ONTOLOGY ENABLED DISASTER MANAGEMENT WEB SERVICE USING DATA INTEGRATION

Shilpa P. Pimpalkar

Assistant Professor, AISSMS IOIT, Pune, Maharashtra, India. shilpa.pimpalkar@aissmsioit.org

Shreyas Suresh Rao

Associate Professor, Dept. Of CSE, Sahyadri College of Engineering and Management, Mangalore, India shreyasrao.cs@sahyadri.edu.in

Nitant Dube

Scientist/Engineer-G, SAC, ISRO, Ahmedabad, India nitant@sac.isro.gov.in

Abstract

Meteorological disasters caused by extreme weather may result in damage to infrastructure, transport, livelihood, agro-industries and affects the overall economy of a country. To reduce losses due to disasters, an effective information retrieval system is required with the capability of dissemination of information about preparedness and an early warning to the disaster management authorities as well as individuals caught in the disaster. The paper presents an implementation of a web service that demonstrates archival and dissemination of disaster-related information, alerts to users in a platform-independent and machine-friendly manner. The proposed Web service processes MOSDAC INSAT3D satellite data. MOSDAC (Meteorological and Oceanographic Satellite Data Archival Centre) is a data repository of the Indian Space Research Organization (ISRO) and Government of India that deals with meteorology, oceanography, and tropical water cycles. The proposed implementation of web service aims to disseminate rainfall data at a location, provide information about nearby places (e.g., airport, hotels, hospitals, information centers, etc.), and give specific location details (e.g., elevation, population, humidity, high-temp, low temp). Web service demonstrates the capability of dissemination of alerts, early warning related to rainfall, and information required to prepare better rescue plans, which will help individuals as well as disaster management authorities for disaster management.

Keywords: Ontology; Satellite Data; SPARQL; Linked Data

1. Introduction

The semantic Web is a generative technology that supports the representation of domain knowledge in Machine understandable form. It aims to build a giant global database of applications in the World Wide Web [Kamran and Sheraz ,2018]. With semantic technology, knowledge is represented using a formal specification of shared concepts, commonly referred to as "ontology" [Abburu and Dube ,2017]. Ontology provides users with the necessary structure to link one piece of information with other pieces on the Web of Linked Data. Ontology enables-1) common modeling representation of data from distributed and heterogeneous systems and databases, 2) database interoperability, 3) cross-database search, and 4) smooth knowledge management [Bernasconi et al. ,2020]. With ontology-based representation and querying of domain-specific knowledge, the performance of current information retrieval systems can be enhanced, and solutions can be provided for the issues of modern information systems [Paulo et. al. ,2020], [Jose et.al. ,2018], [Kai et al.,2019]. The effectiveness of semantic data retrieval can be enhanced through an ontology-based knowledge management system.

In this paper, we present novel research work that supports ontology-based satellite data retrieval and interlinking with other data sources on the Web (e.g., DBpedia). The paper is organized as follows: Section 2 presents the related research work and need for ontology-enabled satellite data, Section 3 illustrates the proposed process architecture, Section 4 enumerates the semantic concept-based query results, and Section 5 concludes the paper.

DOI: 10.21817/indjcse/2022/v13i1/221301037 Vol. 13 No. 1 Jan-Feb 2022 59

2. Related Research Work:

Satellite data plays an essential role in getting synoptic observations of climatic conditions [Thies and Bendix,2011]. Satellites collect data for weather, climate, and environmental monitoring applications. The data includes precipitation, sea surface temperatures, atmospheric temperatures & humidity, sea ice extent, forest fires, etc. Data collected from satellites help in better planning and management of disasters. [Abburu and Dube ,2017] The real-time data available through satellites at various data centers can be used for societal applications. Many researchers use ontologies for the semantic representation of satellite images and retrieval. Limited work has been carried out towards interoperability of satellite data with linked data sources available on the Web.

A few papers which propose a mechanism for ontology-enabled satellite data have been reported in this area, and we were able to find four such studies [Abburu and Dube ,2017], [Mitra et al. 2018], [Abburu and Dube,2015], [Alirezaie et al. ,2017]. One of these papers [Abburu and Dube ,2017] presents the development of strategies for ontology concept-based satellite data management, which evaluates the performance of the semantic query retrieval system by taking real-time data from MOSDAC and semantic queries related to weather, climate, forecast, and environment ontologies. The focus of the paper was providing a mechanism for ontology-based satellite data management.

Paper [Abburu and Dube,2015] proposed a generic approach representing sensor data products with international standard Sensor Parameter Ontology vocabulary supporting data exchange, information interoperability among various scientific schemes of data centers. This work focused on merging ontology vocabularies and querying data available in ISRO data centers.

Another paper [Mitra et al. 2018] presents assessments of operational INSAT-3D derived rainfall products and their comparison with IMERG/DMSG and actual rainfall data. Study presented shows that HEM (Hydro-Estimator Method) is better for heavy rainfall estimation.

Paper [Alirezaie et al. ,2017] presents a framework to classify satellite images and augment it with additional semantic information, which will enable queries about what can be found on the map at a particular location, but also about paths that can be taken.

Disaster prediction is always a significant activity, and its framework needs to be continuously updated with changing needs [Abburu and Dube ,2017]. Decision support/querying systems for disaster prediction can be effectively demonstrated by adapting semantic reasoning for the satellite data. The weather data can better support decision-making during a disaster if it is ontology enabled and interlinked with data from other sources. For this, data from different systems/sources needs to be represented using an ontology, interlinked, aggregated, and queried. The approach of combining various sources, which includes multidisciplinary, multitemporal, and heterogeneous datasets using Linked Data Principles, is advocated in the paper [Kopsachilis et al. ,2020] for publishing geospatial linked data.

We propose a novel approach to enrich satellite data using geopy, DBpedia, and Google Maps API, which is not currently available in the literature. Interlinking of satellite data with web resources is also not available in the literature. The proposed approach provides semantic interoperability and interlinking to DBpedia resources available on the Web. The proposed web service demonstrates semantic technology for satellite data processing, thereby delivering advanced solutions to societal applications (i.e., disaster management).

The proposed implementation uses the INSAT3D/3DR data received from MOSDAC, SAC, Ahmedabad. Data set used for the research is available online (MOSDAC basic search available at http://www.mosdac.gov.in/metaDataBrowse.do?mode=displaySatelliteForm.)MOSDAC data center hosts weather information services and disseminates the weather data through Web and mobile applications. This dissemination is beneficial to better assess the impact of a disaster and to improve decision support.

The proposed system demonstrates the representation, interlinking, and querying of ontology-enabled satellite data.

3. Process Architecture

The architecture is proposed to extract 'Hydro Estimation' rainfall data of INSAT3D/3DR and query the satellite data using the weather ontology. The snippet of database to ontology mapping is given in Table 1. The proposed system supports archival and dissemination of disaster-related information and alerts to users in a platform-independent manner (i.e., through web services) and in a machine-friendly mode.

Figure 1 shows the process architecture. The process consists of Offline (backend) and Web-based Online activities. First, as part of the backend activity, the Weather data in HD5 format is converted to CSV and subsequently to N-Triples format, based on the Weather Ontology. Second, the Administrator of the system uploads the weather file (N-Triples format) onto a Web portal (datastore). Third, the end-user uses the Web Portal frontend to query the weather data by providing search parameters such as latitude, longitude, city, etc. Based on the interlinking of the weather data with other data sources on the Web, the query returns the result, which is displayed to the end-user.

DOI: 10.21817/indjcse/2022/v13i1/221301037 Vol. 13 No. 1 Jan-Feb 2022 60

The detailed activities within the workflow are explained in the following diagram.

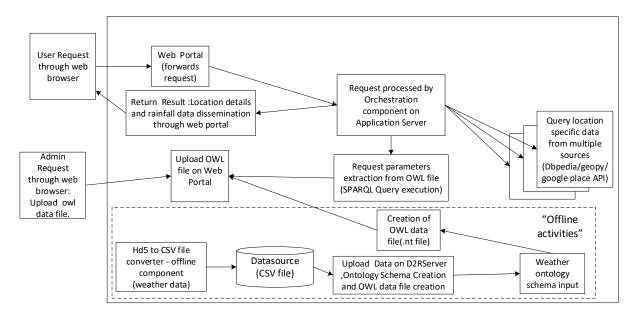


Figure 1. Process Architecture of the System

3.1 hd5 to CSV file converter

INSAT3D/3DR data is in .h5 format. This is a hierarchical file format. It needs to be converted to a CSV file. This file contains HE rainfall, latitude, and longitude for specific dates and times. This architecture supports the conversion of the hd5 file to CSV in offline mode (i.e., in batch mode). The CSV is modified by using the geopy and Google Location API. This module supports adding location-specific details such as latitude, longitude, rainfall, city, state, pin code, country, and an exact address.

3.2 Generation of OWL file:

Based on the data in the CSV file and weather ontology schema as inputs, an OWL data file is generated in the N-Triple format. D2R Server is used for the "triplification" process, i.e., converting the CSV data into N-Triple format, through the extraction, naming, enrichment, mapping, and conversion stages, as described in the paper [Rao and Nayak ,2017].

In our project, the weather ontology schema was created in Turtle format. Mapping of database to ontology entity types is shown in Table 1

Database to ontology mapping details						
Resource name: Location						
URI pattern: http://ontology.isro.in/weather/geolocation						
RDF Type: geo:	RDF Type: geo: Feature					
Database entity	Database	Ontology Entity	Ontology entity	Description		
	entity type		type			
Location	Table	geo: Feature	Class	Location Details		
Id	Column	rdfs: label	Property	Unique Id for Location		
Latitude	Column	gn: lat	Property	Latitude		
Longitude	Column	gn: long	Property	Longitude		
City	Column	geo: City	Property	City		
State	Column	geo: state	Property	State		
Country	Column	geo: country	Property	Country		
Countrycode	Column	geo: countryCode	Property	Countrycode		
Resource name: Address						
URI pattern: http://ontology.isro.in/weather/adddress						
RDF Type: locn: Address						
Address	Table	locn: Address	Class	Location Address		
Id	Column	rdfs: label	Property	Location ID mapped to		
				Address ID		

Region	Column	locn: addressArea	Property	Area of Location	
Postal_code	Column	locn: postcode	Property	Postal code of Location	
Address	Column	locn: fullAddress	Property	Detailed Address of	
				Location	
Resource name:	Rainfall				
URI pattern: http://ontology.isro.in/weather/resource					
RDF Type: vocab:rainfall					
Rainfall	Table	vocab:rainfall	Class	Rainfall rate	
Id	Column	vocab: rainfall_Id	Property	Location ID mapped to	
		_		Rainfall ID	
rainfall_rate	Column	sosa: precipitation_rate	Property	Precipitation rate	

Table 1: Database to ontology mapping details

3.3 Orchestration component

This component queries the weather database and location-specific data. This component queries ontology (.nt) file and also executes SPARQL query using JENA API on DBpedia endpoint. This component also supports querying using Google Maps API to query nearby places information. The details of these are provided below

3.3.1. Query Ontology (.nt file) and plot all the locations with specific rainfall categories.

The data received from INSAT 3D gives precipitation rate at a particular location in mm. This data can belong to one of the following categories, as shown in Table 2.

Categories of HEM data			
Precipitation rate	Category		
0.0 mm	No Rain		
2.5 mm to 7.5 mm	Light Rain		
7.6 mm to 35.5 mm	Light/Moderate Rain		
35.6 mm to 64.4 mm	Moderate heavy		
64.5 mm to 124.4 mm	Heavy Rain		
124.5 mm to 244.4 mm	Very Heavy		

Table 2. Categories of HEM data[Mitra et al. 2018]

Querying component allows end-user to select required category through drop-down option given in UI, based on a selected category all locations in India will be plotted on the map. For, e.g. If a user selects heavy rain, all the places with precipitation rates in the range 64.5 mm to 124.4 mm (heavy rainfall) will be plotted on the map.

3.3.2. Query location-specific data from multiple sources

Based on the latitude and longitude values specified by the end-user, location-specific data like nearby airports, train stations, hospitals, etc., will be fetched using Google map API and presented to the end-user. Information about nearby places such as railway stations, airports, hotels, hospitals, etc., may assist individuals and disaster management authorities to prepare appropriate rescue plans in the event of any disasters.

3.3.3. SPARQL query execution to query DBpedia endpoint

The proposed implementation demonstrates a Java application which *queries* the DBpedia *endpoint using* Apache *Jena*. Apache Jena framework supports SPARQL query execution which is the standard query language and protocol for Linked Open Data and RDF databases. SPARQL is designed to query a great variety of data. SPARQL can efficiently extract information hidden in non-uniform data, stored in various formats, and available from multiple sources.

Interlinking of satellite data with DBpedia: In our application, we have interlinked the weather satellite data with DBpedia data source based on the "city' attribute. We have used the SILK framework for this interlinking activity.

Proposed ontology-based architecture for information extraction and interlinking of satellite data is tested for various samples of .h5 files and observed that it provides results with 100% accuracy, completeness, computational efficiency, and consistency. Meteorological data centers are facing challenges of storing a large volume of data fetched from satellites. The proposed architecture executes query runtime and extracts required information without storage of repositories at data centers, thereby improving the storage efficiency. The following Table 3 lists the extracted data using Geopy, DBPedia, and Google Map API.

Enrichment of the Satellite data using Geopy, DBPedia, and Google Map API			
Data Source	Extracted Data		
INSAT 3D Satellite Data	Latitude		
	Longitude		
	Date		
	Time		
	Rainfall		
Geopy Google API (Data extraction based on Latitude and	State		
Longitude)	Country		
	Country Code		
	Postal Address		
DBpedia SPARQL query execution with DBpedia endpoint	Population		
(interlinking based on the city)	Elevation		
	Area		
	Humidity		
	Low temperature		
	High Temperature		
Google Map API (location details extraction based on latitude and longitude)	Nearest popular places (hospitals, railway station, airport, hotels)		

Table 3: Enrichment of the Satellite data using Geopy, DBPedia, and Google Map API

4. Result:

The proposed architecture is implemented and tested using Jena API. Satellite data is extracted into a CSV file. Geopy and Google location API is used to get additional location details such as rainfall, latitude, longitude, region city, state, country, country code, postal code, detail address.CSV file is re-generated with these additional location-specific details. An ontology map file is generated using the D2R server. Interlinking is implemented using the SILK tool. The Weather data file (in NT format) is deployed in Virtuoso Universal Server, which provides a SPARQL endpoint for querying. A java program was written to execute the user queries; the program runs the Jena SPARQL API against the hosted Virtuoso database, executes the query, and extracts the required rainfall data.

Table 4. shows a few SPARQL queries executed on NT file (Ontology enabled satellite database) using JENA API.

Sr. No	Sample SPARQL Query executed with Result			
1	Get latitude, longitude, country, city for criteria "heavy rainfall". SPARQL query tested on			
	sample ontology which retrieves location details for criteria "heavy rainfall". Sample result			
	shown here contains one record in result set due to limit set to 1 in query.			
	For given attribute "heavy rainfall" query returns city, country,lat,long			
	String queryString="prefix geo: <a +<="" href="https://www.geonames.org/ontology>\r\n" td="">			
	"prefix gn: \r\n" +			
	"prefix sosa: https://www.w3.org/ns/sosa/observableProperty>http://www.w3.org/ns/sosa/observableProperty>https://www.wa.org/ns/sosa/observableProperty>https://www.wa.org/ns/sosa/observableProperty			
	"prefix info: <a +"="" href="http://www.w3.org/2002/07/owl#>\r\n">\r\n"+"			
	"PREFIX rdf: \r\n"+			
	"PREFIX rdfs: <http: 01="" 2000="" rdf-schema#="" www.w3.org="">\r\n"+</http:>			
	"\r\n" +			
	"select? c?cn?lat?long\r\n"+			
	"where\r\n" +			
	"{\r\n" +			
	"?s geo:locatedIn ?o . \r \n" +			
	"?o gn:lat ?lat . \r \n" +			
	"?o gn:lat ?long . \r \n" +			
	"?o ?p \"heavy rainfall\" . \r \n"+			
	"?o geo:country ?cn . \r \n"+			
	"?o geo:city ?c . \r \n" +			
	"} limit 1";			

```
Result obtained from Jena SPARQL
              (?lat = "49.7712132") (?long = "49.7712132") (?cn = "Mongolia") (?c = "Bukhmurun")
              Get the Population for the city - "Ajmer" / "Bijnor" etc.
2
              PREFIX dbpedia-owl: <a href="http://dbpedia.org/ontology/">http://dbpedia.org/ontology/</a>
              PREFIX dbpedia: <a href="http://dbpedia.org/resource/">http://dbpedia.org/resource/</a>
              PREFIX dbp: <a href="http://dbpedia.org/property/">http://dbpedia.org/property/</a>
              SELECT DISTINCT (STR(?citylabel) AS ?citylabel) (STR(?pop) AS ?pop)
              WHERE {
                ?city rdf:type dbpedia-owl:City.
                ?city rdfs:label ?citylabel.
                ?city dbpedia-owl:populationTotal ?pop .
                FILTER ( lang(?citylabel) = 'en' and ?citylabel="Ajmer"@en )
              Result:
               citylabel pop
               Ajmer
                           542580
               Bijnor
                           115381
3
              Get the Population and settlement type (Metropolitan) for given city
              PREFIX dbpedia-owl: <a href="http://dbpedia.org/ontology/">http://dbpedia.org/ontology/</a>
              PREFIX dbpedia: <a href="http://dbpedia.org/resource/">http://dbpedia.org/resource/</a>
              PREFIX dbp: <a href="http://dbpedia.org/property/">http://dbpedia.org/property/</a>
              SELECT DISTINCT (STR(?citylabel) AS ?citylabel) (STR(?pop) AS ?pop) (STR(?settleType) AS
              ?settleType)
              WHERE {
                ?city rdf:type dbpedia-owl:City.
                ?city rdfs:label ?citylabel.
                ?city dbpedia-owl:populationTotal ?pop .
                ?city dbp:settlementType ?settleType .
                FILTER (lang(?citylabel) = 'en' and ?citylabel="Ajmer"@en)
              Result:
              citylabel
                                                  settleType
                                    pop
              Ajmer
                                   542580
                                                  Metropolis
                                   115381
              Bijnor
                                                      city
4
              Get the Population, Settlement Type, Latitude, Longitude for given city = "Ajmer"
              PREFIX dbpedia-owl: <a href="http://dbpedia.org/ontology/">http://dbpedia.org/ontology/</a>
              PREFIX dbpedia: <a href="http://dbpedia.org/resource/">http://dbpedia.org/resource/</a>
              PREFIX dbp: <a href="http://dbpedia.org/property/">http://dbpedia.org/property/</a>
              PREFIX geo: <a href="http://www.w3.org/2003/01/geo/wgs84">http://www.w3.org/2003/01/geo/wgs84</a> pos#>
              SELECT DISTINCT (STR(?citylabel) AS ?citylabel) (STR(?pop) AS ?pop) (STR(?settleType) AS
              ?settleType) ?lat ?long
              WHERE {
                ?city rdf:type dbpedia-owl:City.
                ?city rdfs:label ?citylabel.
                ?city dbpedia-owl:populationTotal ?pop .
                ?city dbp:settlementType ?settleType .
                ?city geo:lat ?lat .
                ?city geo:long ?long .
                FILTER ( lang(?citylabel) = 'en' and ?citylabel="Ajmer"@en )
```

```
Result:
             Citylabel
                                               settleType
                                                                  lat
                                                                                long
                                 pop
                                                Metropolis 26.4499
             Ajmer
                                542580
                                                                              74.6399
             Get the Average Rainfall (mm), Minimum Temperature (degree), Maximum Temperature
5
             (degree) and Elevation above Sea Level for City="Ajmer" and Month="January"
             PREFIX dbpedia-owl: <a href="http://dbpedia.org/ontology/">http://dbpedia.org/ontology/</a>
             PREFIX dbpedia: <a href="http://dbpedia.org/resource/">http://dbpedia.org/resource/</a>
             PREFIX dbp: <a href="http://dbpedia.org/property/">http://dbpedia.org/property/</a>
             PREFIX geo: <a href="http://www.w3.org/2003/01/geo/wgs84">http://www.w3.org/2003/01/geo/wgs84">pos#>
             SELECT DISTINCT (STR(?citylabel) AS ?citylabel) ?highTemp ?lowTemp ?rainfall ?elevation
             WHERE {
               ?city rdf:type dbpedia-owl:City.
               ?city rdfs:label ?citylabel .
               ?city dbp:janHighC ?highTemp .
               ?city dbp:janHighC ?highTemp .
               ?city dbp:janLowC ?lowTemp .
               ?city dbp:janRainMm ?rainfall .
               ?city dbo:elevation ?elevation .
               FILTER ( lang(?citylabel) = 'en' and ?citylabel="Ajmer"@en )
             Result:
              citylabel
                                 highTemp
                                                      lowTemp
                                                                           rainfall
                                                                                                elevation
              Ajmer
                                    22.9
                                                                               7.3
                                                                                                   200.0
                                                        7.6
             Get city, state, country, precipitation rate, elevation: SPARQL query on interlinking based on
6
             "City" attribute
             String queryString="prefix geo: <a href="https://www.geonames.org/ontology>\r\n" +
                                  "prefix gn: <a href="mailto://www.w3.org/2003/01/geo/wgs84">\r\n" +
                                  "prefix sosa: <a href="https://www.w3.org/ns/sosa/observableProperty">https://www.w3.org/ns/sosa/observableProperty>htm" +
                                  "prefix info: <a href="http://www.w3.org/2002/07/owl#>\r\n"+">"+
                                  "\r\n" +
                                  "select "+ c+"
                                                    ?state ?country ?rate
                                                                              ?db\r\n" +
                                  "where\r\n" +
                                  "\{\r\n" +
                                 "?s geo:located
In ?o . \r \n" +
                                 "?o gn:lat "+latval+" . r = +
                                 "?o gn:long "+longval+" .\r\n" +
                                 "?o gn:long ?long1 . \r\n" +
                                 "?o geo:city"+c+" . \r \n"+
                                 "?o geo:state?state. \r\n" +
                                 "?o geo:country?country. \r\n"+
                                 "?o info:sameAs ?db .\r\n"+
                                 "?s sosa:precipitation rate ?rate .\r\n" +
                                  "} limit 1";
             String interlink="PREFIX dbpedia-owl: <a href="http://dbpedia.org/ontology/>\r\n"+
                                  "PREFIX dbpedia: <a href="http://dbpedia.org/resource/">http://dbpedia.org/resource/>\r\n"+
                                  "PREFIX dbp: <a href="http://dbpedia.org/property/>\r\n"+">http://dbpedia.org/property/>\r\n"+"
                                 "PREFIX rdf:<a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">\r\n"+
                                 "PREFIX rdfs:<a href="http://www.w3.org/2000/01/rdf-schema#>\r\n"+
                                 "PREFIX geo: <a href="http://www.w3.org/2003/01/geo/wgs84">http://www.w3.org/2003/01/geo/wgs84">pos#>\r\n"+
                                  "PREFIX dbo: <a href="http://dbpedia.org/ontology/>\r\n"+
                                  "SELECT DISTINCT ?citylabel ?pop ?highTemp ?lowTemp ?rainfall
```

```
?elevation\r\n"+

"WHERE {\r\n"+

"?city rdf:type dbpedia-owl:City. \r\n"+

"?city rdfs:label ?citylabel.\r\n"+

"?city dbo:elevation ?elevation . \r \n"+

"FILTER ( ?citylabel=\"Bijnor\"@en )}";

Result:

( ?state = "Uttar Pradesh" ) ( ?country = "India" ) ( ?db = <a href="http://dbpedia.org/resource/Bijnor">http://dbpedia.org/resource/Bijnor</a>) ( ?rate = "0.16990946E0" ) ( ?.0 = "Bijnor" ) ( ?citylabel = "Bijnor"@en ) ( ?elevation = "225.0"^^xsd:double)
```

Table 4. shows few SPARQL queries executed on NT file (Ontology enabled satellite database) using JENA API.

Figure 2 shows a screenshot of the cloud-based web service, which demonstrates the capability to disseminate alerts through the web portal. The web portal is accessible using the public IP of cloud-based Virtual Machines. An application server is deployed on the Virtual Machines. HTTP request submitted through the web interface is processed by the application server.

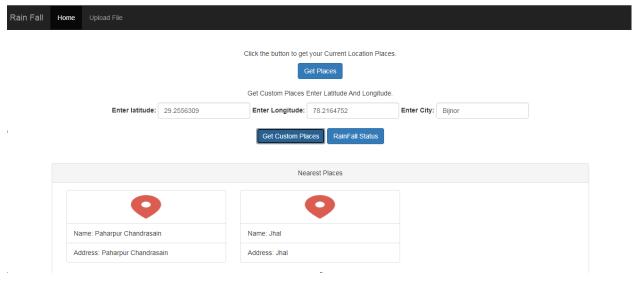


Figure 2. Cloud-based Service for Dissemination of Rainfall Alerts and Information about nearby places based on latitude and longitude

Web service also supports ontology-enabled semantic querying of satellite data. Web service can plot the locations based on the selected category of the rainfall (e.g., plot all locations with light precipitation rate) as shown in figure 4.

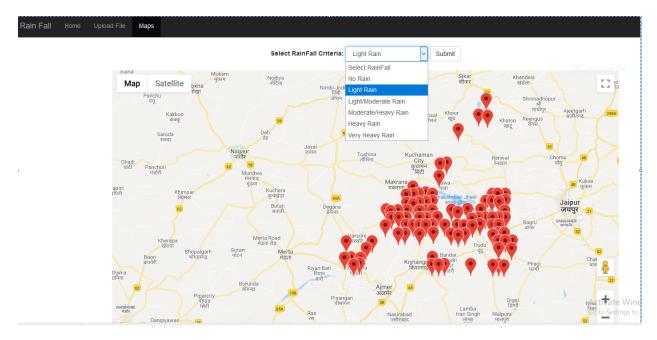


Figure 3. Cloud-based Service for plotting the locations based on the selected category of the rainfall

Linked data which is generated as an outcome of this work, is queried by a client through the web interface for dissemination of the alerts. Linked data generated can be processed in Machine interpretable form and platform-independent way. Overall statistics of the linked weather dataset are shown in Table 5.

Statistics of the Linked Weather dataset		
Description	Count	
Triples	631277	
Entities	78883	
Classes	5	
Properties	18	
Distinct Subjects	78884	
Distinct Objects	277686	
Linked resources with DBpedia	325	

Table 5. Statistics of the Linked Weather dataset (Total no of triples, entities, classes, properties, subjects, objects, linked resources)

5. Conclusion:

Ontology-based semantic representation of satellite data enables efficient data processing and retrieval. In this paper, a semantic web-based architecture is proposed and implemented, which can extract satellite data (INSAT3D/INSAT3DR), map ontology for the data, and execute SPARQL queries based on the weather ontology. This paper also proposes interlinking rainfall data with DBpedia based on the 'city' attribute. In addition, a web service is written for the dissemination of alerts, which is cloud-hosted. The proposed mechanism demonstrates knowledge management, information retrieval, and delivering refined results for

disaster management. The proposed solution significantly contributes to developing state-of-the-art data processing approaches that enrich satellite data and provides semantic interoperability with DBpedia data sources.

Acknowledgment:

This work is sponsored by the Indian Space Research Organization [Project No:182 Ref No: PU/ISRO-STC/2056 dated 12/03/2020]. We are thankful to ISRO-SPPU STC Cell and ISRO SAC Ahmedabad for guidance and support for the work.

Data Availability:

The data that support the findings of this study is available with the corresponding author Shilpa P. Pimpalkar

References

- [1] Abburu S. and Dube N.(2017), "Ontology Concept-Based Management and Semantic Retrieval of Satellite Data", Journal of Intelligent Systems, 26(2): 197-213, 2017.
- [2] Abburu Sunitha; Dube Nitant (2015), "An Ontology-Based Methodology for Satellite Data Semantic Interoperability, "Advances in Electrical and Computer Engineering Volume 15, Number 3, 2015.
- [3] Alirezaie Marjan ; Kiselev Andrey; Langkvist Martin; Klugl Franziska; Loutf Amy,(2017) "An Ontology-Based Reasoning Framework for Querying Satellite Images for Disaster Monitoring, DOI:10.3390/s17112545 Sensors (Basel, Switzerland)
- [4] Bernasconi Anna; Canakoglu Arif; Masseroli Marco; Ceri Stefano (2020), "META-BASE: ANovel Architecture for Large-Scale Genomic Metadata Integration" IEEE/ACM Transactions on Computational Biology and Bioinformatics 2020
- [5] Huang; Weiming; Harrie; Lars. (2020), "Towards knowledge-based geovisualisation using Semantic Web technologies: a knowledge representation approach coupling ontologies and rules" 13(9) 1753-8947, 2020, DOI: 10.1080/17538947.2019.1604835
- [6] Jose Aguilar; Marxjhony Jerez; Taniana Rodríguez. (2018), "CAMeOnto: Context awareness meta ontology modeling", Elsevier Applied Computing and Informatics Volume 14, Issue 2, July 2018, Pages 202-213
- [7] Kai Sun ;Yunqiang Zhu; Peng Pan; ZhiweiHou; Dongxu Wang; Weirong Li ;Jia Song.(2019), "Geospatial data ontology: the semantic foundation of geospatial data integration and sharing", Volume 3, 2019 Issue 3: Geospatial semantics, ontology and knowledge graphs for big Earth data
- [8] Kamran Munir, Sheraz Anjum M. (2018), "The use of ontologies for effective knowledge modeling and Information Retrieval", Applied Computing and Informatics, Volume 14, Issue 2, July 2018, Pages 116-126
- [9] Kopsachilis Vasilis; Michail Vaitis; Nikos Mamoulis; Dimitris Kotzinos. (2020), "Recommending Geo-semantically Related Classes for Link Discovery," Journal on Data Semantics, volume 9, pages 151–177 (2020)
- [10] Mitra A.K.; Kaushik Nitesh; Singh Anil Kumar; Parihar Shailesh; Bhan S.C. (2018), "Evaluation of INSAT-3D satellite-derived precipitation estimates for heavy rainfall events and its validation with gridded GPM (IMERG) rainfall dataset over the Indian region", Remote Sensing Applications: Society and Environment 2018.
- [11] Paulo H. L. Rettore; Bruno P. Santos; Roberto Rigolin F. Lopes; Guilherme Maia. (2020), "Road Data Enrichment Framework Based on Heterogeneous Data Fusion for ITS", IEEE transaction on Intelligent Transporation Systems, Vol. 21, No. 4, April 2020
- [12] Rao S.S.;Nayak A. (2017), "LinkED: A Novel Methodology for Publishing Linked Enterprise Data," Journal of Computing and Information Technology, 25(3): 191-209, 2017. DOI: 10.20532/cit.2017.1003477
- [13] Thies B.; Bendix. J. (2011), "Satellite-based remote sensing of weather and climate: recent achievements and future perspectives", Meteorological Applications, 18(3): 262-295, 2011.

Authors Profile



Shilpa P. Pimpalkar is working as Assistant Professor in the Department of Computer Engineering at AISSMS IOIT, Pune, Maharashtra, India. She has 18+ years of experience in teaching.

Her qualification is ME (Computer Engineering) from Pune university. Worked as Principal Investigator for the research project "Ontology enabled Disaster Management Web Service using Data Integration" sponsored by Indian Space Research Organisation (Duration June 2018 to July 2021). Worked as consultant for upGrad as Expert for the subject Software Design and Analysis.

Her research interests include Machine Learning, Natural Language Processing, Semantic Web and Web Services.



Dr. Shreyas Suresh Rao is working as Associate Professor and Research Coordinator in the Department of Computer Science and Engineering at Sahyadri College of Engineering and Management, Mangalore. He has 17+ years of experience in IT, Teaching, and Research domains.

He obtained Ph.D. from MIT, Manipal Academy of Higher Education (MAHE) in the field of Enterprise Software Engineering in 2018. He has 7 Scopus indexed journal publications, 3 SCI publications, 1 patent (filed), and several conference publications at national and international levels. He has an MS degree from BITS, Pilani in the field of Software Systems.

His previous stint includes working as Assistant Professor at Presidency

University and Sir MVIT. He worked as a Business Analyst and Team Lead at SLK Software Services Pvt. Ltd, Bangalore for a period of 7 years. His research interests include Deep Learning, Natural Language Processing, Semantic Web, Linked Open Data, Crowdsourcing, Knowledge Engineering, and Web Services.



Dr. Nitant Dube joined Space Applications Centre (SAC), ISRO in 1993 after completing his Computer Engineering. He has been responsible for Design and Development of Data Products and Information Processing Systems for Indian Remote Sensing Satellites. He has contributed towards Development and Operationalization of Data Processing Systems at Indian and International ground stations. He has played a key role in development of Indigenous Meteorological Data Processing and Dissemination System. Currently as Group Director, MOSDAC Research Group, he is involved in design development of Geospatial software for Meteorological and Oceanographic applications. He is ISRO nominated member for CEOS Working Group on Information System and Services (WGISS). His research field includes Satellite Image Processing, Big Data Analytics, Service Oriented Architecture, Geo-Intelligence and Web-based processing.