Egyptian Informatics Journal 23 (2022) 447–467

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

A method for transferring BIM data into domain ontologies: A case study based on airport services

J.J. Herrera-Martín [a], I. Castilla-Rodríguez [a], E.J. González [a], N. Martín-Dorta [b]

[a] Departamento de Ingeniería Informática y de Sistemas 4 Universidad de La Laguna, Avda. Astrofísico Fco. Saánchez s/n, 38200, AP 456. La Laguna, Canary Islands, Spain
[b] Departmento de Técnicas y Proyectos en Ingeniería y Arquitectura 7 Universidad de La Laguna, C. Padre Herrera s/n 38200, AP 456. La Laguna, Canary Islands, Spain

ARTICLE INFO

Article history:
Received 3 February 2022
Revised 3 April 2022
Accepted 10 April 2022
Available online 29 April 2022

Keywords:
Airport facilities
BIM
Ontologies
Semantic Web

ABSTRACT

Building Information Modeling (BIM) has revolutionized the construction industry as a platform for core integrated design, modeling, asset planning, and collaboration. Although BIM simplifies the retrieval and use of information in construction projects, BIM tools use native formats that pose challenges for data reuse and exchange. This article proposes a method for mapping BIM data into sets of concepts of a specific domains, and the use of Semantic Web tools for the exchange of data and information. Domain ontologies are a widely used tool for the formal definition of a field of knowledge, which facilitates the exchange of information in heterogeneous systems through technologies such as the Semantic Web. The proposed mapping can be used to enrich information and improve data integration in systems based on semantic tools that manage services or maintain facilities and building infrastructures. The article also presents a case study based on the management of airport facilities to illustrate the practical application of the method.

1. Introduction

According to the definition of the Technical Report of the European Commission of the Joint Research Center (JRC), Building Information Modeling (BIM) is a digital tool for central integrated design, modeling, asset planning and cooperation, which provides all interested parties with a digital representation of the characteristics of a building throughout its life cycle [1]. Over the last few years, the popularity of BIM has increased, and its promotion by organizations and public administrations has led to a widespread adoption of BIM among professionals in architecture, engineering, construction and operation (AEC/O) [2]. A recent Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions, points out that EU will provide a recommendation to promote BIM in public procurement for construction and provide a methodology to public clients to conduct cost-benefit analysis for the use of BIM in public tenders. Digital industrial platforms will allow stakeholders to collect and make better use of this data [3].

Although applications should easily retrieve, exchange and use the information contained in BIM models, the fact is that most BIM-based software tools store information in native formats, posing challenges for reusing data in the life cycle of projects. In this sense, the Industry Foundation Classes (IFC) data model schema, developed and maintained by BuildingSmart International, tries to overcome these difficulties and support interoperability within the industry AEC/O [4]. The IFC schema is rich and extensive, but it is also complex and redundant due to the need to represent objects and relationships for a wide range of AEC/O subdomains [5]. Therefore, software developers require clear guidance to deal with the multiple ways to represent any given semantic concept which, in turn, complexifies the development of exportation and importation interfaces [6]. More specifically, compatibility of software is challenging, since not all of them are fully compatible with all IFC schemas or with all functionalities (for example, a number of applications allow importing IFC files but not exporting them). Besides, IFC files obtained from different design software tools may present interoperability problems in terms of information

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2022.04.002
1110-8665/

on the models, types of objects or properties of objects, among others [7].

In recent years, the use of Semantic Web [8] technologies has increased notably in AEC/O areas, generally being considered complementary to BIM software [9]. Semantic Web comprises tools that codify knowledge for a true semantic interoperability in the exchange of information between computer applications. Among such tools, ontologies stand out. Ontologies are formal models that include the definition of the concepts and relationships for a domain. The utilities of ontologies include, among other applications, the retrieval of information [10,11], the development of data models [12], and the implementation of search engines [13]. In the field of AEC/O, several contributions stand out in this research line: the semantic enrichment of IFC files [14], the exchange of BIM information through Semantic Web tools such as Linked Data [15], and the development of ontologies for the BIM domain such as IFC OntoStep [16,17] and ifcOWL [18]. Along these lines, we find a number of contributions that use ontologies to facilitate the interoperability and the mapping of information from BIM models into different business processes, such as: energy management [12,19], emergency management [20], indoor navigation [21], urban management [22], event management [23], and performance evaluation [24], among others. However, for many areas there is still no standard for information exchange between BIM and semantic tools, with the result that several AEC/O sectors are not exploiting the full potential of Semantic Web [25].

This article is positioned to leverage the strength and popularity of BIM information exchange methods and Semantic Web-based software applications. Hence, we propose a method to map BIM information into sets of concepts of a specific domain that can be reused by applications. This way, domain ontologies are enriched through the mapping of IFC entities and attributes to domain classes and properties. By means of this method, it is possible to give semantic meaning, within the domain context of a service or infrastructure to be managed, to the available BIM model and thus facilitate its retrieval, exchange and use within the application area. Although the proposed method is intended to be generally applicable, the authors highlight its usefulness in the area of Facility Management (FM). Among the disciplines of AEC/O, FM is in charge of managing the operational phase of buildings or infrastructures and their services to guarantee the functionality, safety, comfort and integration of people, places and technologies [26,27]. In this context, the use of semantic tools such as domain ontologies [28,29] and BIM is beneficial [30,31].

To illustrate the method, a case study based on the management of airport facilities is presented. Airports are large infrastructures in continuous maintenance and growth. The use of the information contained in the BIM models could help us to enrich the domain ontologies for the management of airport operations such as arrivals, departures or the allocation of resources such as stands, boarding gates, etc., and, in turn, collaborate with the management of the maintenance or remodeling activities of the facilities.

This article is organized as follows: Section 2 provides a theoretical introduction to the concepts on which the article is based, such as BIM and Semantic Web, among others; Section 3 details an outline of the proposed method; and Section 4 presents a case study of the method based on an airport domain.

2. Background

Traditionally, AEC/O projects have required experts in different domains to exchange information stored in heterogeneous file formats. Hence, the integration of information about a building element required manually extracting and combining data from several project documents [25]. This situation has changed with the popularization of BIM.

One of the benefits of the widespread use of BIM has been the growing exchange of "semantic" information about buildings or facilities, highlighting the use of the IFC open standard, defined by ISO 16739 [32] and developed and maintained by BuildingSmart International [4]. IFC is a common data exchange format within the AEC/O industry [33], and offers a neutral environment for interoperability, by providing a comprehensive specification of information throughout the life cycle of an AEC/O project [34].

IFC uses the EXPRESS language to represent geometry, relationships, processes and materials, performance, manufacturing and other properties, necessary for design and production [35]. EXPRESS is not based on formal semantics, which makes it difficult to apply generic methods of reasoning and consultation in IFC construction models and poses a challenge for the exchange of well-defined information between the different actors of the product life cycle [36]. Conversely, some authors such as Barbau et al. [17] highlight that the use of Semantic Web tools for the formalization of ontologies such as the language OWL (Ontology Web Language) offer interesting modeling advantages in the representation of knowledge and the exchange of semantic data. A semantic representation of a building information model would provide anyone involved in a project with a suitable framework to easily express and combine the information about building elements [25].

The benefits of using OWL formats have arisen the interest of multiple authors, who have contributed on the transformation of EXPRESS schemas to OWL formats [37], and on the definition of ontologies such as OntoStep [16,17] or ifcOWL [18]. OntoSteep proposes general conversion procedures for any EXPRESS schema into an OWL ontology [16,17]; whereas ifcOWL [18] is specific for IFC, and it is supported by BuildingSMART. Additionally, Pauwels and Terkaj propose a specific conversion procedure for IFC schemes, based on the ifcOWL ontology [36]. An implementation of their proposal is available in a public repository [38]. However, both OntoStep and ifcOWL schemes are still complex enough as to hinder their direct usage by information systems of specific domains. An intermediate mapping mechanism (as it is proposed in this article) between the IFC file and the semantic data on the specific domain would contribute to simplify their interoperation, and enhance the extraction of knowledge.

3. Transferring IFC-based BIM data into domain ontologies

As stated in the introduction, many organizations already have at their disposal ontology-based information systems and BIM-based models of their infrastructures. Nevertheless, such organizations find difficulties to jointly exploit this knowledge in a comprehensive and operative manner. The rest of this section explores a method to combine these sources of information and improve the management of the facilities or services offered in the infrastructures. It is noted that the proposed method could be used in a facility management system or any other service management system based on semantic tools and that requires facility information provided by BIM.

Fig. 1 illustrates the main stages of the proposed method, which defines a mapping procedure between IFC entities and domain classes using the ifcOWL ontology [18] as an intermediate element. First, a mapping file is defined based on the characteristics of the domain ontology and the construction project. This file defines the relationships between IFC entities and domain ontology classes. The second step uses the construction project to classify the elements based on the IFC entities and types, as well as defining the custom IFC properties that will be used in the transfer of BIM information to the domain ontology. The third step uses the

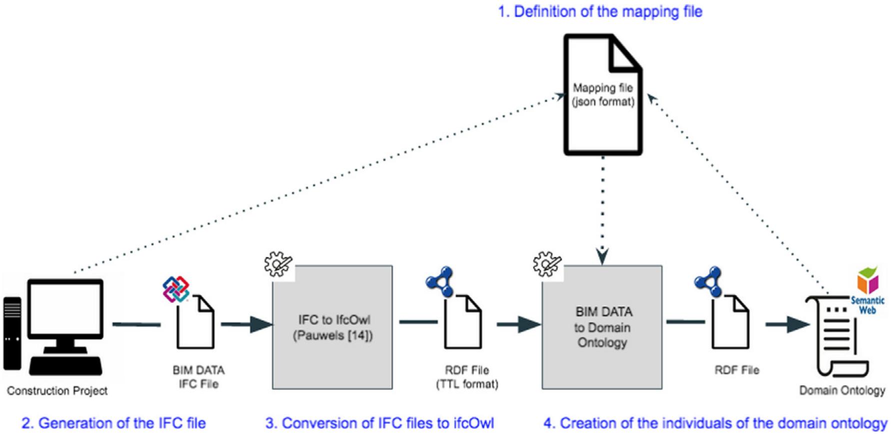
1. Definition of the mapping file
Mapping file (json format)
Construction Project
BIM DATA IFC File
IFC to IfcOwl (Pauwels [14])
RDF File (TTL format)
BIM DATA to Domain Ontology
RDF File
Semantic Web
Domain Ontology
2. Generation of the IFC file
3. Conversion of IFC files to ifcOwl
4. Creation of the individuals of the domain ontology

Fig. 1. General scheme of the exchange method. Solid lines indicate the mapping process flow; while dotted lines represent data flows for the configuration and use of the mapping file.

EXPRESS-to-OWL conversion tool proposed by Pauwels and Terkaj [36] to obtain an instance of the ifcOWL [18] ontology from the IFC file. The last stage uses this ontology to perform a process of extracting the IFC elements and attributes associated with the domain ontology classes, based on the definition of the mapping file. The following subsections describe each stage in more detail, by paying special attention to the first and last ones.

3.1. Definition of the mapping file

The first stage involves studying the characteristics of the domain ontology and the information of the BIM model of the construction project. Based on those characteristics, we define the mapping file (section S4 of the technical appendix provides a detailed description of the structure of the mapping file proposed in the method).

The mapping file contains two different blocks: the first block includes the global parameters that define general aspects used in the mapping of all domain classes; the second block defines a set of specific parameters for each domain class. The specific parameters are divided into two groups, one that defines the mapping between the domain class itself and the associated IFC entity by means of a regular expression, and another to define the mapping of the properties of the domain class. The first group of parameters ensures a correct mapping between objects of the IFC entity and individuals of the domain class. The *ifcAttributeMapping* global parameter defines the IFC attribute used to match the regular expression for all domain classes. The regular expression for each domain class is defined by using the *patternMapping* parameter.

Table 1 presents a summary of the main parameters, both global and specific to each domain class, that are included in the mapping file.

The method proposes four types of mapping for properties:

- *IfcMappingProperties* relate properties of domain classes with IFC attributes declared directly in the definition of the IFC entity mapped with the domain class.
- *customMappingProperties* define the mapping of custom attributes of IFC entities defined in the BIM software to properties of domain classes.
- *domainIndClassReferences* define the mapping of a domain class property with the reference to one of the domain class individuals mapped in the input IFC file mapping process.
- *newIndDomainClasses* relate a property of the domain class with the reference to a new individual of a domain class that is not included in the mapping file. In turn, for these new individuals, mappings of the newIndDomainClassProperties type can be defined to associate custom IFC attributes with properties of the new individual's domain class.

Table 1
Parameters of the mapping file.

Name	Description
	Global parameters
ifcSchema	IFC schema associated with the input IFC file
iriBaseDomainOntology	Base Internationalized Resource Identifier (IRI) of the domain ontology
ifcAttributeMapping	IFC parameter used in the regular expression to check the mapping of the domain classes
domainRdfLanguageFile	Format of the output RDF file that defines the domain individuals
	Domain class definition
domainClassName	Name of the domain class
ifcEntity	IFC entity that maps onto the domain class
patternMapping	Regular expression to check the mapping of the domain class
ifcMappingProperties	Definition of mappings of the ifcMappingProperties properties from the domain class
customMappingProperties	Definition of mappings of the customMappingProperties properties from the domain class
domainIndClassReferences	Definition of mappings of the domainIndClassReferences properties from the domain class
newIndDomainClasses	Definition of mappings of the newIndDomainClasses properties from the domain class

The values of the properties associated with mappings of type *ifcMappingProperties*, *customMappingProperties*, and *newIndDomainClassProperties* can have different data types. To define the type associated with these mappings, an attribute called *ValueType* is proposed, which can take four values:

- *Simple*: the property will take the value of a simple data type, based on the type conversion from IFC to OWL data, as in Pauwels and Terkaj [36]
- *RefOntologyDomainIndClass*: the property value is the reference to an individual in the domain ontology

- *RefifcOWLIndClass*: the property value is the reference to an individual in the ifcOWL ontology generated from the input IFC file.
- *Function*: the property value will be obtained from the function defined in the functionName attribute. The result of this function will be a simple type, or the reference to an individual.

BIM provides us with entities of type *IfcRelDefinesByProperties*, which define the relationship among properties and IFC objects. We use these entities to map properties of type *customMappingProperties*, *domainIndClassReferences* and *newIndDomainClasses*.

3.2. Generation of the IFC file

The second stage of the method is developed within the framework of the BIM modeling software. We classify the elements that make up the construction project based on IFC entities and types. Additionally, we create custom IFC properties for those elements that do not have a defined BIM classification. The mapping file generated in the previous stage drives this classification, which will be used to transfer BIM information to the domain ontology. The result of the classification is exported as a file in IFC format.

3.3. Conversion of IFC files to ifcOWL

As stated in Section 2, the ifcOWL ontology is a connection point between the IFC standard and Semantic Web technologies. ifcOWL facilitates the exploitation of IFC data through Semantic Web technologies in terms of data distribution, data model extensibility, queries and reasoning. To obtain the ifcOWL equivalent of an IFC file, Pauwels proposes an EXPRESS-to-OWL conversion tool [36], which allows us to obtain an instance of the ifcOWL ontology in RDF-Turtle format from an input IFC file. Hence, step 3 of our method consists of obtaining the ifcOWL instance from the IFC file obtained in the previous step by means of the EXPRESS-to-OWL conversion tool [36].

3.4. Creation of the individuals of the domain ontology

The last stage of the method uses both the mapping file and the ifcOWL ontology created in the previous stage to generate the individuals of the domain ontology that we require. Besides, the properties associated with each individual are defined. Fig. 2 summarizes the main tasks performed during this stage.

We start by loading the ifcOWL ontology model created in the previous stage. The ontology model allows us to make queries and obtain the information of the classes, individuals and properties of the ifcOWL ontology associated with the input IFC file.

From the combination of the ifcOWL model and the mapping file, we obtain a list of domain individuals. Algorithm 1 describes this process: we create a new individual of the corresponding domain class for each object of the IFC entities that matches the mapping pattern defined in the mapping file.

Algorithm 1: Creation of a new individual

```
MapFile = mapping file
ifcOWLOnt = ifcOWL ontology model
listIndDO = empty list of individuals from domain ontology
for ifcEntity: get IFC entity referenced in mapFile do
  for ifcObject: get object of ifcEntity from ifcOWLOnt do
    if ifcEntity.patternMapping matches with ifcObject.
  ifcAttributeMapping then
      indDO = Create new individual from ifcObject
      Add indDO to listIndDo
      for ifcProp: get ifcMappingProperties related to
  ifcObject
        Add ifcProp to indDO
      end for
    end if
  end for
end for
```

The pseudocode described in Algorithm 1 incorporates properties of type *IfcMappingProperties* to the created individuals. In a subsequent step (Algorithm 2), properties of type *customMappingProperties*, *domainIndClassReferences*, *newIndDomainClasses* and *newIndDomainClassProperties* are incorporated. Such properties obtain their values from ifcOWL individuals associated with *IfcRelDefinesByProperties* entities. Especifically, properties of type *newIndDomainClasses* create a new relationship between an exist-

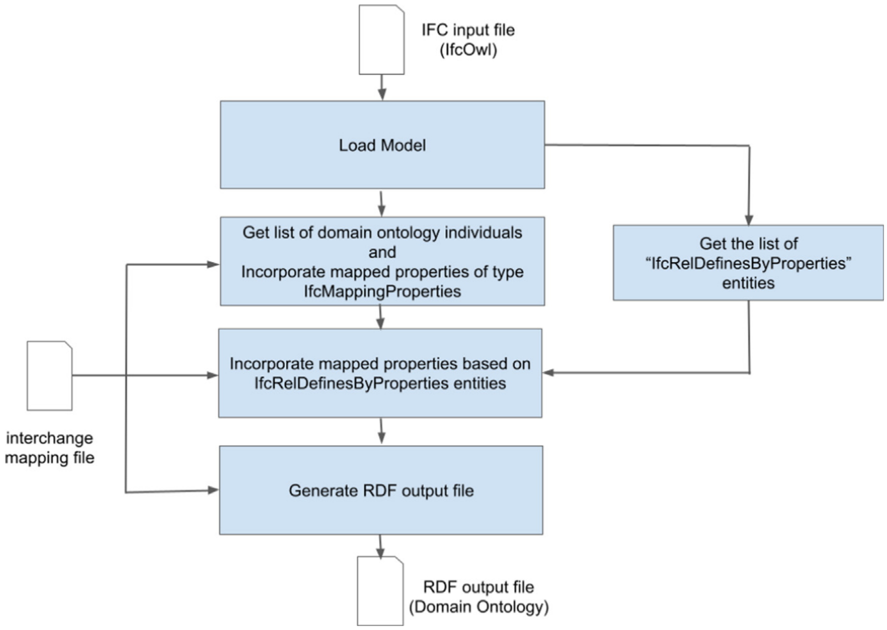
IFC input file
(IfcOwl)
Load Model
Get list of domain ontology individuals
and
Incorporate mapped properties of type
IfcMappingProperties
Get the list of
"IfcRelDefinesByProperties"
entities
Incorporate mapped properties based on
IfcRelDefinesByProperties entities
interchange
mapping file
Generate RDF output file
RDF output file
(Domain Ontology)

Fig. 2. Process diagram of phase 3.

ing individual (populated in the previous step) and a new individual by means of a property. Section S4 from the technical appendix shows further details on these steps.

Algorithm 2: Creation of new properties

```
MapFile = mapping file
ifcOWLOnt = ifcOWL ontology model
listIndDO = get list of individuals from domain ontology
  (generated according to the pseudocode in Algorithm 1)
listIcfRDBP = get list of individuals from ifcOWLOnt
  associated with the entity IfcRelDefinesByProperties
for ifcRDBP: get individual from listIfcRDBP do
  for ind: get individual from listIndDO do
    for prop: get property from ind do
      if type of prop == newIndDomainClasses then
        newInd = new Individual created from ifcRDBP
        Add newInd to listIndDO
        Add prop to ind
        ind.prop = newInd
      else
        newProp = new property created from ifcRDBP with
  the same type as prop
        Add newProp to ind
      end if
    end for
  end for
end for
```

Once the list of individuals of the domain ontology and their properties have been defined, the last step is to generate the output RFC file in the format indicated in the global parameter *domainRdfLanguageFile*.

4. Case study

We illustrate the application of the method described above with a case study based on the domain of airport operations. The case study is based on an airport whose information systems are based on a subset of the Airm-mono ontology, developed in the BEST project within the framework of the *SESAR* macroproject funded by the EU [39,40]. The only modification performed in the domain ontology is the addition of an *IfcProject* class that contains information about an IFC project.

4.1. Method implementation

As stated above, the proposed method makes it possible to map classes of the domain ontology and IFC entities; to reference individuals of classes of the domain ontology from attributes of IFC entities; and to create new individuals of domain ontology classes from attributes of IFC entities. Fig. 3 shows the scheme of the domain ontology used in this case study.

The Airport Room IFC Building of BuildingSmart International [41] is currently working on incorporating items associated with airport environments into the IFC standards. These items include, for instance, the security screening system, the automated baggage drop system or the automated border control systems. The lack of these items in the IFC standard required the explicit definition of mappings between domain classes and IFC entities (Table 2). Hence, the *Terminal* class refers to the airport terminal building, which is associated with an *ifcBuilding* entity. Gates or PassengerLoadingBridges are mapped to the *IfcBuildingElementProxy* entity, which is an abstract entity that can represent any item from a building or infrastructure. Finally, those classes associated with the spaces in the movement zone of aircrafts were mapped to the *IfcSpace* entity. In the event that future versions of the IFC standard include more specific entities for such or new items, our method allows us to easily incorporate them into the mapping file.

The mapping file was implemented using JSON [42]. Table 3 summarizes some of the most relevant global parameters defined in the mapping file, as well as the configuration parameters of the *IfcProject*, *Terminal* and *AircraftStand* classes. With regards to the mapping of properties, the case study implemented examples of the different mapping options proposed in the method, as well as functions for the treatment of IFC attributes when mapping them in properties of the domain classes. Section S5 of the technical appendix presents the complete mapping file used in the implementation of the case study.

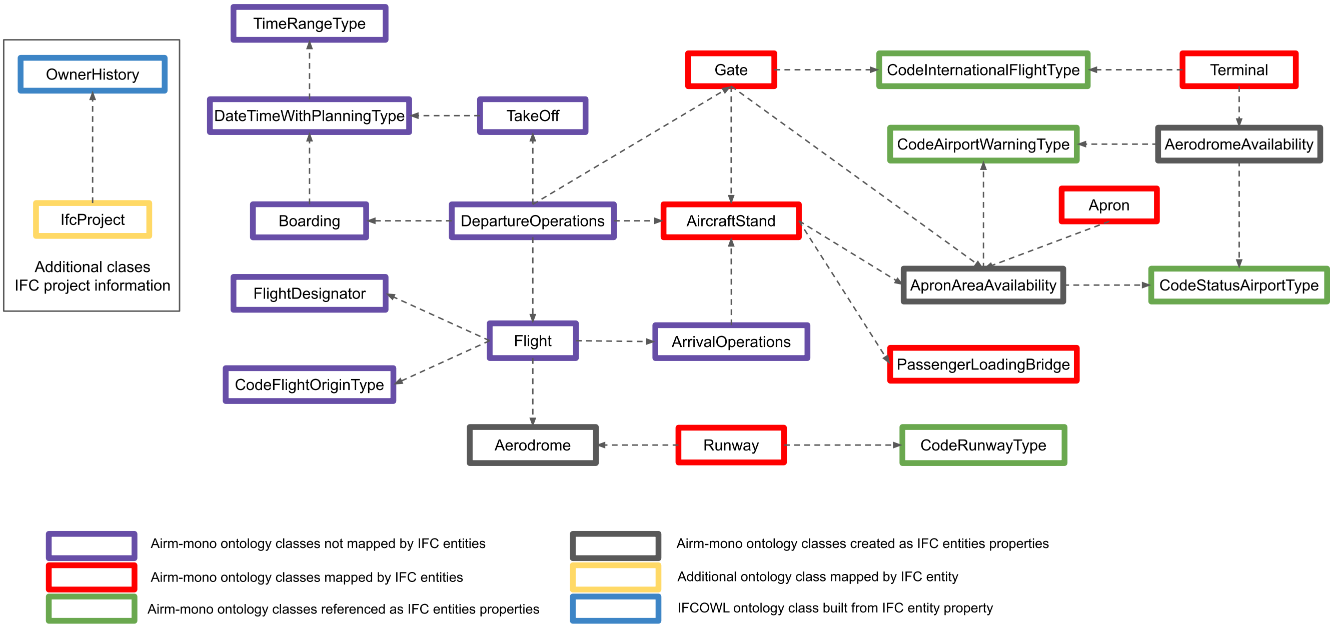

Fig. 3. Simplified schema of the mapping from a subset of the Airm-mono ontology.

Table 2
Mapping among domain classes and IFC entities.

Class Ontology Domain	IFC Entity
Terminal	IfcBuilding
Gate	IfcBuildingElementProxy
PassengerLoadingBridge	IfcBuildingElementProxy
AircraftStand	IfcSpace
Apron	IfcSpace
Runway	IfcSpace

After applying all the stages of the proposed method, we obtain a mapping between the physical elements defined in an airport construction project and the *Airm-mono* domain classes [40]. For example, a terminal building and an airport runway can be mapped, respectively, to the *Terminal* and *Runway* classes from *Airm-mono*.

The product obtained from the method proposed is an RDF file with the definition of the mapped individuals enriching the domain ontology. An example of the RDF files obtained from the implementation of the case study is presented in Section S5.2 of the technical appendix.

4.2. Example of use

By means of the proposed method it would be possible to load information associated with physical elements forming part of the objective facility. For our case study, we would be able to load or fulfill the existing information about the elements of an airport (e.g. boarding gates, stands, or fingers) from the BIM model.

Moreover, we can enrich the management information used to provide the services of the infrastructure. For example, airports are very dynamic facilities that very often suffer from maintenance or expansion works. Within this context, our method would allow managers to dynamically incorporate into the information systems of the airport the details from the BIM projects that describe the modifications being performed. Hence, for instance, we could define an *ApronAreaAvailability-operationalStatus* property from the construction project for each stand or boarding gate. This property would be associated with the maintenance or remodeling tasks, and would indicate the operatibility of the stand or boarding gate in the ontology.

The combination of the domain ontology and the BIM information allows a manager to pose their information systems queries such as *which stands are available for a domestic flight?*. This application is illustrated in Listing 1, by means of a SparQL query that combines, among other data, the *standAvailability* individuals obtained from the mapping of the IFC entities.

Listing 1: SparQL query to determine available stands

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX airm: <http://project-best.eu/owl/airm-mono#>
SELECT distinct  ?stands ?standAvailability
WHERE {
   ?gates rdf:type owl:NamedIndividual .
   ?gates rdf:type airm:Gate .
   ?gates airm:Gate-internationalType airm:
         CodeInternationalFlightType-DOMESTIC_AIR_SERVICE .
   ?gates airm:Gate-aircraftstand ?stands .
   ?stands airm:AircraftStand-availability ?standAvailability .
   ?standAvailability airm:ApronAreaAvailability-operationalStatus
         airm:CodeStatusAirportType-NORMAL
}
```

Listing 2 shows an additional example of query in the context of maintenance works. In this case, the question is *which stands, from those that are currently available, are not assigned to a flight during the time foreseen for the maintenance action?*.

Listing2: SparQL query about to determine stands not assigned to a flight

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX airm: <http://project-best.eu/owl/airm-mono#>

SELECT distinct  ?stands
WHERE {
        {
              ?gates rdf:type owl:NamedIndividual .
              ?gates rdf:type airm:Gate .
              ?gates airm:Gate-aircraftstand ?stands.
              ?stands airm:AircraftStand-availability ?
                  standAvailability .
              ?standAvailability airm:ApronAreaAvailability-
                  operationalStatus airm:CodeStatusAirportType-
                  NORMAL
        }
        MINUS {
               {
                  ?flight rdf:type airm:Flight .
                  ?flight airm:Flight-arrivalOperations ?arrival  .
                  ?arrival airm:ArrivalOperations-arrivalStand ?
                      stands .
                  ?arrival airm:ArrivalOperations-landing ?land .
                  ?land airm:Landing-time ?dtptArrival  .
                  ?dtptArrival airm:DateTimeWithPlanningType-
                      timeTolerance  ?trtArrival .
                  ?trtArrival airm:TimeRangeType-earlier ?
                      dateArrivalEarlier .
                   FILTER (?dateArrivalEarlier < "2020-01-12T09
                       :00:00"^^xsd:dateTime )
            }
               UNION  {
                   ?departures rdf:type airm:DepartureOperations .
                   ?departures airm:DepartureOperations-
                       departureStand ?stands .
```

(continued on next page)

(*continued*)

Listing 2: SparQL query about to determine stands not assigned to a flight

```
                ?departures airm:DepartureOperations-takeOff ?
                    takeoffs .
                ?takeoffs airm:TakeOff-time ?dtpttakeoff .
                ?dtpttakeoff airm:DateTimeWithPlanningType-
                    timeTolerance ?trttakeoff .
                ?trttakeoff airm:TimeRangeType-later ?
                    dateTakeoffLater
                FILTER (?dateTakeoffLater > ”2020-01-12T08:00:00
                    ”^^xsd:dateTime )
            }
        }
}
```

5. Conclusions

Both BIM and the semantic tools, such as domain ontologies, are technologies designed for the organization and exchange of information. However, for many domains there is still no standard for exchanging information between BIM and semantic tools.

In this article, we have proposed a method to define mappings between the information of the BIM models and the ontologies of a domain. Within the context of service management systems based on semantic tools, which generally require facility information provided by BIM, our approach would facilitate the exchange of enriched information and the development of computer applications.

To define the mapping between the BIM information and a domain ontology, a flexible methodology has been proposed that allows defining the mapping elements, even when there are no classifications of BIM elements defined for that domain. This methodology can be very suitable in dynamic infrastructure environments that, due to their operations, have associated maintenance, expansion or change works of their facilities. This kind of facilities comprises airports, railway stations, and ports; and they all benefit from this approach since the BIM information associated

Table 3
Most relevant parameters included in the mapping file of the case study.

Type	Name	Value
Global	ifcSchema	IFC4
Global	iriBaseDomainOntology	http://project-best.eu/owl/airm-mono#
Global	ifcAttributeMapping	name_IfcRoot
Domain class definition domainIfcProject		
domain class	domainClassName	IfcProject
domain class	ifcEntity	IfcProject
Property type ifcMappingProperties	domainPropertyName	OwnerHistory
Property type ifcMappingProperties	ifcAttributeName	ownerHistory_IfcRoot
Domain class definition terminal		
domain class	domainClassName	Terminal
domain class	ifcEntity	IfcBuilding
Property type ifcMappingProperties	domainPropertyName	Terminal-description
Property type ifcMappingProperties	ifcAttributeName	description_IfcRoot
Property type customMappingProperties	domainPropertyName	Terminal-internationalType
Domain class definition aircraftStand		
domain class	domainClassName	AircraftStand
domain class	ifcEntity	IfcSpace
Property type domainIndClassReferences	domainPropertyName	AircraftStand-passengerLoadingBridge
Property type domainIndClassReferences	referencedClassName	PassengerLoadingBridge
Property type newIndDomainClasses	domainPropertyName	AircraftStand-availability
Property type newIndDomainClasses	newDomainClassName	ApronAreaAvailability
Property newIndDomainClassProperties type ApronAreaAvailability	domainPropertyName	ApronAreaAvailability-operationalStatus

with these works can be used from the information systems that deal with the management of the operation or maintenance of these infrastructures.

Apart from the practical applications stated in this article, the mapping between the information of a BIM model and the domain ontologies could be used to enrich the Digital Twin tools that are being developed for this kind of facilities.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Technical Appendix

.

Introduction

This appendix develops a number of technical aspects treated in the article *A method for transferring BIM data into domain ontologies: A case study based on airport services*. This appendix is organized as follows: section S2 discusses the most important features of the Express language that are included in the development of the proposed method; section S3 comments on some aspects of the structure of the Industry Foundation Classes (IFC) data schema that limit the scope of the proposed method; section S4 presents technical aspects of the implementation of the mapping file; and section S5 provides a deeper insight on the case study.

Express

This section explains those characteristics of the EXPRESS language [43], referred to attributes, that must be taken into account for the proposed method.

Attributes in EXPRESS are declared within the scope of an entity, and follow a hierarchical structure. We can find *explicit*, *derived* and *inverse* attributes. Attributes establish a relationship with other data types that are declared elsewhere in the EXPRESS schema [43]. Derived attributes are especially challenging, since they depend on the content and structure of other attributes. Hence, converting them to IfcOwl requires additional specific rules, and they are not covered by Pauwels and Terkaj [36]. Pawels and Terkaj [36] do not convert neither inverse attributes in certain situations, such as when an IFC element has more than one inverse attribute defined, or when the inverse attribute has a grouping data type like LIST or ARRAY in its range. With this background, we have decided not to take into account neither the derived attributes nor the inverse attributes in our method. In any case, this problem is a possible future line of work.

IFC data schema

IFC is based on a hierarchy of entities in the form of a tree, starting from an *IfcRoot* node from which all other entities inherit, directly or indirectly [45]. IFC splits all entities into rooted and non-rooted entities. Rooted entities are derived from *IfcRoot* and have a concept of identity (they have a GUID), along with name, description, and revision control attributes. Non-rooted entities have no identity GUIDs, and instances only exist if they are referenced from a rooted instance, either directly or indirectly. *IfcRoot* is subdivided into three abstract concepts: object definitions, relationships, and property sets. In relation to object definitions, the *IfcObject* entity is a generalization of any semantically treated thing or process. The occurrences of the *IfcObject* entity are subdivided into six fundamental concepts: actors (*IfcActor*), controls (*IfcControl*), groups (*IfcGroup*), products (*IfcProduct*), processes (*IfcProcess*), and resources (*IfcResource*).

In the scope of the proposed method, all IFC entities associated with the mapping are defined as rooted, since the GUID is used in the mapping process.

In the context of defining relationships in IFC, the *IfcRelDefines* entity, which is a child entity of *IfcRelationship*, handles the link between object definitions, property definitions, and declaration context. These types of relationships are subdivided into 4 concepts:

- *IfcRelDefinesByObject*: defines the relationship between an object that participates in the decomposition of an object type and the occurrences of an object that participates in a decomposition of occurrences of that type.
- *IfcRelDefinesByTemplate*: defines the relationships between the property set template and the property sets themselves.
- *IfcRelDefinesByType*: defines the relationship between an object type and the occurrences of that object.
- *IfcRelDefinesByProperties*: defines the relationships between property set definitions and objects.

IFC properties comprise two different concepts:

- *Property sets and property set templates*: define the syntax and data types for sets of properties and individual properties.
- *Property set occurrences*: define sets of extensible and shareable properties that can be attached to object occurrences.

Our method uses the *IfcRelDefinesByProperties* type relationships for mapping the properties of IFC entities. Although the rest of *IfcRelDefines* subtypes are not taken into account within the scope of this article, we are considering their inclusion in future works.

Mapping file

The first step in our method consists of studying the characteristics of the domain ontology and the information of the BIM model of the construction project. The mapping file that we propose is divided into two blocks, one with global parameters and the other with specific mapping parameters for each of the domain classes.

The global parameter block defines the general aspects used in the mapping of all domain classes. The following global parameters are defined in the mapping file:

- *ifcSchema*: schema used in the source IFC file. For example IFC2X3 or IFC4.
- *iriBaseDomainOntology*: Internationalized Resource Identifier (IRI) base of the domain ontology.
- *iriBaseIfcOwlOntology*: IRI base of the IfcOwl ontology that is generated from the source IFC file, by means of the EXPRESS-to-OWL converter proposed by Pauwels and Terkaj [44].
- *ifcAttributeMapping*: IFC attribute used to check the mapping between IFC entities and the domain ontology classes. For the mapping to be performed, the domain class specific mapping pattern must be met.
- *domainRdfLanguageFile*: format of the RDF file of domain ontology classes individuals that offers the method as output. For example, this parameter could be Terse RDF Triple Language or RDF / XML.
- *domainNameSpacePrefixes*: list of namespaces of the domain ontology. For these namespaces, in addition to the prefix, the

associated IRI is defined. These prefixes are used in the construction of the output RDF file.
- *ifcOwlNameSpacePrefixes*: defines the list of namespaces of the IfcOwl ontology generated from the input IFC file. These namespaces are used to represent the *RefIfcOwlIndClass* type mappings (the property value is the reference to an individual from the IfcOwl ontology generated from the input IFC file).

The second block defines a set of specific parameters for each class of the ontology. These parameters are divided into two groups: one that defines the mapping between the domain class and the associated IFC entity, and another to define the mapping of the domain class properties.

For the mapping of each domain class, the IFC entity associated to the individuals of that domain class is defined in the BIM model, in such a way that the individuals of the IFC entity map an individual of the domain class as long as it meets a mapping regular expression. The IFC attribute that is used to verify the mapping regular expression is defined in a general way for all domain classes in the global parameter *ifcAttributeMapping* and the regular expression to be matched is defined in a particular way for each domain class with the parameter *patternMapping*. In the mapping file, the following attributes are defined for the particular definition of each domain class:

- *domainClassName*: name of the domain class.
- *domainClassNameSpacePrefix*: namespace prefix of the domain class, based on the list of namespace prefixes defined by the global parameter *domainNameSpacePrefixes*.
- *ifcEntity*: IFC entity that maps to the domain class.
- *patternMapping*: regular expression to check the mapping of the domain class.

Each domain class included in the mapping can define four types of property sets: *ifcMappingProperties* , *customMappingProperties*, *domainIndClassReferences*, and *newIndDomainClasses*. Each property associated with these property sets allows different data types to be used. Such a type is specified in an attribute called *valueType*:

- *simple*: the property will take the value of a simple data type, based on the conversion of types of IFC to OWL data, as in Pauwels and Terkaj [44].
- *refOntologyDomainIndClass*: the property value is the reference to an individual in the domain ontology.
- *refIfcOwlIndClass*: the property value is the reference to an individual in the IfcOwl ontology generated from the input IFC file.
- *function*: the value of the property will be obtained from the function defined in the *functionName* attribute. The result of this function will be a simple type, or the reference to an individual.

Although each type of property set requires different parameters in the mapping file, they all share two base parameters:

- *domainPropertyName*: name of the property of the domain class from which the mapping is carried out.
- *domainPropertyNameSpacePrefix*: namespace prefix of the domain class property, based on the list of namespace prefixes defined by the global parameter *domainNameSpacePrefixes*.

Properties of type *IfcMappingProperties* map IFC attributes declared directly in the definition of the IFC entity related to the domain class. To obtain the value of the IFC attribute, a relation entity *IfcRelDefinesByProperties* is not required. For the definition of the *IfcMappingProperties*, and apart from the base parameters, the following parameters are configured in the mapping file:

- *ifcAtributeName*: name of the IFC attribute that maps the property defined in the parameter *domainPropertyName*.
- *ifcAtributeNameSpacePrefix*: namespace prefix of the IFC attribute, based on the list of namespace prefixes defined by the global parameter *ifcOwlNameSpacePrefixes*.
- *valueType*: indicates that the value of the property is represented in the IFC file.
- *functionName*: if *valueType* is *Function*, the name of the function from which the value of the property is obtained.

Properties of type *customMappingProperties* map custom attributes of IFC entities to properties of domain classes. These custom attributes are defined in the BIM model, and they allow the method to carry out simple property mappings that are not defined by default by the attributes of IFC entities. In our proposal, the value of the custom attributes is associated with an *IfcPropertySingleValue* entity, which will be part of an IfcPropertySet entity. IfcPropertySingleValue entities are related to values of simple type. Nevertheless, in case of having to relate the property with a value of a more complex type, the *IfcPropertySingleValue* entity can indicate the reference to the individual of the IFC entity that represents that type by means of the *valueType* parameter. The relationship between the mapping IFC entity and the *IfcPropertySet* is defined by means of a relationship entity *IfcRelDefinesByProperties*. In the mapping file, the *customMappingProperties* mappings include the base, *valueType* and *functionName* parameters (as defined for the *IfcMappingProperties*). Additionally, they define two extra parameters:

- *ifcPropertySetName*: value of the *name_ifcRoot* attribute of the *IfcPropertySet* entity associated with the custom property.
- *ifcPropertySingleValueName*: value of the *name_ifcRoot* attribute of the *IfcPropertySingleValueName* entity associated with the custom property.

Properties of type *domainIndClassReferences* define references to a domain class individual generated in the mapping process from the input IFC file. The configuration of this type of mappings follows the same scheme proposed for the *customMappingProperties* type mappings: the reference attribute is defined in the BIM model, and the value of the attribute is associated with an *IfcPropertySingleValue* entity, which forms part of an *IfcPropertySet* entity. The mapping file contains four new parameters for this kind of property sets:

- *referencedClassName*: name of the domain class of the individual referenced by the property.
- *referencedClassNameSpacePrefix*: namespace prefix of the individual referred to in the property, based on the list of namespace prefixes defined by the global parameter *domainNameSpacePrefixes*.
- *ifcPropertySetName*: value of the *name_ifcRoot* attribute of the *IfcPropertySet* entity associated with the reference property.
- *ifcPropertyReferencedClassId*: value of the *name_ifcRoot* property of the *IfcPropertySingleValueName* entity associated with the reference property.

The properties of type *newIndDomainClasses* define the mapping of domain class properties that refer to an individual domain class that is not included in the class mapping of the input IFC file. This type of mapping generates a new individual in the output RDF file, and a reference to the new individual is included in the property in which the *newIndDomainClasses* mapping is defined. The mapping file *newIndDomainClasses* mappings define three new parameters:

- *newDomainClassName*: name of the domain class from which the new individual will be created, which will be referenced in the property .
- *newDomainClassNameSpacePrefix*: namespace prefix of the new individual, which will be referenced in the property.
- *newDomainClassProperties*: list of properties of the new individual, which will be referenced in the property.

The properties defined in the *newIndDomainClassProperties* parameter have the same parameters and follow the same scheme of the *customMappingProperties* type properties.

Together with this article, we have created a public software repository that includes the *mapping_schema.json* file that presents the schema of the mapping file in JSON (JavaScript Object Notation) format [46].

Listing 1: Mapping file used in the implementation of the case study

```
{
 "globalParameters": {
  "ifcSchema": "IFC4",
  "iriBaseDomainOntology": "http://project-best.eu/owl/airm-mono#",
  "iriBaseIfcOwlOntology": "http://linkedbuildingdata.net/ifc/resources20200902_001806/",
  "ifcAttributeMapping": "name_IfcRoot",
  "domainRdfLanguageFile": "RDF/XML",
  "domainNameSpacePrefixes": [
   {
    "prefix": "airm-mono ",
    "iri": "http://project -best.eu/owl/airm-mono#"
   },
   {
    "prefix": "extra-ifc",
    "iri": "http://jjhm.project.ull.es/owl/airm-mono#"
   }
  ]
 },
 "domainClassMappings": [
  {
   "domainClassName": "IfcProject",
   "domainClassNameSpacePrefix": "extra-ifc",
   "ifcEntity ": "IfcProject",
   "patternMapping": "(^project|^PROJECT)-[a-zA-Z0-9]*",
   "ifcMappingProperties": [
    {
     "domainPropertyName": "OwnerHistory",
     "domainPropertyNameSpacePrefix": "extra-ifc",
     "ifcAttributeName": "ownerHistory_IfcRoot",
     "ifcOwlAttributeNameSpacePrefix": "base",
     "functionName": null,
     "valueType": "RefIfcOwlClass"
    }
   ],
   "customMappingProperties": null,
   "domainIndClassReferences": null,
   "newIndDomainClasses": null
  },
  {
   "domainClassName": "Terminal",
   "domainClassNameSpacePrefix": "base",
   "ifcEntity": "IfcBuilding",
   "patternMapping" : "(^terminal|^TERMINAL)-[a-zA-Z0-9]*",
   "ifcMappingProperties": [
    {
     "domainPropertyName": "Terminal-description",
     "domainPropertyNameSpacePrefix": "base",
     "ifcAttributeName": "description_IfcRoot",
     "ifcOwlAttributeNameSpacePrefix": null,
     "functionName": null,
     "valueType": "Simple"
    }
   ],
   "customMappingProperties": [
    {
```

(continued on next page)

(*continued*)

Listing 1: Mapping file used in the implementation of the case study

```
          "domainPropertyName": "Terminal-numberOfGateRooms",
          "domainPropertyNameSpacePrefix": "base",
          "ifcPropertySetName": "dynamic_attributes",
          "ifcPropertySingleValueName": "numberofgaterooms",
          "ifcOwlAttributeNameSpacePrefix": null,
          "functionName": null,
          "valueType": "Simple"
        },
        {
          "domainPropertyName": "Terminal-internationalType",
          "domainPropertyNameSpacePrefix": "base",
          "ifcPropertySetName": "dynamic_attributes",
          "ifcPropertySingleValueName": "internationaltype",
          "ifcOwlAttributeNameSpacePrefix": null,
          "functionName": null,
          "valueType": "RefOntologyDomainClass"
        }
      ],
      "domainIndClassReferences": null,
      "newIndDomainClasses": [
        {
          "domainPropertyName": "Terminal-availability",
          "domainPropertyNameSpacePrefix": "base",
          "newDomainClassName": "AerodromeAvailability",
          "newDomainClassNameSpacePrefix": "base",
          "newIndDomainClassProperties": [
            {
              "domainPropertyName": "AerodromeAvailability-operationalStatus",
              "domainPropertyNameSpacePrefix": "base",
              "ifcPropertySetName": "dynamic_attributes",
              "ifcPropertySingleValueName": "operationalstatus",
              "ifcOwlAttributeNameSpacePrefix": null,
              "functionName": "fmAddPrefixCodeStatusAirportType",
              "valueType": "Function"
            },
            {
              "domainPropertyName": "AerodromeAvailability-warning",
              "domainPropertyNameSpacePrefix": "base",
              "ifcPropertySetName": "dynamic_attributes",
              "ifcPropertySingleValueName": "warning",
              "ifcOwlAttributeNameSpacePrefix": null,
              "functionName": "fmAddPrefixCodeAirportWarningType",
              "valueType": "Function"
            }
          ]
        }
      ]
    },
    {
      "domainClassName": "Gate",
      "domainClassNameSpacePrefix": "base",
      "ifcEntity": "IfcBuildingElementProxy",
      "patternMapping": "^GateRoom-[a-zA-Z0-9]*",
      "ifcMappingProperties": null,
      "customMappingProperties": [
        {
```

(*continued*)

Listing 1: Mapping file used in the implementation of the case study

```
       "domainPropertyName": "Gate-internationalType",
       "domainPropertyNameSpacePrefix": "base",
       "ifcPropertySetName": "dynamic_attributes",
       "ifcPropertySingleValueName": "internationaltype",
       "ifcOwlAttributeNameSpacePrefix": null,
       "functionName": "fmAddPrefixCodeInternationalFlightType",
       "valueType": "Function"
      }
     ],
     "domainIndClassReferences": [
      {
       "domainPropertyName": "Gate-aircraftstand",
       "domainPropertyNameSpacePrefix": "base",
       "referencedClassName": "AircraftStand",
       "referencedClassNameSpacePrefix": "base",
       "ifcPropertySetName": "dynamic_attributes",
       "ifcPropertyReferencedClassId": "stand"
      }
     ],
     "newIndDomainClasses": [
      {
       "domainPropertyName": "Gate-availability",
       "domainPropertyNameSpacePrefix": "base",
       "newDomainClassName": "ApronAreaAvailability",
       "newDomainClassNameSpacePrefix": "base",
       "newIndDomainClassProperties": [
        {
         "domainPropertyName": "ApronAreaAvailability-operationalStatus",
         "domainPropertyNameSpacePrefix": "base",
         "ifcPropertySetName": "dynamic_attributes",
         "ifcPropertySingleValueName": "operationalstatus",
         "ifcOwlAttributeNameSpacePrefix": null,
         "functionName": "fmAddPrefixCodeStatusAirportType",
         "valueType": "Function"
        },
        {
         "domainPropertyName": "ApronAreaAvailability-warning",
         "domainPropertyNameSpacePrefix": "base",
         "ifcPropertySetName": "dynamic_attributes",
         "ifcPropertySingleValueName": "warning",
         "ifcOwlAttributeNameSpacePrefix": null,
         "functionName": "fmAddPrefixCodeAirportWaningType",
         "valueType": "RefOntologyDomainClass"
        }
       ]
      }
     ]
    },
    {
     "domainClassName": "PassengerLoadingBridge",
     "domainClassNameSpacePrefix": "base",
     "ifcEntity": "IfcBuildingElementProxy",
     "patternMapping": "^PassengerLoadingBridge-[a-zA-Z0-9]*",
     "ifcMappingProperties": null,
     "customMappingProperties": null,
     "domainIndClassReferences": null,
```

(*continued on next page*)

(*continued*)

Listing 1: Mapping file used in the implementation of the case study

```
   "newIndDomainClasses": null
  },
  {
   "domainClassName": "AircraftStand",
   "domainClassNameSpacePrefix": "base",
   "ifcEntity": "IfcSpace",
   "patternMapping": "^Stand-[a-zA-Z0-9]*",
   "ifcMappingProperties": null,
   "customMappingProperties": null,
   "domainIndClassReferences": [
    {
     "domainPropertyName": "AircraftStand-passengerLoadingBridge",
     "domainPropertyNameSpacePrefix": "base",
     "referencedClassName": "PassengerLoadingBridge",
     "referencedClassNameSpacePrefix": "base",
     "ifcPropertySetName": "dynamic_attributes",
     "ifcPropertyReferencedClassId": "passengerloadingbridge"
    }
   ],
   "newIndDomainClasses": [
    {
     "domainPropertyName": "AircraftStand-availability",
     "domainPropertyNameSpacePrefix": "base",
     "newDomainClassName": "ApronAreaAvailability",
     "newDomainClassNameSpacePrefix": "base",
     "newIndDomainClassProperties": [
      {
       "domainPropertyName": "ApronAreaAvailability-operationalStatus",
       "domainPropertyNameSpacePrefix": "base",
       "ifcPropertySetName": "dynamic_attributes",
       "ifcPropertySingleValueName": "operationalstatus",
       "ifcOwlAttributeNameSpacePrefix": null,
       "functionName": "fmAddPrefixCodeStatusAirportType",
       "valueType": "Function"
      },
      {
       "domainPropertyName": "ApronAreaAvailability-warning",
       "domainPropertyNameSpacePrefix": "base",
       "ifcPropertySetName": "dynamic_attributes",
       "ifcPropertySingleValueName": "warning",
       "ifcOwlAttributeNameSpacePrefix": null,
       "functionName": "fmAddPrefixCodeAirportWarningType",
       "valueType": "Function"
      }
     ]
    }
   ]
   },
   {
    "domainClassName": "Apron",
    "domainClassNameSpacePrefix": "base",
    "ifcEntity": "IfcSpace",
    "patternMapping": "^Apron-[a-zA-Z0-9]*",
    "ifcMappingProperties": null,
    "customMappingProperties": null,
    "domainIndClassReferences": null,
```

(*continued*)

Listing 1: Mapping file used in the implementation of the case study

```
    "newIndDomainClasses": [
     {
      "domainPropertyName": "Apron-availability",
      "domainPropertyNameSpacePrefix": "base",
      "newDomainClassName": "ApronAreaAvailability",
      "newDomainClassNameSpacePrefix": "base",
      "newIndDomainClassProperties": [
       {
        "domainPropertyName": "ApronAreaAvailability-operationalStatus",
        "domainPropertyNameSpacePrefix": "base",
        "ifcPropertySetName": "dynamic_attributes",
        "ifcPropertySingleValueName": "operationalstatus",
        "ifcOwlAttributeNameSpacePrefix": null,
        "functionName": "fmAddPrefixCodeStatusAirportType",
        "valueType": "Function"
       },
       {
        "domainPropertyName": "ApronAreaAvailability-warning",
        "domainPropertyNameSpacePrefix": "base",
        "ifcPropertySetName": "dynamic_attributes",
        "ifcPropertySingleValueName": "warning",
        "ifcOwlAttributeNameSpacePrefix": null,
        "functionName": "fmAddPrefixCodeAirportWarningType",
        "valueType": "Function"
       }
      ]
     }
    ]
   },
   {
    "domainClassName": "Runway",
    "domainClassNameSpacePrefix": "base",
    "ifcEntity": "IfcSpace",
    "patternMapping": "^Runway-[a-zA-Z0-9]*",
    "ifcMappingProperties": null,
    "customMappingProperties": [
     {
      "domainPropertyName": "Runway-type",
      "domainPropertyNameSpacePrefix": "base",
      "ifcPropertySetName": "dynamic_attributes",
      "ifcPropertySingleValueName": "type",
      "ifcOwlAttributeNameSpacePrefix": null,
      "functionName": null,
      "valueType": "RefOntologyDomainClass"
     }
    ],
    "domainIndClassReferences": null,
    "newIndDomainClasses": [
     {
      "domainPropertyName": "Runway-associatedAerodrome",
      "domainPropertyNameSpacePrefix": "base",
      "newDomainClassName": "Aerodrome",
      "newDomainClassNameSpacePrefix": "base",
      "newDomainClassProperties": [
       {
        "domainIndPropertyName": "Aerodrome-name",
```

(*continued on next page*)

(*continued*)

Listing 1: Mapping file used in the implementation of the case study

```
        "domainPropertyNameSpacePrefix": "base",
        "ifcPropertySetName": "dynamic_attributes",
        "ifcPropertySingleValueName": "aerodromename",
        "ifcOwlAttributeNameSpacePrefix": null,
        "functionName": null,
        "valueType": "Simple"
       },
       {
        "domainPropertyName": "Aerodrome-designatorIATA",
        "domainPropertyNameSpacePrefix": "base",
        "ifcPropertySetName": "dynamic_attributes",
        "ifcPropertySingleValueName": "designatoriata",
        "ifcOwlAttributeNameSpacePrefix": null,
        "functionName": null,
        "valueType": "Simple"
       },
       {
        "domainPropertyName": "Aerodrome-country",
        "domainPropertyNameSpacePrefix": "base",
        "ifcPropertySetName": "dynamic_attributes",
        "ifcPropertySingleValueName": "countrycode",
        "ifcOwlAttributeNameSpacePrefix": null,
        "functionName": "fmGetCountryFromIataCode",
        "valueType": "Function"
       }
      ]
     }
    ]
   }
 ]
}
```

Case study

To validate the proposed method to map BIM data to domain ontologies, a case study based on the domain of airport operations was implemented. The implementation of this case study was developed in Java and the Apache Jena Semantic Web libraries [47], to facilitate integration with the EXPRESS to IfcOwl conversion tool by Pauwels and Terkaj [44]. The source code for this implementation is available in the open software repository [46]. The following sections comment on the mapping file and the RDF output file of the application of the method to the case study.

Mapping table of the case study

The case study is based on a subset of the Airm-mono ontology [48], developed in the BEST project within the framework of the SESAR macro-project funded by the EU [49], to which the *IfcProject* class has been incorporated. This class has an *OwnerHistory* attribute that contains information about an IFC project. The mapping file used in the case study is presented below. The complete mapping file used in the implementation of the case study is presented in Listing 1.

RDF file that results of the case study

The product obtained from the proposed method is an RDF file with the definition of the mapped individuals that enriches the domain ontology. In the implementation of the case study, most of the names of the individuals have been defined using the pattern < *name_ifcRoot*>-< *IfcGloballyUniqueId*> . The exceptions are the individuals generated from a property of type newIndDomainClasses: they use the pattern < *name_ifcRoot*>-< *domainPropertyName parameter*>, where *name_ifcRoot* indicates the name of the IFC object in which the property is set and *domainPropertyName* is the domain class of the new individual. For example, the individual *Stand-4-ApronAreaAvailability* is created from a *newIndDomainClasses* property of the *Stand-4* object, and has the *domainPropertyName* attribute of the domain class *ApronAreaAvailability*.

Listing 2 shows an example of the file obtained from the proposed method for the implementation of the case study.

Listing 2: Example of RDF file obtained from the proposed method for the implementation of the case study

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:extra-ifc="http://jjhm.project.ull.es/owl/airm-mono#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:airm-mono="http://project-best.eu/owl/airm-mono#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#">

 <owl:Ontology rdf:about="http://project-best.eu/owl/airm-mono#"/>

 <owl:NamedIndividual rdf:about ="http://project-best.eu/owl/airm-
      mono#Stand-2-1K3qjb7UJDHBxFfEpra42T">
    <airm-mono: AircraftStand-availability>
      http://project-best.eu/owl/airm-mono#Stand-2-
          ApronAreaAvailability
    </airm-mono:AircraftStand-availability>
    <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
        AircraftStand"/>
 </owl:NamedIndividual>

 <owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
      mono#GateRoom-1-2g8JzIRvCKHgTFICSRtLAI">
    <airm-mono:Gate-internationalTyperdf:resource="http://project-
        best.eu/owl/airm-mono#CodeInternationalFlightType-
        INTERNATIONAL_AIR_SERVICE"/>
    <airm-mono:Gate-aircraftstand>
      http://project-best.eu/owl/airm-mono#Stand-1-0
          v9OGrnZkdIQYM8ER6Yfma
    </airm-mono:Gate-aircraftstand>
    <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
        Gate"/>
 </owl:NamedIndividual>

 <owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
      mono#Stand-5-2BHnbKSa5UGAK3X4KVJjdJ">
    <airm-mono:AircraftStand-availability>
      http://project-best.eu/owl/airm-mono#Stand-5-
          ApronAreaAvailability
    </airm-mono:AircraftStand-availability>
    <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
        AircraftStand"/>
 </owl:NamedIndividual>

 <owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
      mono#Stand-4-2VUYBV_$a8GBZ7HGRvBeIh">
    <airm-mono:AircraftStand-availability>
      http://project-best.eu/owl/airm-mono#Stand-4-
          ApronAreaAvailability
    </airm-mono:AircraftStand-availability>
    <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
        AircraftStand"/>
 </owl:NamedIndividual>

 <owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
      mono#GateRoom-2-ApronAreaAvailability">
    <airm-mono:ApronAreaAvailability-operationalStatus rdf:resource
        ="http://project-best.eu/owl/airm-mono#
        CodeStatusAirportType-LIMITED"/>
    <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
        ApronAreaAvailability"/>
```

(*continued on next page*)

(*continued*)

Listing 2: Example of RDF file obtained from the proposed method for the implementation of the case study

```
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
    mono#Stand-1-0v9OGrnZkdIQYM8ER6Yfma">
  <airm-mono:AircraftStand-availability>
    http://project-best.eu/owl/airm-mono#Stand-1-
        ApronAreaAvailability
  </airm-mono:AircraftStand-availability>
  <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
      AircraftStand"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
    mono#GateRoom-2-1XgGxbtPJ2HBK3pZXz1kx0">
  <airm-mono:Gate-internationalType rdf:resource="http://project-
      best.eu/owl/airm-mono#CodeInternationalFlightType-
      DOMESTIC_AIR_SERVICE"/>
  <airm-mono:Gate-aircraftstand>
    http://project-best.eu/owl/airm-mono#Stand-2-1
        K3qjb7UJDHBxFfEpra42T
  </airm-mono:Gate-aircraftstand>
  <airm-mono:Gate-availability>
    http://project-best.eu/owl/airm-mono#GateRoom-2-
        ApronAreaAvailability
  </airm-mono:Gate-availability>
  <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
      Gate"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
    mono#Apron-1-3tkfUGSUSxGf8tkVcty$yD">
  <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
      Apron"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
    mono#Stand-1-ApronAreaAvailability">
  <airm-mono:ApronAreaAvailability-operationalStatus rdf:resource
      ="http://project-best.eu/owl/airm-mono#
      CodeStatusAirportType-NORMAL"/>
  <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
      ApronAreaAvailability"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
    mono#Stand-2-ApronAreaAvailability">
  <airm-mono:ApronAreaAvailability-operationalStatus rdf:resource
      ="http://project-best.eu/owl/airm-mono#
      CodeStatusAirportType-CLOSED"/>
  <airm-mono:ApronAreaAvailability-warning rdf:resource="http://
      project-best.eu/owl/airm-mono#CodeAirportWarningType-
      WORK_IN_PROGRESS"/>
  <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
      ApronAreaAvailability"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
```

(*continued*)

Listing 2: Example of RDF file obtained from the proposed method for the implementation of the case study

```
    mono#TERMINAL-1-AerodromeAvailability">
  <airm-mono:AerodromeAvailability-operationalStatus rdf:resource
      ="http://project-best.eu/owl/airm-mono#
      CodeStatusAirportType-NORMAL"/>
  <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
      AerodromeAvailability"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
    mono#TERMINAL-1-3W4XDoLT4GI9Lweemh40jg">
  <airm-mono:Terminal-numberOfGateRooms>
    2
  </airm-mono:Terminal-numberOfGateRooms>
  <airm-mono:Terminal-internationalType rdf:resource="http://
      project-best.eu/owl/airm-mono#CodeInternationalFlightType-
      INTERNATIONAL_AIR_SERVICE"/>
  <airm-mono:Terminal-availability>
    http://project-best.eu/owl/airm-mono#TERMINAL-1-
        AerodromeAvailability
  </airm-mono:Terminal-availability>
  <airm-mono:Terminal-description>
    Example of a terminal
  </airm-mono:Terminal-description>
  <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
      Terminal"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
    mono#Runway-1-34yJK6u_ZpJfdxC7GKjJ7G">
  <airm-mono:Runway-type rdf:resource="http://project-best.eu/owl
      /airm-mono#CodeRunwayType-FATO"/>
  <airm-mono:Runway-associatedAerodrome>
    http://project-best.eu/owl/airm-mono#Runway-1-Aerodrome
  </airm-mono:Runway-associatedAerodrome>
  <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
      Runway"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
    mono#Stand-4-ApronAreaAvailability">
  <airm-mono:ApronAreaAvailability-operationalStatus rdf:resource
      ="http://project-best.eu/owl/airm-mono#
      CodeStatusAirportType-CLOSED"/>
  <airm-mono:ApronAreaAvailability-warning rdf:resource="http://
      project-best.eu/owl/airm-mono#CodeAirportWarningType-
      WORK_IN_PROGRESS"/>
  <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
      ApronAreaAvailability"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
    mono#PassengerLoadingBridge-2-3JpsdGPVZYHw_68$ds5JHb">
  <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
      PassengerLoadingBridge"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
```

(*continued on next page*)

(*continued*)

Listing 2: Example of RDF file obtained from the proposed method for the implementation of the case study

```
      mono#Stand-3-ApronAreaAvailability">
    <airm-mono:ApronAreaAvailability-operationalStatus rdf:resource
        ="http://project-best.eu/owl/airm-mono#
        CodeStatusAirportType-NORMAL"/>
    <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
        ApronAreaAvailability"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
      mono#PassengerLoadingBridge-1-1jNA9zIcSfGga22WQlgWgz">
    <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
        PassengerLoadingBridge"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
      mono#Stand-3-0GDFxANJfgIQ9ZxyUwCpro">
    <airm-mono:AircraftStand-availability>
      http://project-best.eu/owl/airm-mono#Stand-3-
          ApronAreaAvailability
    </airm-mono:AircraftStand-availability>
    <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
        AircraftStand"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
      mono#Stand-5-ApronAreaAvailability">
    <airm-mono:ApronAreaAvailability-operationalStatus rdf:resource
        ="http://project-best.eu/owl/airm-mono#
        CodeStatusAirportType-NORMAL"/>
    <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
        ApronAreaAvailability"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://jjhm.project.ull.es/owl/
      airm-mono#PROJECT-1-3GzT59e3EzIeZEXAYfx4hb">
    <extra-ifc:OwnerHistory rdf:resource="http://linkedbuildingdata
        .net/ifc/resources20201206_110342/IfcOwnerHistory_2"/>
    <rdf:type rdf:resource="http://jjhm.project.ull.es/owl/airm-
        mono#IfcProject"/>
</owl:NamedIndividual>

<owl:NamedIndividual rdf:about="http://project-best.eu/owl/airm-
      mono#Runway-1-Aerodrome">
    <airm-mono:Aerodrome-designatorIATA>
      tfs
    </airm-mono:Aerodrome-designatorIATA>
    <airm-mono:Aerodrome-country>
      Spain
    </airm-mono:Aerodrome-country>
    <rdf:type rdf:resource="http://project-best.eu/owl/airm-mono#
        Aerodrome"/>
  </owl:NamedIndividual>

</rdf:RDF>
}
```

References

[1] M. Poljansek, JRC Publications Repository - Building Information Modelling (BIM) standardization (2017). doi:10.2760/36471. URL:https://publications.jrc.ec.europa.eu/repository/handle/JRC109656.

[2] European Parliament and the council of the European, Directive 2014/24/EU of the European Parliament and of the Council of 26 February 2014 on public procurement and repealing Directive 2004/18/E (2014). URL:https://eur-lex.europa.eu/eli/dir/2014/24/oj

[3] European Commission., Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions - A Renovation Wave for Europe - greening our buildings, creating jobs, improving lives (2020). URL:https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52020DC0662.

[4] Industry Foundation Classes (IFC) - buildingSMART Technical (2017). URL: https://technical.buildingsmart.org/standards/ifc/.

[5] Eastman CM, Jeong Y-S, Sacks R, Kaner I. Exchange Model and Exchange Object Concepts for Implementation of National BIM Standards. J Computing Civil Eng 2010;24(1):25–34. doi: https://doi.org/10.1061/(asce)0887-3801(2010)24:1(25).

[6] Sacks R, Kaner I, Eastman CM, Jeong YS. The Rosewood experiment - Building information modeling and interoperability for architectural precast facades. Autom Constr 2010;19(4):419–32. doi: https://doi.org/10.1016/j.autcon.2009.11.012.

[7] Lai H, Deng X. Interoperability analysis of ifc-based data exchange between heterogeneous BIM software. J Civil Eng Manage 2018;24(7):537–55. doi: https://doi.org/10.3846/jcem.2018.6132.

[8] Berners-Lee T, Hendler J. Publishing on the semantic web. Nature 2001;410:1023–4. doi: https://doi.org/10.1038/35074206.

[9] Pauwels P, Zhang S, Lee YC. Semantic web technologies in AEC industry: A literature overview. Autom Constr 2017;73:145–65. doi: https://doi.org/10.1016/j.autcon.2016.10.003.

[10] Hliaoutakis A, Varelas G, Voutsakis E, Petrakis EG, Milios E. Information retrieval by Semantic Similarity. Semantic Web Inform Syst 2006;2(3):55–73. doi: https://doi.org/10.4018/jswis.2006070104.

[11] I. NKISI-ORJI, Ontology driven information retrieval, Ph.D. thesis, Robert Gordon University (2019). URL:https://openair.rgu.ac.uk

[12] Tomašević NM, Batić M, Blanes LM, Keane MM, Vraneš S. Ontology-based facility data model for energy management. Adv Eng Inform 2015;29(4):971–84. doi: https://doi.org/10.1016/j.aei.2015.09.003.

[13] Sayed A, Al Muqrishi A, IBRI-CASONTO: Ontology-based semantic search engine, Egyptian Informatics Journal 18 (3) (2017) 181–192. doi:10.1016/j.eij.2017.01.001.

[14] Belsky M, Sacks R, Brilakis I. Semantic Enrichment for Building Information Modeling. Computer-Aided Civil Infrastructure Eng 2016;31(4):261–74. doi: https://doi.org/10.1111/mice.12128.

[15] Linked Building Data Community Group (2017). URL:https://www.w3.org/community/lbd/.

[16] Krima S, Barbau R, Fiorentini X, Sudarsan R, Sriram RD, OntoSTEP: OWL-DL Ontology for OntoSTEP: OWL-DL Ontology for, Tech. rep., National Institute of Standards and Technology (NIST) (2009).

[17] Barbau R, Krima S, Rachuri S, Narayanan A, Fiorentini X, Foufou S, Sriram RD. Ontostep: Enriching product model data using ontologies. Comput Aided Des 2012;44(6):575–90. doi: https://doi.org/10.1016/j.cad.2012.01.008.

[18] Beetz J, Van Leeuwen J, De Vries B. IfcOWL: A case of transforming EXPRESS schemas into ontologies. Artificial Intell Eng Design, Anal Manuf: AIEDAM 2009;23(1):89–101. doi: https://doi.org/10.1017/S0890060409000122.

[19] Corry E, Pauwels P, Hu S, Keane M, O'Donnell J. A performance assessment ontology for the environmental and energy management of buildings. Autom Constr 2015;57:249–59. doi: https://doi.org/10.1016/j.autcon.2015.05.002.

[20] Lee J, Jeong Y, Oh YS, Lee JC, Ahn N, Lee J, Yoon SH. An integrated approach to intelligent urban facilities management for real-time emergency response. Autom Constr 2013;30:256–64. doi: https://doi.org/10.1016/j.autcon.2012.11.008.

[21] González E, Piñeiro JD, Toledo J, Arnay R, Acosta L. An approach based on the ifcOWL ontology to support indoor navigation. Egyptian Inform J 2020;22(1):1–13. doi: https://doi.org/10.1016/j.eij.2020.02.008.

[22] Mignard C, Nicolle C. Merging BIM and GIS using ontologies application to Urban facility management in ACTIVe3D. Comput Ind 2014;65(9):1276–90. doi: https://doi.org/10.1016/j.compind.2014.07.008.

[23] Besenyoi Z, Krämer M, Husain F, Building Information Modelling in Agile Environments - An Example of Event Management at the Airport of Tempelhof, in: MATEC Web of Conferences, Vol. 251, 2018, p. 03064. doi:10.1051/matecconf/201825103064.

[24] Kučera A, Pitner T. Semantic BMS: Allowing usage of building automation data in facility benchmarking. Adv Eng Inform 2018;35(January):69–84. doi: https://doi.org/10.1016/j.aei.2018.01.002.

[25] Niknam M, Karshenas S. A shared ontology approach to semantic representation of bim data. Autom Constr 2017;80:22–36. doi: https://doi.org/10.1016/j.autcon.2017.03.013. URL:https://www.sciencedirect.com/science/article/pii/S0926580517302364.

[26] International Facility Management Association (IFMA), Definition of facility management (2016). URL:https://www.ifma.org/about/what-is-facility-management

[27] Pittet P, Cruz C, Nicolle C. An ontology change management approach for facility management. Comput Ind 2014;65(9):1301–15. doi: https://doi.org/10.1016/j.compind.2014.07.006.

[28] Chen W, Chen K, Cheng JC, Towards an ontology-based approach for information interoperability between BIM and facility management, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 10864 LNCS, Springer International Publishing, 2018, pp. 447–469. doi:10.1007/978-3-319-91638-5_25.

[29] Kim K, Kim H, Kim W, Kim C, Kim J, Yu J. Integration of ifc objects and facility management work information using Semantic Web. Autom Constr 2018;87:173–87. doi: https://doi.org/10.1016/j.autcon.2017.12.019.

[30] Becerik-Gerber B, Jazizadeh F, Li N, Calis G. Application Areas and Data Requirements for BIM-Enabled Facilities Management. J Constr Eng Manage 2012;138(3):431–42. doi: https://doi.org/10.1061/(asce)co.1943-7862.0000433.

[31] Gao X, Pishdad-Bozorgi P. BIM-enabled facilities operation and maintenance: A review. Adv Eng Inform 2019;39(January):227–47. doi: https://doi.org/10.1016/j.aei.2019.01.005.

[32] ISO - ISO 16739-1:2018 - Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries – Part 1: Data schema (2018). URL:https://www.iso.org/standard/70303.html.

[33] Eastman K, Teicholz C, Sacks P, Liston R, A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors, 2nd Edition, John Wiley & Sons, Inc, Hoboken, New Jersey, 2008.

[34] Venugopal M, Eastman CM, Sacks R, Teizer J. Semantics of model views for information exchanges using the industry foundation class schema. Adv Eng Inform 2012;26(2):411–28. doi: https://doi.org/10.1016/j.aei.2012.01.005.

[35] ISO - ISO 10303-11:2004 - Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual (2004). URL:https://www.iso.org/standard/38047.html.

[36] Pauwels P, Terkaj W. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. Autom Constr 2016;63:100–33. doi: https://doi.org/10.1016/j.autcon.2015.12.003.

[37] Schevers H, Drogemuller R, Converting the industry foundation classes to the web ontology language, in: Proceedings - First International Conference on Semantics, Knowledge and Grid, SKG 2005, 2005, pp. 73–75. doi:10.1109/SKG.2005.59.

[38] Pauwels P, GitHub - pipauwel/IFCtoRDF: IFCtoRDF is a set of reusable Java components that allows to parse IFC-SPF files and convert them into RDF graphs. (2016). URL:https://github.com/pipauwel/IFCtoRDF.

[39] BEST - SESAR (2016). URL:https://project-best.eu/publications.html.

[40] Best project, Airm-mono ontology (2018). URL:http://www.project-best.eu/downloads/ontologies/airm-mono/airm-mono.owl.

[41] Technical - buildingSMART International (2020). URL:https://www.buildingsmart.org/standards/rooms/airport/ https://www.buildingsmart.org/standards/rooms/technical/.

[42] E. International, The JSON Data Interchange Syntax - Standard ECMA-404 (2017). URL:https://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf.

[43] International Organization for Standardization (ISO), ISO 10303-11:2004 - Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual (2004). URL:https://www.iso.org/standard/38047.html

[44] P. Pauwels, W. Terkaj, EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology, Automation in Construction 63 (2016) 100–133. doi:10.1016/j.autcon.2015.12.003.

[45] International Organization for Standardization (ISO), ISO 16739-1:2018 - Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries – Part 1: Data schema (2018). URL:https://www.iso.org/standard/70303.html

[46] J.J. Herrera-Martin, Open repository of the project IfcToDomain (2020). URL: https://gitlab.com/jjhm/ifctodomain

[47] Apache Software Foundation, Apache Jena Semantic Web Framework (2010). URL:https://jena.apache.org/

[48] Best project, Airm-mono ontology (2018). URL:http://www.project-best.eu/downloads/ontologies/airm-mono/airm-mono.owl

[49] BEST - SESAR (2016). URL:https://project-best.eu/publications.html