Inferring New Information from a Knowledge Graph in Crisis Management: A Case Study

Julie Bu Daher[1], Tom Huygue[1], Nathalie Hernandez[2] and Patricia Stolf[2]
[1]IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, Toulouse, France
[2]IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, UT2, Toulouse, France

Keywords: Semantic Web, Crisis Management, Knowledge Management, Information Inference.

Abstract: Natural crises are dangerous events that can threaten lives and lead to severe damages. Crisis-related data can be heterogeneous and be provided from multiple data sources. These data can be formally described using ontologies and then integrated and structured forming knowledge graphs. Inferring new information from knowledge graphs can strongly assist in the various phases of the crisis management process. Different approaches exist in the literature for inferring new information from knowledge graphs. In this paper, we present a case study of a flood crisis where we discuss three approaches for inferring flood-related information, and we experimentally evaluate these approaches using real flood-related data and synthetic data for further analysis. We discuss the interest of using each of these approaches and detail its advantages as well as its limitations.

1 INTRODUCTION

Natural crises, such as floods are adverse events resulting from natural processes of the Earth. They could lead to severe consequences such as loss of lives, disruption of normal life of the population and materialistic damage in properties, infrastructure and economy. From here comes the urgent need of the crisis management in order to limit these consequences. Crisis-related data can be exploited for taking important decisions that can assist in the crisis management process. These data are heterogeneous and can be provided from multiple sources. Managing these data is important for two main reasons. First, integrating and structuring the crisis-related data allows the actors involved in the management process to access the needed data at the right time. Second, structuring the data allows managing its heterogeneity and thus inferring new information that enriches the initial data shared by the actors in real-time during the crisis. This can be attained using semantic web technologies that allow the structuring of the data and the inference of new information from it.

An ontology allows a structuring and a logical representation of the knowledge through concepts and relations among concepts of an ontology. They are known for managing heterogeneity and having a consistent shared understanding of the meaning of information (Elmhadhbi et al., 2019). Heterogeneous crisis-related data can thus be integrated and structured using the concepts and relations of an ontology to form a knowledge graph. This enhances the interoperability of the data among the various actors involved in the crisis management process, and it allows inferring new information and making it explicit which helps in the decision making process of the crisis management.

Inferring new information from the crisis-related data can help in past, present and future aspects. It can help in improving a past experience through inferring new information from past crisis-related data. The situation of a current crisis can also be analyzed, and we can infer new information that helps taking current actions. In addition, we can infer new information that helps in predicting a future crisis or its phases and properties.

Crisis management has been described in the literature as a lifecycle, and it is categorized into four main phases: mitigation, preparedness, response and recovery (Franke, 2011). This work is in the frame of a project that aims at integrating several disciplinary expertises to limit the consequences of flash floods. It is a case study on real flood-related data where the aim is to propose solutions for limiting the consequences of a flood during the flood response phase. When a flood occurs, the safety of the population is the most

Bu Daher, J., Huygue, T., Hernandez, N. and Stolf, P.
Inferring New Information from a Knowledge Graph in Crisis Management: A Case Study.
DOI: 10.5220/0011511200003335
In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) - Volume 2: KEOD*, pages 43-54
ISBN: 978-989-758-614-9; ISSN: 2184-3228

important concern; therefore, an evacuation process of the population in demand points should take place where a demand point represents a place that can be impacted by the flood and thus needs to be evacuated. This process is handled by the firefighters who are responsible for taking rapid decisions and actions concerning evacuation. In our case study, we aim at proposing evacuation priorities to demand points in order to assist in the evacuation process of flood victims. In this frame, we have proposed an ontology that formally describes the flood-related data and, and we have integrated the heterogeneous data in a knowledge graph using the shared vocabulary of the ontology (Bu Daher et al., 2022). Using this knowledge graph, we aim at inferring new information representing evacuation priorities to all the demand points in our study area, and we then aim at enriching the knowledge graph with this information about priorities and updating it constantly with real-time data. The aim of this paper is to evaluate three approaches for inferring evacuation priorities of demand points through a case study on real data representing a past occurring flood.

The paper is organized as follows. Section 2 discusses the related work in this domain. Section 3 presents the problem statement containing our problematic, data description and the used ontology and knowledge graph. Section 4 presents the three approaches that are proposed as solutions for our problem which are later evaluated in section 5. Finally, section 6 discusses the conclusion and the future work.

2 RELATED WORK

Ontology-based approaches have been proposed in the literature in the domain of crisis management. The main purpose behind proposing ontologies in this domain is the information management and sharing among different actors involved in the crisis management process.

An ontology has been proposed by (Katuk et al., 2009) for integrating flood-related data to allow the coordination of response activities among different agencies involved in the management process and to provide up-to-date information that facilitates the decision making by the management committee chairman. Another ontology was proposed by (Yahya and Ramli, 2020) to formally integrate flood-related data in order to be shared by all related agencies in the management process. They propose an ontology for each agency including one describing data about evacuation centers, and they then aim at integrating all the ontologies in a global one that shares information among all agencies. An flood ontology was proposed by (Khantong et al., 2020) for managing and sharing flood information among different responders, organizers or processes that are handled by different systems in organizations in order to carry out disaster relief operations. The ontology manages static and dynamic data. Static data represent the data that don't change during a flood, while dynamic data represent the data evolving throughout a flood. Their static data are described through concepts including area and resource, and their dynamic data represent coordination and production acts concerning the crisis.

Some of the proposed ontologies include concepts related to victims' evacuation such as victim, flood as well as evacuation areas, resources and centers (Khantong et al., 2020; Yahya and Ramli, 2020); however, these concepts are not exploited for inferring new information that assist in the evacuation process.

Although not widely existing, some approaches in the literature propose inferring various kinds of information in this domain. An ontology-based framework for risk assessment is proposed by (Scheuer et al., 2013) to manage and share the knowledge of stakeholders and decision makers in risk management and to infer information from the ontology concerning elements at risk against certain event types based on the user's input. Different types of events such as floods are defined in the ontology where the user chooses an event type and defines the event intensity in order to assess elements at risk against this event. The proposed framework identifies the intensity parameters that are suitable for this flood event using the relation "IntensityOf" defined in the ontology between "Event" and "Intensity" classes. Then, the framework infers the elements at risk that are susceptible to this event through matching susceptibility functions against the chosen event type using the "isSusceptibilityTo" relation that is used to link each susceptibility function to the respective event types. A susceptibility function takes one or more intensity parameters as input and allows obtaining a damage ratio. The relevant elements at risk against the event type are then inferred by matching susceptibility functions using the "susceptibilityOf" relation defined in the ontology between "Susceptibility Function" and "Element AtRisk" classes. (Wang et al., 2018) propose a hydrological sensor web ontology, based on three existing ontologies: SOSA[1], Time[2] and GEOSPARQL[3]

[1]https://www.w3.org/TR/vocab-ssn/

[2]https://www.w3.org/TR/owl-time/

[3]https://opengeospatial.github.io/ogc-geosparql/geosparql11/index.html

to integrate heterogeneous data provided from different sensors effectively during natural disasters. They then use SWRL rules on their constructed knowledge graph to infer flood phases from the precipitation of water level and observation data. (Kurte et al., 2017) propose an ontology that captures dynamically evolving phenomena to understand the dynamic spatio-temporal behaviour of a flood disaster. ”SIIM” (Kurte et al., 2016) and ”Time” ontologies were used to describe geospatial and time concepts, and SWRL rules were then used to retrieve image regions based on their temporal interval relations. (Sun et al., 2016) propose an ontology that allows inferring flood states as well as their properties, such as precipitation and water course in the frame of a context-aware system. Jess rules (Hill, 2003) are used to infer new information that enrich their context-aware system so that its components can be adapted according to context changes.

We notice from the approaches proposed in the domain of crisis management that new information can be inferred using the relations defined among concepts of the ontology (Scheuer et al., 2013) and using rules such as SWRL rules (Wang et al., 2018; Kurte et al., 2017). SPARQL queries are used to query the knowledge graphs in order to extract information (Wang et al., 2018).

Inferring new information using defined concepts and relations' characteristics of the ontology is an approach that can be conducted using existing tools and reasoners. SPARQL query language, that is usually used for querying the knowledge graph, can be also used for inferring new information from the knowledge graph. Rules have been usually used in the domain of crisis management in order to infer new information. There exists a different kind of rules for inferring new information from knowledge graphs. In our work, we propose to evaluate three different approaches for inferring new information in the domain of crisis management.

3 PROBLEM STATEMENT

In this section, we discuss our problem statement including our problematic, the data representing our case study, our proposed ontology and knowledge graph, the evacuation priorities defined for the study area and the proposed solutions for our problematic.

3.1 Problematic

Flood crisis management is a critical process as it concerns severe consequences where the most adverse consequence is having victims in danger. Therefore, the evacuation of victims is an essential process in the flood response phase. The proposed solutions for assisting in this process should respect the delicacy of the situation in being rapid and efficient. In our work, we propose evaluating three different approaches for inferring information that can assist in the evacuation process of flood's victims. These approaches are based on approaches existing in the literature for inferring new information from knowledge graphs. The evaluated approaches propose inferring the same information differently. The first approach proposes inferring the information that enrich the knowledge graph from the concepts and relations of the ontology using OWL constructors, the second approach uses SPARQL query language to infer new information through directly enriching the knowledge graph using insert and delete queries, and the third approach proposes inferring the information using rules and enriching the knowledge graph with the inferred information. Rules are frequently used in the literature for inferring information from knowledge graphs. Different kinds of rules have been used for this purpose including SWRL rules[4]. In this approach, we propose using SHACL rules[5], a more recent kind of rules that overcomes certain limitations of other kinds and that hasn't been used in the domain of crisis management yet.

Each approach relies on the knowledge graph containing the flood-related data to infer evacuation priorities of demand points and to enrich the knowledge graph with the inferred priorities. We experimentally evaluate the performance of the three approaches on a real use case of a flood crisis in order to discuss the importance of inferring new information that helps in the crisis management process and to find the most efficient approach applicable to this use case. In addition, we extend this use case with synthetic data when needed in order to further analyze the specificities of each approach.

3.2 Data Description

The data in our study area concern the Pyrénées flood that occured in June 2013 in Bagnères-de-Luchon, south-western France. It was a torrential flood, destructive and dangerous for the population. The consequences of this flood include destructed houses, cut roads, flooded campsites and damaged farms. The food-related data are heterogeneous and are provided from various data sources. These sources in-

[4]https://www.w3.org/Submission/SWRL/
[5]https://www.w3.org/TR/shacl-af/

clude institutional databases such as BD TOPO[6] and GeoSirene[7] providing data about hazards, vulnerability, damage and resilience. Some sources provide data about geographical locations of roads, buildings, companies and establishments in France. Other data sources provide various data including data sensors providing data about water levels and flows, a hydrological model computing flood generation, a hydraulic model for flood propagation as well as other sources providing other kinds of data such as socio-economic and population data as well as danger and vulnerability indices of the flood calculated by the domain experts. The vulnerability index measures the vulnerability of a demand point, and it is calculated using topographic and social data like population density, building quality and socio-economic conditions. The danger index measures the level of danger of a demand point, and it is calculated using the water speed and level obtained from a hydraulic model. These data can be categorized as static and dynamic data. Our static data include the number of floors and geographic locations, while our dynamic data include the water level and the number of population in a demand point.

3.3 Ontology and Knowledge Graph Construction

In a previous work, we have proposed an ontology that formally describes our heterogeneous flood-related data with all the needed concepts and relations, and we constructed our knowledge graph through integrating the heterogeneous flood-related data of our study area (Bu Daher et al., 2022). Our ontology consists of 41 classes, 6 object properties and 23 data properties, and it is available online via the following URL: *https://www.irit.fr/recherches/MELODI/ontologies/i-Nondations.owl*. It is used in a further step to infer new information assisting in the flood response phase of the management process. We present as follows the concepts and relations that are strictly related to inferring new information concerning evacuation priorities.

We define in our ontology a class named "demand point" representing either an infrastructure or an infrastructure aggregation. This class is characterized by four subclasses representing evacuation priorities that are used in a further step for inferring new information concerning the priorities of demand points. The class "Material infrastructure" describes all possible types of infrastructure in the study area including homes, working places and facilities such as healthcare facilities. We also define a class named "Infrastructure aggregation" that allows managing different types of infrastructure in an aggregated manner by regrouping them in districts, buildings and floors. For example, we can describe that the district has buildings, the building has floors, and the floor has apartments using the relation "has part". This class is useful when data about different kinds of infrastructure are unavailable. For example, when data about a building are not available, we consider the data about its district.

The defined class "population"describes the population in an infrastructure including fragile and non-fragile population defined using the relations "is in" and "contains".

The object and data properties are divided into static and dynamic properties to represent static and dynamic flood-related data. The static object properties represent the relations between concepts describing static data such as "has part" describing an infrastructure or an infrastructure aggregation, while the dynamic object properties represent the relations between concepts describing dynamic data such as "contains" that describes the population in an infrastructure or an infrastructure aggregation.

Concerning the data properties, the static data properties include building's vulnerability index and number of floors, and the dynamic data properties include danger index, submersion height, flood duration, number of population and whether a demand point is inhabited or not.

Using the previously described ontology, we have constructed our knowledge graph integrating static and dynamic flood-related data. The static data were integrated only once at the beginning of the flood, transformed into RDF triples and added to the ontology. In contrary, the dynamic data were transformed into RDF triples and updated in real-time throughout the flood. The transformation of static and dynamic data into RDF triples has been performed using "rdflib" library [8] in python that maps data according to the corresponding concepts and relations of the ontology (Bu Daher et al., 2022).

3.4 Evacuation Priorities

There are four evacuation priorities defined by the domain experts in flood management for the evacuation process as follows:

"Evacuate immediately", "Evacuate in 6 hours", "Evacuate in 12 hours" and "No evacuation". Each

[6]https://www.data.gouv.fr/en/datasets/bd-topo-r/

[7]https://data.laregion.fr/explore/dataset/base-sirene-v3-ss/

[8]https://rdflib.readthedocs.io/

evacuation priority is represented as a set of conditions defined for certain properties of the demand points where the conditions of each priority are also defined by the domain experts. The properties used for defining the evacuation priorities are: danger index, duration of flood, number of floors, submersion height, vulnerability index, housing type and whether the building is inhabited or not. The conditions are defined such that the evacuation priorities are exclusive and consider all the possible values of properties describing the demand points in the study area.

3.5 Inferring New Information concerning Priorities

We recall that our aim is to manage and share heterogeneous flood-related data among different actors involved in the flood crisis management in order to help them access the needed data at the right time and take decisions with the help of the inferred information.

Based on the knowledge graph that integrates all the flood-related data, we propose inferring new information concerning the evacuation priorities of the demand points of the study area to help the firefighters take rapid decisions and actions in the evacuation process of flood victims. As previously detailed, we have four evacuation priorities that are represented as subclasses of the class ”Demand point” in the ontology. Each demand point will then be typed with one of these four evacuation priorities according to its properties. In other words, we classify the instances of demand points in our study area into four categories where each category represents one of the four evacuation priorities. The inferred information is then used to enrich the knowledge graph so that it is shared by different actors involved in the management process.

4 THREE APPROACHES FOR INFERRING EVACUATION PRIORITIES

In this section, we discuss the three approaches for inferring the evacuation priorities of the demand points in our study area.

4.1 Inferring Evacuation Priorities using FaCT++ Reasoner in Protégé

The first approach consists of enriching the knowledge graph using the semantics of the ontology. Each class representing an evacuation priority is defined with axioms that express the conditions that a demand point should satisfy, and a reasoner is used to automatically classify the demand points' instances according to the four priority classes.

Several reasoners already exist in the literature for inferring new information from knowledge graphs where some of them can be plugged in Protégé[9] including Pellet (Sirin et al., 2007), HermiT (Glimm et al., 2014) and FaCT++ (Tsarkov and Horrocks, 2006) reasoners. This allows inferring new information directly through the ontology editor and visualizing the demand points and their corresponding evacuation priorities. We choose the FaCT++ reasoner in Protégé to infer new information from our knowledge graph which represent evacuation priorities to demand points of the study area. FaCT++ is more efficient on our knowledge graph than other reasoners plugged in Protégé. For example, The information about the evacuation priorities is inferred from the knowledge graph using FaCT++ in 1.24 hours, while it takes 10 hours to be inferred using Hermit reasoner.

Figure 1 displays the four defined classes of priorities in our ontology visualized in Protégé, and figure 2 displays the axioms defined for the priority class ”Evacuate in 12h”. Similarly, axioms are defined for the corresponding classes of the three other evacuation priorities.

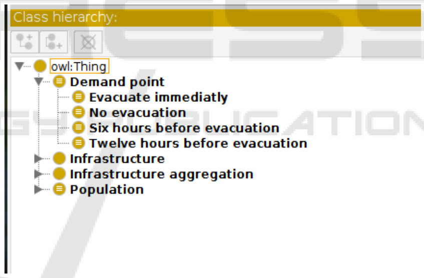

Figure 1: Classes of Evacuation priorities in Protégé.

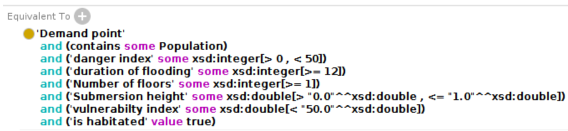

Figure 2: Evacuation priority: ”Evacuate in 12 hours”.

When loading the knowledge graph in Protégé, FaCT++ reasoning can be directly conducted to infer the evacuation priorities of demand points. The instances of demand points are thus classified among the four evacuation priorities according to their properties where each demand point can be classified and viewed under the priority class that it satisfies. Figure 3 displays an example of some demand points classified under the priority class ”Evacuate in 12h”

[9]https://protege.stanford.edu/

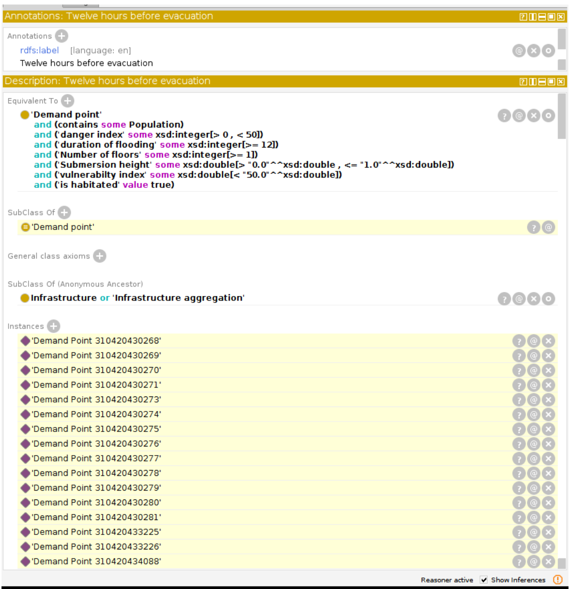

Figure 3: ”Inferences for ”Evacuate in 12h” priority in Protégé.

We then obtain the knowledge graph enriched with the inferred priorities, and this enriched knowledge graph can thus be shared with the different actors of the flood management process, particularly the firefighters concerned in the victims' evacuation in order to help them take decisions concerning the evacuation process.

4.2 Inferring Evacuation Priorities using SPARQL Queries

The second approach that we evaluate is enriching the knowledge graph with inferred information about the evacuation priorities using SPARQL queries[10]. In this approach, the knowledge graph integrating the flood-related data is first stored in a triplestore. We have chosen ”Virtuoso” triplestore for storing knowledge graphs as it is proved to be efficient in storing a big number of triples in a relatively short time[11]. For instance, the results of a benchmark show that Virtuoso loads 1 billion RDF triples in 27 minutes while it takes hours to load the same triples in other triplestores such as BigData, BigOwlim and TDB[12].

After storing the knowledge graph in Virtuoso, SPARQL insert and delete queries are implemented to classify the demand points among the four defined evacuation priorities according to their properties. The definitions of each priority class in the first approach are now expressed as conditions of each evacuation priority in its ”WHERE” condition of the SPARQL query.

We define an ”insert” statement in the SPARQL query of each evacuation priority in order to type each demand point that satisfies the conditions of the query with this evacuation priority after making sure that we have deleted all possible priorities with which this demand point could have been typed before. A new triple is thus obtained for each demand point expressing its typed priority. Figure 4 displays the SPARQL query implemented for the priority ”Evacuate in 12h”.

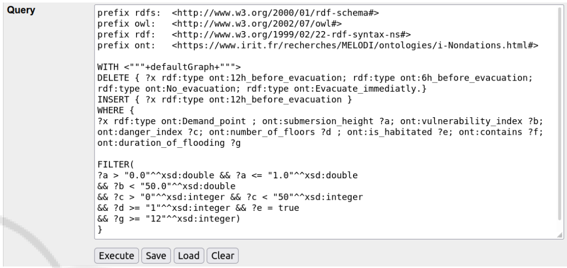

Figure 4: ”SPARQL query of the priority ”Evacuate in 12”.

The knowledge graph is directly enriched with the new triples defining the evacuation priorities of demand points on the triplestore, and it can be shared with different actors to assist in the evacuation process.

4.3 Inferring Evacuation Priorities using SHACL Rules

The third evaluated approach is inferring the evacuation priorities of demand points using SHACL rules. A Shapes Constraint Language (SHACL) rule is a recent kind of rules that can be used for inferring new information from knowledge graphs while having advantages over other kinds of rules used in the literature, and it has not been used in the domain of crisis management yet. A SHACL rule is identified through a unique Internationalized Resource Identifier (IRI) not like other kinds of rules. In addition, it can be activated or deactivated upon its usage purpose where a deactivated rule is ignored by the rules engine and is not executed. An execution order can also be determined for SHACL rules when more than one rule is implemented.

Rules are executed using TopBraid SHACL API[13] which is an open source implementation of the W3C SHACL based on Apache Jena[14].

[10]https://www.w3.org/TR/rdf-sparql-query/
[11]https://virtuoso.openlinksw.com/
[12]http://wbsg.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V7/#exploreVirtuoso
[13]https://github.com/TopQuadrant/shacl
[14]https://jena.apache.org/

There exist different types of SHACL rules including SPARQL rules, which are based on SPARQL query language that allows writing rules in SPARQL notation. In this approach, we use SPARQL rules to infer the evacuation priorities of demand points. For each priority, we define a rule as follows. We first define the node shapes representing the classes that describe the priorities and the property shapes representing the properties used to define the priorities; then, we define the rules. Referring to the axioms defined in the first approach for each priority class, they are now defined as rules determining the conditions of each evacuation priority. The SPARQL rule defining the priority "Evacuate in 12h" in the shape file is displayed as follows.

```
sh:rule [
rdf:type sh:SPARQLRule ;
sh:prefixes ns1: ;
sh:construct """
PREFIX ns1: <https://www.irit.fr/recherches/
MELODI/ontologies/i-Nondations.owl# >
CONSTRUCT
{?this ns1:priority ?priority.}
WHERE
{ ?this ns1:danger_index ?danger_index.
?this ns1:duration_of_flooding
?duration_of_flooding.
?this ns1:number_of_floors ?number_of_floors.
?this ns1:submersion_height ?submersion_height.
?this ns1:vulnerability_index ?vulnerability_index.
?this ns1:is_habitated ?is_habitated.
FILTER
(?danger_index >0
&& ?danger_index <50
&& ?duration_of_flooding ≥ 12
&& ?number_of_floors ≥ 1
&& ?submersion_height >0.0
&& ?submersion_height ≤ 1.0
&& ?vulnerability_index <50.0
&& ?is_habitated = true ) .
BIND ("12h_before_evacuation" AS ?priority).
}
""" ;
sh:condition ns1:12h_before_evacuation ;
```

The rules are executed on the knowledge graph to infer new triples. Each inferred triple consists of a demand point typed with an evacuation priority according to its properties. The knowledge graph is then enriched by adding the inferred triples to it, and it can be shared by different actors in the crisis management process.

5 EXPERIMENTAL EVALUATION

In this section, we discuss the experimental evaluations conducted in order to evaluate the three approaches for inferring the evacuation priorities of the demand points. The evaluations are divided into three main categories. The first category concerns analyzing the impact of the variation of the number of instances in a knowledge graph on the complexity of the process of inferring new information. The second category concerns analyzing the complexity of the evacuation priorities. The third category concerns analyzing the impact of the variation of the number of evacuation priorities on the complexity of the process of inferring new information.

All the conducted experiments run in 4h and 1min on 8 CPUs Core i7-1185G7, and draw 0.28 kWh. Based in France, this has a carbon footprint of 11.01 g CO_2e, which is equivalent to 0.01 tree-months (calculated using green-algorithms.org v2.1 (Lannelongue et al., 2021)).

5.1 Variation of Number of Instances

In our knowledge graph, evacuation priorities are inferred for demand points; therefore, the number of instances of demand points determines the number of evacuation priorities inferred. It also represents the number of times the conditions of each evacuation priority need to be tested in order to infer a priority for a demand point. We thus aim at analyzing the impact of the variation of the number of instances in the knowledge graph on the complexity of three approaches of inferring the priorities in terms of execution time.

5.1.1 Knowledge Graph

The knowledge graph containing all the flood-related data of our study area is composed of 472,594 triples. There are 15,078 demand points in our study area; therefore, 15,078 new triples representing evacuation priorities of demand points are inferred.

5.1.2 Experimental Results

A demand point is described by different properties and thus by different instances representing these properties. To analyze the impact of the variation of the number of instances, we study the execution time with decreasing percentages of demand points in the knowledge graph from 75% to 25% of the total number of demand points.

Table 1 displays the percentage of demand points in the knowledge graph (named KG in the tables) with the number of demand points for each percentage which thus represents the number of inferred evacuation priorities of demand points.

Table 1: Percentage of demand points in KG.

% of demand points in KG	**Number of evacuation priorities**
100 %	15,078
75 %	11,308
50 %	7,539
25 %	3,769

Table 2 presents the execution times (in seconds) of the three approaches for inferring the priorities with decreasing percentages of demand points in the knowledge graph. We recall that the three approaches are as follows: FaCT++ in Protégé, SPARQL insert and delete queries as well as SHACL rules.

Table 2: Execution times (s) of inferring priorities using 3 approaches with decreasing percentages of demand points.

	100% of KG	75% of KG	50% of KG	25% of KG
FaCT++	4464.2	1809.5	1131.5	164.1
SPARQL queries	80.62	41.37	23	8.09
SHACL rules	12.86	8.78	6.55	4.56

From the results in table 2, we notice that the execution times decrease with decreasing percentages of demand points in the knowledge graph using the three approaches. Inferring the priorities using SHACL rules takes 12.86 seconds for generating 15,078 priorities (100% of demand points in knowledge graph) and only 4.56 seconds for generating 3,769 priorities (25% of demand points in knowledge graph). It is the most efficient approach in terms of execution time compared to the two other approaches.

In the SPARQL approach, the knowledge graph should first be loaded on the triplestore which takes from 3 to 4.5 seconds depending on its size. After that, 2 SPARQL queries (delete and insert queries) are executed for each evacuation priority in order to define the priorities for all demand points and obtain the enriched knowledge graph. On the other hand, the chosen SHACL implementation is independent of a triplestore where the knowledge graph, the node-shapes of classes and properties as well as the rules are defined in a single Turtle RDF file. The new inferred triples are then added to the knowledge graph to obtain the enriched one.

FaCT++ in Protégé takes around 36 times longer than SHACL rules to infer the priorities for 25% of the demand points in the knowledge graph. We note that we use FaCT++ reasoner in Protégé which could not be conducted independently of all other Protégé functionalities; this would thus increase the complexity of the process which explains the long execution time of this approach compared to the other two approaches.

5.2 Evaluation of Complexity of Evacuation Priorities

We define the complexity of an evacuation priority as the number of demand points that are typed with this priority and the time taken to classify them using the three different approaches. An evacuation priority is defined as a set of conditions on certain properties that describe the demand points where these conditions are combined through ”and” and ”or” logic operators such that all the possibilities of demand points’ properties are considered for the four evacuation priorities in an exclusive manner. We define the worst case scenario of an evacuation priority as the case where all its conditions should be tested in order to determine if a demand point is typed with this priority.

The number of conditions defining each of the four evacuation priorities as well as the properties used in these conditions vary from one priority to another. For example, the property ”duration of flood” is not used in the conditions defining the priority ”Evacuate immediately”; however, it is used in the conditions defining the three other priorities. Table 3 displays the number of conditions defining each evacuation priority with considering the worst case scenario for all the priorities.

Table 3: Number of conditions constituting evacuation priorities.

Evacuation priority	**Number of conditions**
Evacuate in 12 hours	8
Evacuate in 6 hours	16
Evacuate immediately	18
No evacuation	27

Due to the different number of conditions and properties among the four evacuation priorities, we can tell that they have different complexity. We evaluate the complexity of each of the four evacuation priorities using the three different approaches.

5.2.1 Evaluation on Knowledge Graph Containing Real Data

The knowledge graph that contains the real flood-related data corresponding to our study area consists of 15,078 demand points with the various instances of properties describing them. In this experiment, we evaluate the complexity of the four evacuation priorities on this knowledge graph using the three approaches for inferring the priorities of demand points.

Table 4 shows the execution times (in seconds) of each evacuation priority in the three approaches, and table 5 presents the number of demand points that are typed with each priority.

Table 4: Execution times (s) of evacuation priorities.

Evacuation priority	**FaCT++**	**SPARQL queries**	**SHACL rules**
Evacuate in 12h	2080.72	12.88	6.61
Evacuate in 6h	2080.83	7.2	7.28
Evacuate immediately	2054.88	13.19	7.27
No evacuation	2424.88	64.52	9.96

Table 5: Number of demand points per evacuation priority.

Evacuation priority	**Number of demand points**
Evacuate immediately	412
Evacuate in 12h	398
Evacuate in 6h	31
No evacuation	14,237

From these results, we can conclude that the SHACL approach is the most efficient one when being applied on the real flood-related data of our study area, and the SPARQL approach is more efficient than the approach that uses FaCT++ in Protégé which proves to be inefficient on this knowledge graph. However, our interest is to analyze the complexity of the evacuation priorities further and in a more generic manner. Using this knowledge graph, it is difficult to draw more precise conclusions due to two main reasons. First, the number of conditions defining each evacuation priority is not identical among different priorities, and they are not necessarily defined by the same properties. Second, the number of demand points that are typed with each evacuation priority is not identical as we can see in table 5. For example, although the number of conditions defining the priority "Evacuate in 12 hours" is less than that of the priority "Evacuate in 6 hours", inferring the information related to the former priority takes more time (12.88 seconds) than the latter (7.2 seconds) using SPARQL queries. The reason is that there are 398 demand points typed with the former priority while only 31 demand points typed with the latter.

5.2.2 Evaluation using a Synthetic Knowledge Graph

In order to evaluate the complexity of the evacuation priorities more precisely, we propose to evaluate the impact of the number of conditions defining each evacuation priority on its complexity. To do that, there are two important factors to be taken into consideration. First, the number of demand points satisfying each evacuation priority should be fixed as it impacts the complexity. Second, the worst case scenario should be considered for all evacuation priorities in order to ensure a precise evaluation of their complexity. Considering the worst case scenario represents having demand points whose properties ensure the need of testing each and every condition in all the evacuation priorities. Therefore, we propose a new synthetically generated knowledge graph for the purpose of evaluating the complexity of the priorities.

This synthetic knowledge graph is generated such that there are four categories of demand points; each category contains 4,000 demand points whose properties satisfy the conditions of an evacuation priority. The total number of demand points in the knowledge graph is then 16,000. This knowledge graph is generated as follows. Random data are automatically generated representing random ID's of demand points with their properties respecting the conditions of the evacuation priorities. These data are then transformed into RDF triples to form the knowledge graph relying on the concepts and relations of our ontology. The RDF triples constituting the knowledge graph are generated using JENA Java library[15].

As the experimental results of the approach of inferring evacuation priorities using FaCT++ in Protégé has proved to be inefficient compared to the two other approaches in terms of time, we exclude this approach from the coming experiments, and we will thus only evaluate the two other approaches, that are SPARQL insert and delete queries and SHACL rules.

Table 6 presents the experimental results in terms of execution time (in seconds) of each evacuation priority using the two approaches.

We can notice from these results that the execution time increases as the number of conditions defining each priority increases in both approaches (refer to table 3 for the number of conditions defining each priority). This confirms that the number of conditions defining an evacuation priority impacts its complexity. In addition, inferring the priorities using the SHACL approach takes less time than the SPARQL approach for all the evacuation priorities which proves that it is the most efficient approach for

[15]https://jena.apache.org/

Table 6: Execution times (s) of evacuation priorities of the synthetic knowledge graph.

Evacuation priority	SPARQL queries	SHACL rules
Evacuate in 12h	104.54	7.97
Evacuate in 6h	128.49	8.59
Evacuate immediately	158.99	9.01
No evacuation	187.89	9.13

inferring the priorities.

5.3 Variation of Number of Evacuation Priorities

The number of evacuation priorities represents the number of times that the properties of each demand point are tested against the conditions of all the evacuation priorities in order to type this demand point with its appropriate evacuation priority. Therefore, the number of evacuation priorities tested in an integrated manner on the knowledge graph has an impact on the complexity of the process of inferring the evacuation priorities.

There are four evacuation priorities defined by the domain experts concerning the evacuation of victims. These priorities are defined such that they consider all the possibilities of demand points, and they are exclusive. In the following experiments, we aim at evaluating the impact of the variation of the number of evacuation priorities on the complexity of the process of inferring the priorities of demand points. For the sake of this evaluation, we propose defining new evacuation priorities in addition to the four priorities defined by the domain experts. As the demand points of our knowledge graph representing the real data of our study area are typed with the four existing evacuation priorities, we propose to enrich the knowledge graph so that it contains new demand points whose instances of properties satisfy the new evacuation priorities.

5.3.1 New Evacuation Priorities

The evacuation priority having the lowest complexity among the four priorities defined by the domain experts is ”Evacuate in 12h”. We thus propose to add three new evacuation priorities that have the same complexity as this priority for the sake of simplicity of evaluations. As previously mentioned, the four evacuation priorities are exclusive and consider all the possibilities. Therefore, in order to add new priorities, we choose to modify the existing evacuation priority ”Evacuate in 12h” by adding only one condition such that we can define new priorities from this one through modifying this condition and maintaining the exclusiveness of the different priorities. Let’s recall the conditions defining this evacuation priority as presented in figure 2. We propose to add a condition concerning the property ”duration of flooding” to allow the division of the values’ range of this property in order to define several evacuation priorities.

We then define three new evacuation priorities from the priority ”Evacuate in 12h”. The difference among these priorities is the value range of the property ”duration of flooding” as displayed in table 7.

Table 7: The intervals defining the property ”duration of flooding” in the new evacuation priorities.

Evacuation priority	”duration of flooding” interval
Evacuate in 12h (modified priority)	>=12h , <15h
Evacuate in 15h	>=15h , <18h
Evacuate in 18h	>=18h , <24h
Evacuate in 24h	>=24h , <1000h

5.3.2 Knowledge Graph Enrichment

We propose to enrich our knowledge graph with additional RDF triples representing demand points with their instances satisfying the three new evacuation priorities. For this purpose, we have generated synthetic RDF triples of three categories. Each category contains demand points whose instances satisfy the conditions of one of the new evacuation priorities. We define 398 demand points for each category that represents the number of demand points initially satisfying the evacuation priority ”Evacuate in 12h”. The enriched knowledge graph is generated similarly to the generation of the synthetic knowledge graph for the previous experimental evaluation, and it thus contains 16,272 demand points with their instances.

5.3.3 Experimental Evaluation

We first evaluate the complexity of the modified evacuation priority ”Evacuate in 12h” on the enriched knowledge graph using the two approaches: SPARQL insert and delete queries and SHACL rules. We then evaluate the complexity of the four existing evacuation priorities on the enriched knowledge graph using the two approaches. After evaluating the existing priorities, we evaluate the impact of the variation of the number of evacuation priorities on the process of inferring the priorities of demand points from the enriched knowledge graph using the two approaches. Table 8 displays the results of these evaluations in terms of execution time (in seconds) where ”4 priorities" represent the four existing evacuation priorities defined by the domain experts. We notice from

Table 8: Time for inferring priorities of demand points (s) with 7 evacuation priorities progressively.

New evacuation priority	SPARQL queries	SHACL rules
Evacuate in 12h	12.88	6.61
4 priorities	84.32	13.1
4 priorities + Evacuate in 15h	94.93	15.34
4 priorities + Evacuate in 15h + Evacuate in 18h	104.59	17.61
4 priorities + Evacuate in 15h + Evacuate in 18h + Evacuate in 24h	115.08	19.18

the results that the time increases after the addition of each evacuation priority using the two approaches. The time increase after adding each new priority is by around 10 seconds using the SPARQL approach which is close to the time of inferring the priority "Evacuate in 12 hours" for the corresponding demand points. While the time increase after adding each priority is around 2 seconds using the SHACL approach which is less than the time taken to infer the priority "Evacuate in 12 hours" for the corresponding demand points. It takes less time to infer the priorities of 16,272 demand points using the SHACL approach (19.18 seconds) than using the SPARQL approach (115.08 seconds).

5.4 Discussion

The evacuation process of flood victims is a critical process that should be rapid and efficient as it threatens the lives of the population. The SHACL approach is proved to be the most efficient one among the three approaches for inferring the evacuation priorities in a real use case as well as using synthetic data that are generated for various experimental purposes. It is capable of generating priorities for a big number of demand points in a short time compared to the two other approaches.

The constraint of inferring the evacuation priorities of demand points rapidly in order to assist in taking rapid actions concerning victims' evacuation is not satisfied in the approach of FaCT++ in Protégé due to the long time that it takes to infer the priorities. It is thus considered inefficient for inferring the priorities.

The delicacy of the evacuation process requires proposing solutions that are not only efficient and rapid but also simple to assist in this process. The persons in charge of the evacuation process are usually non-experts in novel technological techniques; therefore, the proposed solutions to assist in taking actions and decisions should facilitate the process. SPARQL queries defining the priorities can be integrated in a tool where a natural language query written by the user concerning an evacuation priority can be transformed into a SPARQL query that allows inferring the information about this priority using existing approaches that propose query transformations (Shaik et al., 2016; Ochieng, 2020). SHACL rules that are used in our third approach are based on SPARQL query language; therefore, it can take the same advantage of transforming natural queries to rules and thus can be integrated in a tool used by the users to obtain information. In addition, integrating SHACL rules in a tool would allow users to choose whether to activate or deactivate rules as well as to set an execution order to different rules based upon their needs.

6 CONCLUSION

Natural crises management is a critical process; therefore, we consider that it is important to propose solutions that help in the information management and sharing among different involved actors as well as to infer new information that assist in the crisis management process. In this paper, we have proposed a case study that relies on a previously proposed ontology and a knowledge graph integrating real flood-related data (Bu Daher et al., 2022) in order to evaluate three approaches for inferring new information representing evacuation priorities for demand points during a flood crisis. The three approaches were evaluated using real data as well as synthetic data for further analysis. The experimental results show that inferring the priorities using SHACL rules is the most efficient approach as it allows inferring the priorities in a short time and assist in taking rapid decisions and actions.

As a future work, we first aim at translating this work into an industrial usage through an interface that integrates SHACL Rules and allows to transform users' natural language queries to rules. This allows the users to obtain information about evacuation priorities of demand points in a study area, to activate or deactivate different rules as well as to set their execution order according to their needs.

We also aim at relying on the ontology and the knowledge graph in order to infer new information concerning the management of the resources (evacuation vehicles) that are used for the evacuation process and the routes organization. In addition to inferring new information that assists in the crisis response phase, we aim at inferring new information to improve a past crisis experience. In this frame, we aim at proposing a learning approach that learns from the data of a past crisis and adjusts the values of the properties that define the conditions of different evacuation priorities with the aim of improving the experience of this crisis.

ACKNOWLEDGMENTS

This work has been funded by the ANR in the context of the project "i-Nondations" (e-Flooding), ANR-17-CE39-0011[16].

REFERENCES

Bu Daher, J., Huygue, T., Stolf, P., and Hernandez, N. (2022). An ontology and a reasoning approach for evacuation in flood disaster response. In *17th International Conference on Knowledge Management 2022*, page To appear. University of North Texas (UNT) digital library.

Elmhadhbi, L., Karray, M. H., and Archimède, B. (2019). A modular ontology for semantically enhanced interoperability in operational disaster response. In *16th International Conference on Information Systems for Crisis Response and Management-ISCRAM 2019*, pages 1021–1029.

Franke, J. (2011). *Coordination of Distributed Activities in Dynamic Situations. The Case of Inter-organizational Crisis Management*. PhD thesis, Université Henri Poincaré-Nancy I.

Glimm, B., Horrocks, I., Motik, B., Stoilos, G., and Wang, Z. (2014). Hermit: an owl 2 reasoner. *Journal of Automated Reasoning*, 53(3):245–269.

Hill, E. F. (2003). *Jess in action: Java rule-based systems*. Manning Publications Co.

Katuk, N., Ku-Mahamud, K. R., Norwawi, N., and Deris, S. (2009). Web-based support system for flood response operation in malaysia. *Disaster Prevention and Management: An International Journal*.

Khantong, S., Sharif, M. N. A., and Mahmood, A. K. (2020). An ontology for sharing and managing information in disaster response: An illustrative case study of flood evacuation. *International Review of Applied Sciences and Engineering*.

Kurte, K. R., Durbha, S. S., King, R. L., Younan, N. H., and Potnis, A. V. (2017). A spatio-temporal ontological model for flood disaster monitoring. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 5213–5216. IEEE.

Kurte, K. R., Durbha, S. S., King, R. L., Younan, N. H., and Vatsavai, R. (2016). Semantics-enabled framework for spatial image information mining of linked earth observation data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(1):29–44.

Lannelongue, L., Grealey, J., and Inouye, M. (2021). Green algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, 8(12):2100707.

Ochieng, P. (2020). Parot: Translating natural language to sparql. *Expert Systems with Applications: X*, 5:100024.

Scheuer, S., Haase, D., and Meyer, V. (2013). Towards a flood risk assessment ontology–knowledge integration into a multi-criteria risk assessment approach. *Computers, Environment and Urban Systems*, 37:82–94.

Shaik, S., Kanakam, P., Hussain, S. M., and Suryanarayana, D. (2016). Transforming natural language query to sparql for semantic information retrieval. *International Journal of Engineering Trends and Technology*, 7:347–350.

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Journal of Web Semantics*, 5(2):51–53.

Sun, J., De Sousa, G., Roussey, C., Chanet, J.-P., Pinet, F., and Hou, K.-M. (2016). Intelligent flood adaptive context-aware system: How wireless sensors adapt their configuration based on environmental phenomenon events. *Sensors & Transducers*, 206(11):68.

Tsarkov, D. and Horrocks, I. (2006). Fact++ description logic reasoner: System description. In *International joint conference on automated reasoning*, pages 292–297. Springer.

Wang, C., Chen, N., Wang, W., and Chen, Z. (2018). A hydrological sensor web ontology based on the ssn ontology: A case study for a flood. *ISPRS International Journal of Geo-Information*, 7(1):2.

Yahya, H. and Ramli, R. (2020). Ontology for evacuation center in flood management domain. In *2020 8th International Conference on Information Technology and Multimedia (ICIMU)*, pages 288–291. IEEE.

[16]https://anr.fr/Projet-ANR-17-CE39-0011