H2020 – Secure societies - Protecting freedom and security of Europe and its citizens

SU-DRS02-2018-2019-2020– Technologies for first responders – Research and Innovation Action (RIA)

**Emerging technologies for the Early location of Entrapped victims under Collapsed Structures & Advanced Wearables for risk assessment and First Responders Safety in SAR operations**

# D3.3 BIM based services and applications - review and service design

| | |
|---|---|
| **Workpackage:** | WP3 – Situation Awareness |
| **Authors:** | UBI |
| **Status:** | FINAL |
| **Due Date:** | 30/09/2021 |
| **Version:** | 1.00 |
| **Submission Date:** | 30/09/2021 |
| **Dissemination Level:** | PU |

(*) Dissemination level.-PU: Public, fully open, e.g. web; CO: Confidential, restricted under conditions set out in Model Grant Agreement; CI: Classified, Int = Internal Working Document, information as referred to in Commission Decision 2001/844/EC.

# Search and Rescue Project Profile

**Grant Agreement No.:** 882897

| | |
|---|---|
| **Acronym:** | Search and Rescue |
| **Title:** | Emerging technologies for the Early location of Entrapped victims under Collapsed Structures & Advanced Wearables for risk assessment and First Responders Safety in SAR operations |
| **URL:** | Website |
| **Start Date:** | 01/07/2020 |
| **Duration:** | 36 months |

## Partners

| | | |
|---|---|---|
| | NATIONAL TECHNICAL UNIVERSITY OF ATHENS (NTUA)<br>Co-ordinator | Greece |
| | AIDEAS OÜ (AIDEAS) | Estonia |
| | SOFTWARE IMAGINATION & VISION S.R.L (SIMAVI) | Romania |
| | MAGGIOLI SPA (MAG) | Italy |
| | KONNEKT-ABLE TECHNOLOGIES LIMITED (KT) | Ireland |
| | THALES ITAIA Italia SPA (THALIT) | Italy |
| | ATOS IT SOLUTIONS AND SERVICES IBERIA SL (ATOS) | Spain |
| | ETHNIKO KENTRO EREVNAS KAI TECHNOLOGIKIS ANAPTYXIS (CERTH) | Greece |
| | UNIVERSITA DEGLI STUDI DI CAGLAIRI (UNICA) | Italy |

| | | |
|---|---|---|
| | UKEMED GLOBAL LTD (UGL) | Cyprus |
| | PUBLIC SAFETY COMMUNICATION EUROPE FORUM AISBL (PSCE) | Belgium |
| | UNIVERSITA DEGLI STUDI DI FIRENZE (UNIFI) | Italy |
| | DEUTSCHES FORSCHUNGSZENTRUM FUR KUNSTLICHE INTELLIGENZ (DFKI) | Germany |
| | UNIVERSITA CATTOLICA DEL SACRO CUORE (UCSC) | Italy |
| | VRIJE UNIVERSITEIT BRUSSEL | Belgium |
| | SYNYO GmbH (SYNYO) | Austria |
| | UNIVERSITEIT HASSELT (UHASSELT) | Belgium |
| | SPOLECZNA AKADEMIA NAUK (SAN) | Poland |
| | GIOUMPITEK MELETI SCHEDIASMOS YLOPOIISI KAI POLISI ERGON PLIROFORIKIS ETAIREIA PERIORISMENIS EFTHYNIS (UBITECH) | Greece |
| **Search and Rescue End-Users** | | |
| | ELLINIKI OMADA DIASOSIS SOMATEIO (HRT) | Greece |

| | | |
|---|---|---|
| | ENOSI PTYCHIOYCHON AXIOMATIKON YPAXIOOMATIKON PYROSVESTIR OY SOMATEIO (EPAYPS) | Greece |
| | JOHANNITER-UNFALL-HILFE EV (JOHAN) | Germany |
| | JOHANNITER OSTERREICH AUSBLIDUNG UND FORSCHUNG GEMEINNUTZIGE GMBH (JOAFG) | Austria |
| | CONSIGLIO NAZIONALE DELLE RICERCHE | Italy |
| | POMPIERS DE L'URGENCE INTERNATIONALE (PUI) | France |
| | ASOCIATA CLUSTERUL ROAMN RENTRU PROTECTIE SI ECOLOGIE IN DOMENIUL MATERIALELOR CHIMICE, BIOLOGICE, RADIOLOGICE/NUCLEARE SI EXPLOZIVE (PROECO) | Romania |
| | SERVICIO MADRILENO DE SALUD (SERMAS) | Spain |
| | FUNDACIÓN PARA LA INVESTIGACIÓN E INNOVACIÓN BIOSANITARIA DE ATENCIÓN PRIMARIA (FIIBAP) | Spain |
| | ESCUELA ESPANOLA DE SALVAMENTO Y DETECCION CON PERROS (ESDP) | Spain |

# Document History

| Version | Date | Author (Partner) | Remarks/Changes |
|---|---|---|---|
| 0.10 | 30/07/2021 | George Vafeiadis (UBI) | Table of Contents |
| 0.20 | 04/08/2021 | George Vafeiadis (UBI) | Draft sections 1,2,4,5,6 |
| 0.40 | 06/08/2021 | Iosif Angelidis, George Vafeiadis (UBI) | Draft section 3 and additions in section 4 |
| 0.50 | 20/08/2021 | KT, SIMAVI | Draft section 4.2.4 |
| 0.60 | 06/09/2021 | KT, UNIFI | Quality and Technical Review |
| 0.70 | 17/09/2021 | Iosif Angelidis, Elena Politi (UBI) | Version ready for Quality Control |
| 0.80 | 29/09/2021 | PROECO | STM Quality Control |
| 0.85 | 29/09/2021 | Iosif Angelidis, Elena Politi (UBI) | Version addressing comments of STM Quality Control |
| 0.90 | 30/09/2021 | Christos Ntanos (NTUA) | Quality Review |
| 1.00 | 30/09/2021 | Christos Ntanos (NTUA) | Final version to be submitted |

# Executive Summary

The current deliverable D3.3 "BIM based services and applications – review and service design" aims at documenting the web service that exposes specific interfaces with the aim of representing building data, while in parallel, the core entities of the Building semantic model are presented. In the frame of the service which exposes the Building data, a RESTful web service is implemented in order to facilitate the data manipulation towards the creation of the necessary correlations which constitute the semantic model. These correlations will be defined for the first responders to have a clear overview of the building during a S&R operation. At the same time, the core components of the web service will be documented. Each of these components aims to fulfil specific requirements for the scope of the service to be delivered, not only against the BIM domain, but also against the Situation Awareness domain and its concepts, which have been documented in D3.2. Moreover, due to the fact that the implementation of the web service is currently ongoing and its first release is expected by M18, an example of how the data can be exposed will be provided. The second and final iteration of the web service, along with the completed semantic model for the building data will be presented in D3.4 "BIM based visualisation support integrated with VR interface" on M30.

# Table of Contents

## ACRONYMS

| | |
|---|---|
| D | Deliverable |
| SA | Situation Awareness |
| BIM | Building Information Modelling |
| UC | Use Case |
| S&R | Search and Rescue |
| EMS | Emergency Management System |

# List of Figures

# List of Tables

# 1 Introduction

## 1.1 Scope and Objectives

This deliverable constitutes a report aiming at documenting the technical activities undertaken within the context of T3.2 "Situation Awareness Model", along with the specification of the Building semantic model. These technical activities, which are described in the present document, provide a coherent overview of the semantic model web service as a whole, highlighting the interactions among its components and emphasizing on how both the SA model and the Building model can be exposed and utilised by other components such as the Emergency Communication System (EMS) in order to ensure, through several correlations, the most up to date operational picture during rescuer operations. Based on this principle, the Building semantic model aims at representing the building data in a way where the first responders will be assisted during their operations.

Considering the aforementioned statements, the main objective of D3.3 is to provide a comprehensive documentation of the semantic model web service, highlighting the key technical aspects of how building data can be utilised through specific interfaces. More specifically, D3.3 presents the components of this web service, along with the core entities which constitute the Building semantic model and how they can be utilised by the other components.

Due to the fact that the Semantic model web service follows an agile methodology in terms of development, this deliverable, which presents the first version of the service that is under implementation, may have updates as the implementation phase is ongoing. Having said that, the second and final release may comprise of additional functionalities, by taking into account the technical workshops which are going to be performed among technical partners. The same applies to the Building semantic model, which may be enriched or altered based on the data availability.

## 1.2 Relationship with Other Tasks and Deliverables

The Deliverable D3.3 documents the technical activities undertaken in the frame of Task T3.2 "Situation Awareness Model". The main objective of this document is to provide a coherent version in terms of technical implementation for both semantic model web service and the Building semantic model specification, by taking under consideration the input from the following deliverables:

- D3.1 "Requirements to knowledge management and SA Model," where general requirements, entities and actors that interacting with each other are described.
- D3.2 "Situation Awareness Model - specification", where the Situation Awareness model and its core concepts are specified. In this deliverable the SA model that is going to be utilized through the semantic model web service is presented.

## 1.3 Structure of the document

In order to address all the aspects relevant to the scope of D3.3, the present deliverable is structured as follows:

- Section 1 introduces the work performed and the scope of this deliverable along with its relevance to other deliverables and the deliverable's structure.

- Section 2 provides an overview of the working methodology by highlighting the way which the Building semantic model has been defined and how it is utilized through the corresponding web service. Furthermore, the main objectives of the semantic model and its utilization is provided.
- Section 3 provides an overview of existing approaches around semantic models that expose knowledge-based building data.
- Section 4 provides a comprehensive documentation of the semantic model web service, which relies on what its components are expected to support from the technical point of view and how the service interacts with other components.
- Section 5 offers an overview of the Building Semantic Model, its core components and correlations, along with how all this information can be exposed.
- In section 6, the final conclusions are provided along with a summary of the main outcome of the BIM based service that is under implementation.

# 2 Methodology

The definition of the Building semantic model, as well as the documentation of the semantic model web service will be delivered in two releases. The first releases of these technical activities, which are documented in the present deliverable, constitute the second part of work that has been performed in the frame of task T3.2 "Situation Awareness Model" until M15. The first part of the technical activities which have been undertaken within the context of this task is about the specification of the Situation Awareness model, which has been documented in D3.2. The second releases will be delivered in M30 and will document the updated version of the BSM, along with the web service based on the relevant work.

The methodology which has been followed for the first version of the definition of the Situation Awareness model includes the following steps:

1. Utilization of the backbone of the web service which has been initiated towards the exposure of the SA model.
2. Definition of the main concepts which have been identified through research based on existing approaches around Building data and semantic models.
3. Construction of the ontology which will represent the Building semantic model by interconnecting the concepts among them.
4. Extension of its components and features accordingly, in order for both the proper semantic model and correct information to be handled. D3.2 was an input source for the accomplishment of this goal.

Due to the fact that the Building semantic model will be the second model that will be constructed within the context of Task T3.2 "Situation Awareness Model", the objectives and the principles of the aforementioned model and its construction will rely on those which have been documented in D3.2 with respect to the Situation Awareness model.

Regarding the semantic model web service, specific characteristics were taken under consideration, which assist the exposure of interfaces in a scalable way. More specifically, some of the key characteristics of a RESTful Web service is a) the explicit use of HTTP methods, b) the fact that they are stateless and a complete request does not require the server while processing the request, c) the definition of directory structure-like URIs by achieving a high level usability, c) the resource representation can be exposed in a standardised format through XML or JSON. Taking all these key aspects into account, the semantic model web service has been implemented with Spring Boot, which is used for the construction of RESTful web services (among other things).

# 3 Existing Approaches Around BIM

In this section, existing approaches in BIM semantic models will be presented, which will be the starting point of the conceptual representation of the Building semantic model in the context of the project. More specifically, several approaches have already been introduced around Building information modeling (BIM) and Geographic Information Systems (GIS), based on using semantic web services technologies. One of these approaches introduces a model which uses BIM capabilities to accurately provide existing information about the indoor environment of buildings and the GIS, to support the wide range of spatial analysis functionalities used in the logistics and management of buildings within their surroundings [1]. Moreover, another approach aims to tackle the challenge which derives from the difficulty of data exchange between two different modelling domains. The proposed solution comes from the benefits of integrating BIM and GIS technologies into one unified model, a so-called Integrated Geospatial Information Model (IGIM) [2]. Last, but not least, Liu et al. [6] introduces a State-of-the-Art analysis on BIM and GIS approaches, as well as the integration methods between them.

Additionally, several approaches which introduce semantic representation of the BIM model can also be taken under consideration. Matějka et al. [3] propose a BIM ontology by using schematics and explaining the role of BIM in traditional project life cycle phases. Zhong eta al. [4] present a framework which integrates building information from BIM, environmental information provided by sensors, and regulatory information based on building regulations and design requirements. In this framework, four specific ontologies are developed to represent relevant knowledge. Finally, Niknam et al. [5] demonstrate a shared ontology approach to semantic representation of building information which facilitates finding and integrating building information distributed in several knowledge bases.

Considering the necessity of the combination of the BIM model with the Geospatial Information, as well as the fact that existing approaches focus on the construction of a unified model which incorporates both of them, specific ontologies which represent these data models conceptually are also considered. First and foremost, the BOT ontology[1] provides the vocabulary to describe the topology of a building as well as the relationships between their main components such as storey, zones, spaces, and building elements. This ontology will be utilized by the Building semantic model in order for the core concepts of a building to be represented. Additionally, The OGC GeoSPARQL[2] standard will be reused, since it supports the representation of geospatial data on the Semantic Web. GeoSPARQL defines a vocabulary for representing geospatial data in RDF, and it defines an extension to the SPARQL[3] query language for processing geospatial data.

[1] https://w3c-lbd-cg.github.io/bot/

[2] https://www.ogc.org/standards/geosparql

[3] https://www.w3.org/TR/rdf-sparql-query/

# 4 BIM and Situation Awareness models as a Service

## 4.1 Semantic Representation

In order to provide the semantic models as a service, a robust technological stack is utilized, both for offering the required REST[4] functionality and for storing the data in an efficient database. The latter is particularly important due to the nature of Search and Rescue, where performance and immediate data processing can be a determining factor between failure and success in operations. To that end, the semantic model web service aims at providing specific interfaces by handling both the Building semantic model and the SA semantic model in a scalable and efficient way. Towards this direction, some core functionalities need to be fulfilled in order for the proper interfaces to be exposed to the proper actors. Having said that, the core functionalities of the semantic model web service which denote its scope can be summarized as follows:

- **Data storage**: The semantic model web service aims at persisting the correlations among resources and metadata which have retrieved by utilising its RDF triplestore. The RDF triplestore is a type of graph database which stores semantic facts. More specifically, triplestores store data as a network of objects with materialized links between them. This makes RDF triplestores the preferred choice for managing highly interconnected data. They are often called semantic graph databases, and they are also capable of not only handling powerful semantic queries, but also inferencing for uncovering new information out of the existing correlations. Based upon these benefits, the semantic model web service utilizes its RDF triplestore instance, which is capable of persisting data from two different data sources and semantic models distinctly.
- **Data transformation into a unified model**: In order for the RDF triplestore to be utilised, the data which have been retrieved by the other components need to be transformed in a specific format. For this purpose, the RDF model is used, which is a standard for describing web resources and data interchange, developed and standardized with the World Wide Web Consortium (W3C). By transforming the data into RDF triples, heterogeneous data from various domains can be interlinked easily, and these interconnections ensure the timely delivery of proper information.
- **Semantic model enrichment**: Another key aspect of the semantic model web service is the need to handle two different semantic models. On the one hand, the Situation Awareness ontology is developed and documented in D3.2 and on the other hand the ontology for Building data has been incorporated, which is composed of different concepts and correlations. This web service has been designed in a way where both ontologies can be utilised effectively and without conflicts, due to the intervention of a specific component which is going to be presented in section 4.2.3.

[4] https://en.wikipedia.org/wiki/Representational_state_transfer

## 4.2 Architecture

The scope of the semantic model web service is to ensure the timely delivery of proper information flow for the building-related and SA-related data that will be retrieved and properly interconnected in the context of the project.

The semantic model web service consists of three subcomponents, namely the Semantic Databases, the Data Transformation Handler and the Semantic Data Model Interceptor. Due to the fact that the Situation Awareness model is documented in D3.2, the focus in terms of the architecture and the data which will be handled by the semantic model web service, mainly lies on the Building semantic model and how the respective data can be retrieved. To that end, not only the components of the semantic model web service will be presented, but also the other relevant Search and Rescue components and data sources which can feed the stored semantic models accordingly.

In brief, as also depicted in Figure 4-1, the main functional parts which compose the overall architectural design of the semantic model web service are the following:

- The **Backend service** that is composed of Data Transformation Handler and the Semantic Data Model Interceptor. Both components will be analysed in the following subsections and constitute the core business logic of the semantic model web service. The aforementioned components will be described in sections 4.2.2 and 4.2.3 respectively.
- The **Semantic Models** where the correlations and the knowledge-based information of each domain are stored. In the context of the project, two separate semantic models are stored and more information will be provided in section 4.2.3 respectively. On the one hand there is the Situation Awareness model which has been documented in D3.2 and on the other hand, the Building semantic model that is documented in the present deliverable.
- The **Communication with the EMS** which will establish the utilisation of both semantic model through CONCORDE platform.

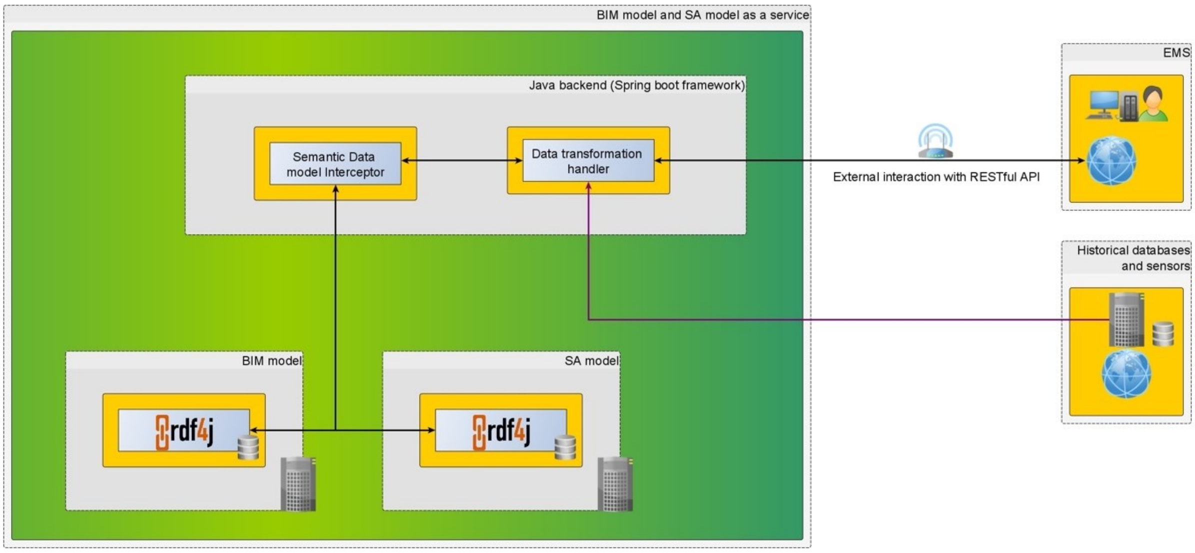


**Figure 4-1: Semantic Model Web Service Architecture**

In addition to the core functional parts and the components, the specific technological approach which constitutes the backbone of the web service will also bedescribed. More specifically, the Building ontology (just like the SA ontology) can be provided as an endpoint with Spring boot[5] REST at its backend and written in Java. The Spring Framework[6] is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE[7] (Enterprise Edition) platform. The Spring Framework is open source. As a result, the provided backend is compatible with state-of-the-art technological stacks and offers a reliable way to extend its functionalities, as the requirements of the project evolve.

Due to the semantic representation of the semantic models and, more specifically, of the Building semantic model, a database with the traditional paradigm of tables (e.g., MySQL[8], PostgreSQL[9] etc.), or other paradigms such as key-value stores (e.g., Redis[10]) cannot properly represent semantic relations, especially when taking into consideration the temporal nature of data, as is the case in Search and Rescue. The nature of semantic data is much more fitting to a graph-based format, where nodes represent entities and the edges linking them represent relations between them. As such, the most proper paradigm is triple stores, which host serialized RDF[11] data. Moreover, triple stores include powerful inferencing engines and thus offer indirect data extraction, an ability which could be utilized to discover critical information hidden in the existing data.

The Resource Description Framework (RDF) is a family of World Wide Web Consortium[12] (W3C) specifications originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information which is implemented via web resources, using a variety of syntax notations and data serialization formats. It is also used in knowledge management applications. A triple store or RDF store is a purpose-built database for the storage and retrieval of triples through semantic queries. A triple is a data entity composed of subject-predicate-object.

There are a few triple store implementations available, but for the purposes of the project, RDF4J[13] is currently used. Originally known as OpenRDF Sesame, it is one of the very first triple stores, it has thorough documentation and support for Java[14], while it offers good performance and flexibility due to its mature implementation.

### 4.2.1 Semantic Databases

As it was previously mentioned, the instances and their correlations of two separate semantic models will be stored in RDF triplestores, or alternatively in a graph database. A graph database uses graph structures for semantic queries with resources, and properties to represent and store data. The graph correlates the data items in the store to a collection of classes and properties, where properties

[5] https://spring.io/projects/spring-boot
[6] https://spring.io/projects/spring-boot
[7] https://www.oracle.com/java/technologies/java-ee-glance.html
[8] https://www.mysql.com/
[9] https://www.postgresql.org/
[10] https://redis.io/
[11] https://www.w3.org/RDF/
[12] https://www.w3.org/
[13] https://rdf4j.org/
[14] https://www.java.com/

represent the relationships between the classes. The relationships allow data in the store to be linked together directly and, in many cases, retrieved with a single operation. Taking all these into account, in the context of S&R, each semantic model will be represented by its own graph, where the resources and the relevant correlations will be defined and stored distinctly.

### 4.2.2 Data Transformation Handler

The Data Transformation Handler is the communication channel among the data sources and the respective data which will have been received from CONCORDE or other S&R applications, and the semantic models which are stored in the semantic model web service. This component is responsible not only for handling the data which are going to be retrieved, but also contains the functionality which transforms the data in a unified format (RDF) in order to be stored in the actual database instances properly. Consequently, the Data Transformation Handler creates the correlations among the resources and these correlations are subsequently stored into the individual graphs.

### 4.2.3 Semantic Data Model Interceptor

The semantic model web service, as it has already been pointed out, aims at exposing data from two separate semantic models. To this end, both the gathered and the stored data must be handled properly in order for the proper semantic model to be used during the data exchange. The Semantic Data Model Interceptor will ensure that the proper model will be utilised, based on some predefined rules, which will be triggered each time a data request or a response is required by the other S&R components.

### 4.2.4 EMS and Data Retrieval

**COncORDE** is a cloud-based platform consisting of a wide set of services, such as incident user, organization, notification among others, which are dedicated in the management of a crisis operation. The COncORDE is built in a Django Framework with the usage of Python. Moreover, in order to connect to external services, apps etc., the Django REST framework was involved in the development of the platform. The Django REST framework is a powerful and flexible toolkit for building Web APIs. For this reason, all data of COncORDE's services can be consumed via HTTP requests to the COncORDE platform API.

A short instance of a request can be the following: GET */api/*incidents/incident/{id}

Through this request, the user can retrieve data for a specific incident (e.g., "fire in a building") as well as details such as the name of the incident, the type of hazard (e.g., earthquake), the available number of Ems units and more.

Apart for the COncORDE platform, the data lake ecosystem also exists, which handles data coming from technologies used in the field, such as sensors, smartwatches, AR glasses and more. During the development of the S&R project, these types of data aim to be part of COncORDE's notification service and get consumed by partners with HTTP requests, such as GET */api/*incidents/incident/7/notifications/ (this request asks notification data on incident 7). COncORDE's notification service indicates any unusual behaviour coming from the field such as unusual environmental temperature, first responders' heart rate beyond the normal limits, new decisions from SOT DSS services (e.g., send 3 EMS_UNITS to Incident 7 with location etc.) and more.

Moreover, there is the possibility to immediately ask for the requested type of data from the data lake ecosystem, especially if there is the need for more specific data, such as building data. This can be achieved via an HTTP request in the HDFS with the help of the WebHDFS API. The data lake ecosystem mainly consists of Apache Hadoop, Apache Yarn and Apache Spark. Apache Hadoop provides a high-performance native protocol for accessing the HDFS. While this is great for Hadoop applications running inside a Hadoop cluster, users often want to connect to the HDFS from the outside. For example, some applications must load data in and out of the cluster, or to interact with the data stored in HDFS from the outside (as is the case for the S&R project). As a result, in order to interact with Hadoop's data and access Data Lake, the HDFS provides an additional protocol (using an industry standard RESTful mechanism), called WebHDFS.

**WebHDFS** defines a public HTTP REST API, which permits clients to access Hadoop from multiple languages without installing Hadoop. As a result, the request can be similar to get "http://host:port/webhdfs/v1/path_with_the_requested_data", where:

- "host" is the server's ip where Apach Hadoop is installed,
- "port" is the port which it listens to (9870 in the current phase of the project)
- "path_with_the_requested_data" is the actual path where the required data are stored in HDFS.

It is critical to mention that the above example describes the retrieval of data from the data lake ecosystem by the requested components. However, this example surpasses the data ingestion part, the part where the data generator (e.g., a field device which produces data) connects to the integration platform and then the data lake ecosystem retrieves the data, in order to aggregate them and finally prepare them to be sent to the rest system. This is part of other components with specified deliverables and demonstrators which interact with the data lake ecosystem, in order to provide data to it.

As a result, the data lake ecosystem will have data from the field, stored into HDFS and specific directories (e.g., /building_data), as it was mentioned before.

Last, but not least, another option of the data lake ecosystem is the Spark Streaming component, which retrieves data from a variety of sources, such as Kafka topics, makes the requested aggregations and then sinks the output to specific destinations (such as COncORDE's database, HDFS or a new Kafka topic). This alternative option will be further exploited after the first integrated platform release at M16.

# 5 BIM Service

## 5.1 Overview

The purpose of the BIM Ontology is to represent the main spatial topological elements of a building. This model, in combination with the SA model, acts as the core module within the semantic model web service, where knowledge-based information is manipulated and delivered in order for the first responders to have a clear view of the incident itself. The model is constructed as an extension of the BOT ontology that provides the vocabulary to describe the topology of a building as well as the relationships between their main components such as storey, zones, spaces, and building elements.

## 5.2 BIM Semantic Representation

The most important concepts related to the Building domain, based on the literature review and the needs of the Search and Rescue project are the following:

- **Building:** Represents a structure which provides shelter for its occupants or contents and stands in one place. The building is also used to provide a basic element within the spatial structure hierarchy for the components of a building project (together with site, storey, and space).
- **Storey:** The building storey has an elevation and typically represents a (nearly) horizontal aggregation of spaces which are vertically bound.
- **Space:** A space represents an area or volume bounded actually or theoretically. Spaces are areas or volumes which provide certain functions within a building.
- **Zone:** A part of the physical world or a virtual world which is inherently both located in this world and having a 3D spatial extent.
- **Apartment:** A self-contained housing unit which is composed by several spaces.
- **Element:** An element is a generalization of all components which make up an AEC product.

The topological element can be considered as one of the following:

- building: Building
- building: Storey
- building: Apartment
- building: Space.

Each of these elements should have an identifier which uniquely identifies them in the model and corresponds to the IfcGlobalId attribute in that schema. The model builds around the root concept building: Building, which incorporates attributes to indicate its physical location by means of geospatial coordinates (wgs84_pos:long, wgs84_pos:lat, and wgs84_pos:alt), as provided by the W3C geo vocabulary. The relationships between the topological elements are taken directly from BOT. A building may have several stories, which may in turn contain one or more residential apartments. These relationships are materialized by the object properties bot: hasStorey and building: hasApartment respectively.

An apartment has different spaces such as a living room, a kitchen or bedroom, and these spaces are comprised by building components which delimit the zone itself and can be summarized as follows:

- building: Door

- building: Slab
- building: Window
- building: Wall

## 5.3 Data Exposure

In order to maximize the utilization of data resources, the compatibility with standards and to reduce the risk of errors by converting data into multiple formats, all data requests will rely on JSON[15] format. One additional reason that JSON was chosen, is because an extension of it, JSON-LD[16], is entirely based on semantic representations, and therefore, it offers us the ability to exchange even semantically annotated data with the represented information.

JSON-LD (JavaScript Object Notation for Linked Data) is a method of encoding linked data using JSON. One goal for JSON-LD was to require as little effort as possible from developers to transform their existing JSON to JSON-LD. It allows data to be serialized in a way that is similar to traditional JSON. It was initially developed by the JSON for Linking Data Community Group before being transferred to the RDF Working Group for review, improvement, and standardization, and is currently maintained by the JSON-LD Working Group. JSON-LD is a World Wide Web Consortium Recommendation and is also openly supported by Google.

In order to support the storage of dynamic data structures representing vital information, and maintaining compatibility with RDF standards, we serialize/store them in XML[17] format. In RDF terms, such information is an XMLLiteral[18], a string containing a serialized XML structure. When such information is required, it is offered in JSON format as a whole, while the receiver can get the XML data and deserialize them, perhaps even converting them back to JSON as well, in order to successfully process them. However, it should be noted that depending on the structure of the original JSON data, it might not be possible to convert into XML at all. This is particularly true for fields which start with a number (XML does not allow this), as well as array fields, because XML cannot (automatically) name them as elements.

In the current section and, more specifically, in both Table 5-1 and Table 5-2 sample Endpoint examples towards the manipulation of the data are provided.

**Table 5-1: GET API call Example**

| | |
|---|---|
| Description | **GET** /buildingData<br>Returns the building data for the specified parameters. |
| Parameters | buildingId: The building identifier for which the caller requires data for. |

[15] https://www.json.org/

[16] https://json-ld.org/

[17] https://www.w3.org/XML/

[18] https://www.w3.org/TR/owl-test/misc-200-xmlliteral

| | |
|---|---|
| | timestampStart: A start for a range to filter building data timestamps. Useful for performance and getting updated status on S&R operations. Optional.<br>timestampEnd: An end for a range to filter building data timestamps. Useful for performance and getting updated status on S&R operations. Optional. |
| Body | *None* |
| JSON Response Message | {“buildingId”: “21332564”, “buildingData”:”<Serialized XML information>”, “timestamp”: 12345324} |
| Input Example | **GET** /buildingData?**buildingId=**21332564 |

**Table 5-2: POST API call Example**

| | |
|---|---|
| Description | **POST** /notification<br>Posts a new notification message, alongside the metadata it carries. |
| Parameters | *None* |
| Body | {“buildingId”: “1124”, “data”: ”<Serialized XML information>”, “timestamp”: 2123245345} |
| Response Message | Request status (200 for successful incorporation to the triple store, or error code in case of an error) |
| Input Example | **POST** /notification |

# 6 Conclusion

The current deliverable documents the first iteration of the semantic model web service, through which both the Building semantic model and the Situation Awareness semantic model will be exposed. In this context, the working methodology in designing the appropriate web service and Building semantic model was described, while existing state-of-the-art approaches to BIM semantic models and relevant ontologies were presented. Furthermore, the document focused on the architecture and how the semantic model web service will fulfil specific requirements by communicating with the Emergency Management System and other applications, if needed. Subsequently, the components and the main functional parts of the web service were presented in detail. Finally, an initial high-level specification of the Building semantic model is also provided and it will be the basis towards its final specification until the end of the T3.2 on M30.

# Annex I: References

[1] Hor, Abdel-Hadi. (2015). A semantic Web platform for BIM-GIS integration. 10.13140/RG.2.1.4176.6643.

[2] Hor, Abdel-Hadi & Jadidi, Mojgan & Sohn, Gunho. (2016). BIM-GIS INTEGRATED GEOSPATIAL INFORMATION MODEL USING SEMANTIC WEB AND RDF GRAPHS. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences. III-4. 73-79. 10.5194/isprsannals-III-4-73-2016.

[3] Matějka, Petr & Tomek, Aleš. (2017). Ontology of BIM in a Construction Project Life Cycle. Procedia Engineering. 196. 1080-1087. 10.1016/j.proeng.2017.08.065.

[4] Zhong, Botao & Gan, Chen & Luo, Hanbin & Xing, Xuejiao. (2018). Ontology-based framework for building environmental monitoring and compliance checking under BIM environment. Building and Environment. 141. 10.1016/j.buildenv.2018.05.046.

[5] Niknam, Mehrdad & Karshenas, Saeed. (2017). A shared ontology approach to semantic representation of BIM data. Automation in Construction. 80. 22-36. 10.1016/j.autcon.2017.03.013.

[6] Liu, Xin & Wang, Xiangyu & Wright, Graeme & Cheng, Jack & Li, Xiao & Liu, Rui. (2017). A State-of-the-Art Review on the Integration of Building Information Modeling (BIM) and Geographic Information System (GIS). ISPRS International Journal of Geo-Information. 6. 53. 10.3390/ijgi6020053.