging semantic technologies, we enhance the compatibility and exchangeability of information. This interoperability empowers the different sensors using different technologies deployed in forests as well as different stakeholders, such as fire station authorities, emergency responders, and forest management agencies, to access and share information seamlessly. Consequently, decision-making processes are streamlined, leading to more effective and timely actions in response to forest fire incidents.

Another significant aspect of our model is the detection of Risk Level or Fire Probability. Through the integration of data from various sensors, the ontology enables the evaluation and determination of the risk associated with different forest locations. By leveraging semantic reasoning, we can infer and assess the likelihood of fire outbreaks in specific areas. This proactive detection allows for timely interventions and preventive measures, minimizing the potential damage caused by forest fires.

The practical feasibility of our ontology is rigorously tested through query processing and semantic reasoning against competency questions. Competency questions serve as benchmarks to evaluate the ontology's ability to provide accurate and meaningful answers to specific inquiries. By formulating semantic queries and employing semantic reasoning techniques, we analyze the real-time data collected through our experiments. The design of FFO ontology ensures that the data is structured and organized in RDF triple format, facilitating efficient query execution and result retrieval. This validation process showcases the ontology's effectiveness in addressing the targeted competency questions, further solidifying its practical applicability.

While our current ontology prototype serves as a foundation and is a lightweight ontological prototype in the forest fire domain, we acknowledge the future scope for enhancements and extensions. As we gain more insights from our experiments and gather feedback from stakeholders, we plan to refine and expand the ontology. One area of future development involves the addition of more rules for extensive reasoning capabilities. By incorporating additional rules and inference mechanisms, we can enable more sophisticated analyses and decision-making processes.

These enhancements will augment the ontology's management steps, facilitating a more comprehensive and proactive approach to forest fire detection and management.

In conclusion, our ontology-based model, FFO: A Forest-Fire Ontology, for Forest Fire Detection and Management presents a holistic framework that addresses various aspects of the domain. By representing the main concepts and properties, standardizing sensor-generated data, promoting information interoperability, and enabling proactive detection and management, our model contributes to the effective handling of forest fire incidents, thereby contributing to the safety of wildlife, humans, and the environment. Through rigorous validation against competency questions, we ensure the practical feasibility and utility of the ontology. As we continue to enhance and extend the ontology based on evolving requirements, we strive to create a robust and comprehensive solution for forest fire detection and management.

References

[1] V. Nunavath, A. Prinz, Data sources handling for emergency management: Supporting information availability and accessibility for emergency responders, 2017, pp. 240-259. doi:10.1007/978-3-319-58524-6_21.

[2] A. Seaborne, E. Prud'hommeaux, Sparql query language for rdf, W3C recommendation, World Wide Web Consortium (W3C) (2013). URL https://www.w3.org/tR/rdf-sparql-query/

[3] D. Brickley, L.Miller, FOAF Vocabulary Specification (2004). URL http://xmlns.com/foaf/0.1/

[4] T. R. Gruber, Toward principles for the design of ontologies used for knowledge sharing?, Int. J. Hum. Comput. Stud. 43 (1995) 907-928.

[5] P. Masa, G. Meditskos, S. Kintzios, S. Vrochidis, I. Kompatsiaris, Ontology-based modelling and reasoning for forest fire emergencies in resilient societies, in: Proceedings of the 12th Hellenic Conference on Artificial Intelligence, SETN '22, Association for Computing Machinery,New York, NY, USA, 2022. doi:10.1145/3549737.3549765. URL https://doi.org/10.1145/3549737.3549765

[6]M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K.Janowicz, W. D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus,A. Nikolov, K. Page, A. Passant, A. Sheth, K. Taylor, The ssn ontology of the w3c semantic sensor network incubator group, Journal of Web Semantics 17 (2012) 25-32. doi:https://doi.org/10.1016/j.websem.2012.05.003.

[7] G. Babitski, S. Bergweiler, O. Grebner, D. Oberle, H. Paulheim, F. Probst, Soknos – using semantic technologies in disaster Management software, in: G. Antoniou, M. Grebelnik, E. Simperl, B. Parsia, D. Plexousakis, P. De Leenheer, J. Pan (Eds.), The Semantic Web: Research and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 183-197.

[8] E. Kontopoulos, P. Mitzias, J. Mobgrader, P. Hertweck, H. van der Schaaf, D. Hilbring, F. Lombardo, D. Norbiato, M. Ferri, A. Karakostas, S. Vrochidis, I. Kompatsiaris, Ontology-based representation of crisis management procedures for climate events, 2018.

[9] H. Knublauch, D. Kontokostas, Shapes constraint language (shacl), W3.org https://www.w3.org/TR/shacl/, [Accessed: May 21, 2023](2017).

[10] R. Chandra, K. Abhishek, S. Agarwal, N. Singh, Semantic sensor network ontology decision system for forest fire management, Ecological Informatics 72 (2022) 101821. doi:https://doi.org/10.1016/j.ecoinf.2022.101821

[11] National Wildfire Coordinating Group, Fire weather index system, https://www.nwcg.gov/publications/pms437/cffdrs/fire-weather-index-system, [accessed: May 22, 2023] (2016).

[12] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean,Swrl: A semantic web rule language, W3.org https://www.w3.org/Submission/SWRL/, [Accessed: May 21, 2023] (2004).

[13] K. Kalabokidis, A. Nikos, M. Vaitis, Ontofire: An ontology-based geo-portal for wildfires, Natural Hazards and Earth System Sciences 11 (2011) 3157-3170. doi:10.5194/nhess-11-3157-2011

[14] Wang, N. Chen, W. Wang, Z. Chen, A hydrological sensor web ontology based on the ssn ontology: A case study for a flood, ISPRS International Journal of Geo-Information 7 (1) (2018). doi:10.3390/ ijgi7010002. URL https://www.mdpi.com/2220-9964/7/1/2

[15] S. Cox, C. Little, Time ontology in owl, W3.org https://www.w3.org/ TR/shacl/, [Accessed: May 21, 2023] (2022).

[16] M. Perry, J. Herring, Geosparql – a geographic query language for rdf data, OGC.org https://www.ogc.org/standard/geosparql/, [Accessed: May 21, 2023] (2012).

[17] M. Bosca, K. Cordeiro, J. Oliveira, M. Campos,Ontoemerge: Construção de uma ontologia core para a área deemergências

[18] K. BITENCOURT, F. Durão, M. Mendonça, L. Santana, An onto-logical model for fire emergency situations (2018), pp. 108-115. doi:10.1587/ transinf.2017SWP0003.

[19] C. Bezerra, F. Freitas, F. Santana, Evaluating ontologies with competency questions, in: 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Vol. 3, 2013, pp. 284-285. doi:10.1109/WI-IAT.2013.199.

[20]M. Uschold, M. Gruninger, Ontologies: principles, methods and applications, The KnowledgeEngineering Review 11 (2) (1996) 93–136. doi:10.1017/S026988890000779.

[21] M. Fernández-López, A. Gómez-Pérez, N. Juristo, Methontology: From ontological art towards ontological engineering, in: AAAI Conference on Artificial Intelligence, 1997.

[22] P. Haase, H. Lewen, R. Studer, D. Tran, M. Erdmann, M. d'Aquin, E. Motta, The neon ontology engineering toolkit, in: The Web Conference, 2008.

[23] A. Haller, K. Janowicz, S. Cox, D. Phuoc, K. Taylor, M. Lefrançois, Semantic Sensor Network Ontology, W3.org https://www.w3.org/TR/vocab-ssn/, [Accessed: June 01, 2023] (2017).

[24] V.Presutti,A.Gangemi (2008), Content Ontology Design Patterns as Practical Building Blocks for Web Ontologies. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds) Conceptual Modeling - ER 2008. ER 2008. Lecture Notes in Computer Science, vol 5231. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-87877-3_11

[25] Open Geospatial Consortium, GeoSPARQL - A Geographic Query Language for RDF Data, https://opengeospatial.github.io/ogc-geosparql/geosparql11/index.html, [accessed: May 30, 2023] (2012).

[26] Ontotext AD, GraphDB: A semantic graph database, https:// graphdb.ontotext.com/, [Online; accessed:May 22, 2023] (2021).

[27] Stanford Center for Biomedical Informatics Research, Protege, https: //protege.stanford.edu/(accessed 2023).

[28] W3C, Pellet - semantic web standards, https://www.w3.org/2001/sw/ wiki/Pellet, [accessed: April 3, 2023].